

Evolución de redes neuronales para comandar criaturas que alcanzan objetivos sorteando obstáculos en un entorno virtual 2D

Autor: Leonardo César Corbalán

NºAlumno:1085/0

Director: Lic. Laura Lanzarini

Institución: Facultad de Informática de la Universidad Nacional de La Plata

Fecha de presentación: marzo de 2002

Una nota sobre el estilo

Siguiendo el consejo de algunos investigadores que han escrito pautas para la correcta escritura de una tesis, intentaré ser suficientemente claro en la exposición de cada uno de los temas. Aunque formalmente dirigido a la Comisión Evaluadora no perderé de vista la posibilidad de una eventual lectura por personas no versadas en los conceptos que iré analizando a lo largo del presente. Creo que la formalidad necesaria en un trabajo de esta naturaleza ha de manifestarse a través del tratamiento serio y documentado de los temas que se abordan, y no debe generar austeridad en las explicaciones que las transforman en comprensibles sólo para los idóneos en la materia.

He escrito esta tesis utilizando la primera persona del singular. Sin embargo, siendo el escribir un acto social y debido que aquello que se afirma puede ser compartido por los lectores, a veces es lícito utilizar el plural mayestático, aunque despojado de la connotación original que le dieran los soberanos, papas y reyes que lo utilizaban para expresar autoridad y dignidad. En todos los casos que utilice la primera persona del plural, será para involucrar al lector en lo que se está exponiendo.

Está claro que un texto que aspire a comunicar resultados en un contexto académico o científico debe ser claro, preciso y su estilo de narración preferentemente formal. No obstante este trabajo está conformado por capítulos de naturaleza ligeramente distinta que me conducen también a una ligera variación del estilo narrativo. Mientras que en algunas secciones relacionadas con disciplinas no informáticas, he puesto especial énfasis en que su lectura resulte amena y relajada, siendo su propósito marcadamente informativo, en otras he utilizado una narración más precisa y directa, en la que deben quedar claros detalles de algoritmos, métodos e implementaciones del ámbito de la Neurocomputación y Computación Evolutiva.

Indice

INDICE	III
INTRODUCCIÓN.....	VI
<i>Organización de contenidos</i>	<i>IX</i>
EVOLUCIÓN Y GENÉTICA. UNA MIRADA DESDE LA BIOLOGÍA.....	1
INTRODUCCIÓN.....	1
EVOLUCIÓN BIOLÓGICA.....	2
<i>Los orígenes de la vida y el comienzo de la evolución</i>	<i>3</i>
<i>Evolución por selección natural –Charles Darwin–.....</i>	<i>6</i>
<i>La oposición a las ideas de Darwin.....</i>	<i>10</i>
<i>Las huellas de la evolución.....</i>	<i>12</i>
GENÉTICA.....	16
<i>Introducción histórica.....</i>	<i>17</i>
<i>Genética mendeleana.....</i>	<i>19</i>
<i>Teoría de la genética de poblaciones –neodarwinismo–.....</i>	<i>23</i>
<i>Teoría cromosómica de la herencia.....</i>	<i>24</i>
COMPUTACIÓN EVOLUTIVA.....	43
INTRODUCCIÓN.....	43
<i>Métodos de búsqueda y optimización</i>	<i>43</i>
ORÍGENES DE LA COMPUTACIÓN EVOLUTIVA	45
ALGORITMOS GENÉTICOS.....	47
<i>Representación –Codificación Genética–.....</i>	<i>49</i>
<i>Crossover y Mutación.....</i>	<i>49</i>
<i>Ejemplo de Algoritmo Genético Simple.....</i>	<i>51</i>
<i>Anatomía de un Algoritmo Genético.....</i>	<i>53</i>
<i>Fundamento teórico.....</i>	<i>59</i>
<i>Aspectos Prácticos.....</i>	<i>64</i>
<i>Evaluación del Comportamiento de los Algoritmos Genéticos.....</i>	<i>74</i>
OTROS PARADIGMAS DE LA COMPUTACIÓN EVOLUTIVA.....	78
<i>Estrategias Evolutivas</i>	<i>80</i>
<i>Programas de Evolución.....</i>	<i>80</i>
<i>Programación Genética.....</i>	<i>80</i>
<i>Programación Evolutiva.....</i>	<i>81</i>
<i>Clasificadores genéticos.....</i>	<i>81</i>
<i>Algoritmos miméticos.....</i>	<i>81</i>
<i>Solidificación Simulada</i>	<i>81</i>
REDES NEURONALES. FUNDAMENTOS BIOLÓGICOS	83
INTRODUCCIÓN.....	83
LAS CÉLULAS DEL SISTEMA NERVIOSO	85
<i>La forma de las neuronas</i>	<i>85</i>
<i>Las otras células del cerebro.....</i>	<i>87</i>
EL SISTEMA NERVIOSO. ESPECIALISTA EN COMUNICACIÓN	88
<i>Neuronas receptoras.....</i>	<i>89</i>
<i>Neuronas motoras.....</i>	<i>91</i>
<i>Acerca de la comunicación.....</i>	<i>93</i>
ARQUITECTURA NEURONAL	99
<i>Circuitos neuronales.....</i>	<i>102</i>
<i>Zonas del cerebro. Distintas funciones.....</i>	<i>104</i>
LAS FUNCIONES SUPERIORES	107
<i>La problemática mente y cerebro</i>	<i>107</i>
<i>La percepción</i>	<i>109</i>
<i>La atención</i>	<i>115</i>
<i>El sentido común.....</i>	<i>116</i>

<i>El lenguaje</i>	116
<i>La memoria y el aprendizaje</i>	118
REDES NEURONALES ARTIFICIALES Y NEUROEVOLUCIÓN.....	123
INTRODUCCIÓN.....	123
REDES NEURONALES ARTIFICIALES –RNA–.....	125
<i>Neurona artificial</i>	125
<i>Perceptrón</i>	127
<i>Adaline</i>	138
<i>Backpropagation</i>	141
<i>Arquitectura neuronal</i>	151
<i>Tipos de aprendizaje</i>	152
<i>Modelos de RNA</i>	155
<i>Aplicaciones de RNA</i>	156
REDES NEURONALES ARTIFICIALES EVOLUCIONARIAS –RNAE–.....	157
<i>¿Por qué evolucionar redes neuronales?</i>	157
<i>Evolución de pesos de conexión</i>	158
<i>Evolución de arquitectura</i>	161
<i>Evolución de función de transferencia de los nodos</i>	165
<i>Evolución simultánea de arquitectura y pesos de conexión</i>	165
<i>Evolución de reglas de aprendizaje</i>	166
CRIATURAS VIRTUALES COMANDADAS POR RNAE.....	168
INTRODUCCIÓN.....	168
<i>El presente trabajo en el contexto de la Inteligencia Artificial</i>	169
DOMINIO DE PROBLEMAS A RESOLVER.....	171
<i>El campo de prueba</i>	172
<i>Los obstáculos</i>	173
<i>Los objetivos –presa o destino–</i>	173
<i>Las simulación</i>	174
<i>Los agentes</i>	175
<i>Asignación de fitness</i>	178
TIPOS DE REDES NEURONALES ARTIFICIALES IMPLEMENTADAS.....	184
<i>RNA Feedforward</i>	186
<i>RNA Recurrentes</i>	186
<i>Conexión de tendencia</i>	191
<i>Múltiples funciones de transferencia</i>	192
ESTRATEGIAS DE NEUROEVOLUCIÓN IMPLEMENTADAS.....	194
<i>Neuroevolución Simple</i>	194
<i>Neuroevolución Simple con Algoritmo Genético Elitista</i>	200
<i>SANE –Symbiotic Adaptive Neuroevolution–</i>	201
<i>ESP –Enforced subpopulations–</i>	215
<i>ESP con población de blueprints</i>	221
<i>Grupos Neuronales Evolutivos –GNE–</i>	224
HERRAMIENTA DE SOFTWARE DESARROLLADA. ENE.....	238
INTRODUCCIÓN.....	238
DESCRIPCIÓN DEL PROGRAMA.....	238
<i>Ensayos neuroevolutivos</i>	239
<i>Análisis estadístico de datos</i>	245
<i>Análisis de controladores neuronales salvados</i>	248
EVALUACIÓN Y ANÁLISIS.....	250
INTRODUCCIÓN.....	250
LAS PRUEBAS.....	250
<i>Obtención de datos</i>	251
<i>Escenarios definidos</i>	253
<i>Evaluación de Arquitectura neuronal</i>	256
<i>Evaluación de Estrategias Neuroevolutivas</i>	264
CONCLUSIONES DEL ANÁLISIS DE LOS DATOS.....	275

<i>Sobre las arquitecturas neuronales</i>	275
<i>Sobre las estrategias neuroevolutivas implementadas</i>	276
CONCLUSIONES FINALES	279
REFERENCIAS	284
APÉNDICE 1. ENSAYOS REALIZADOS	289

Introducción

El cerebro humano constituye una computadora muy notable capaz de interpretar información imprecisa, incompleta o con ruido suministrada por los sentidos a un ritmo increíblemente veloz. Si bien se encuentra en franca desventaja para competir en el terreno de los cálculos matemáticos contra una computadora electrónica, hasta un cerebro de un niño será capaz de superarla en tareas de reconocimiento de tramas complejas como por ejemplo identificar un rostro o un timbre de voz conocido entre muchos otros.

El cerebro es flexible, ajustándose a nuevos ambientes, es robusto y tolerante a fallas soportando la muerte de neuronas sin pérdida de rendimiento. Pero sobre todo, el cerebro aprende por sí mismo, sin instrucciones explícitas de ninguna clase, a crear las representaciones internas que hacen posibles sus habilidades.

Inspirado en la eficiencia de este evolucionado órgano, se ha desarrollado la teoría de las Redes Neuronales Artificiales que emulan a las biológicas consiguiendo resolver algunos problemas cotidianos, tal como lo hace el cerebro humano.

El poder de las redes neuronales artificiales está en la capacidad de extraer conocimiento a partir de ejemplos de aprendizaje modificando su comportamiento en función de la experiencia. Una vez entrenadas, son capaces de generalizar –comportamiento típico del pensamiento abstracto– reconociendo entradas nunca antes presentadas por similitud a otras conocidas. De esta manera pueden extraer o abstraer las características esenciales de las entradas aprendidas, procesando correctamente datos incompletos o distorsionados.

Además, dependiendo de la tarea realizada y del tamaño de la red, el fallo de algún nodo de la misma puede no afectar su rendimiento, pues el conocimiento acumulado se halla distribuido en numerosas conexiones logrando así cierta tolerancia a fallas.

Las redes neuronales artificiales han demostrado ser exitosas resolviendo adecuadamente muchos problemas, pero como todo paradigma también presentan debilidades, tales como dificultad para hallar una topología adecuada, dificultad para elegir el valor apropiado de algunos parámetros del entrenamiento, fuerte dependencia con los valores iniciales de los pesos escogidos aleatoriamente, a veces requieren funciones de transferencia derivables, y entre otras más, pueden caer mínimos locales de la función de error que desean minimizar.

Estas dificultades suelen resolverse a fuerza de la experiencia y habilidades de los expertos humanos. Ahora bien, la Computación Evolutiva, que junto al Conexionismo de las redes neuronales artificiales conforma la rama subsimbólica de la Inteligencia Artificial, puede mejorar estos resultados. Así es posible reemplazar el entrenamiento tradicional de las redes neuronales artificiales por la aplicación de los principios darwinianos –evolución por selección natural– a las redes neuronales artificiales debidamente codificadas en un cromosoma. Esto resuelve algunas de las debilidades mencionadas y da origen al nuevo paradigma denominado Neuroevolución.

Los investigadores del área han propuesto distintas estrategias dentro de este paradigma. Entre otras cosas se utilizan los algoritmos evolucionarios para conseguir la arquitectura de la red, los pesos de las conexiones y las funciones de transferencia de los nodos. Incluso se han propuesto hibridaciones combinando aprendizaje evolucionario con las técnicas de entrenamiento propias de las redes neuronales artificiales.

Esta tesina se enmarca dentro del paradigma de la Neuroevolución y, si bien me he ajustado a la propuesta de trabajo de grado presentada, aspiro a tratar unos cuantos conceptos propios de esta renovada rama de la Inteligencia Artificial.

En primer lugar pretendo en este trabajo haber revisado críticamente literatura bastante alejada de los contenidos que habitualmente se estudian en nuestra Facultad de Informática. He intentado interrelacionar distintos puntos de vistas y distintas disciplinas ofreciendo una visión –moldeada, sin lugar a dudas, por mi formación informática– sobre ciertos contenidos propios de la biología, la genética, la psicología, la neurología y en menor proporción de la filosofía de la ciencia. Esta información aspira a reforzar el marco referencial que ayudará a comprender la forma en que las ciencias de la computación articula con otras disciplinas.

Ya sobre terrenos informáticos, pretendo realizar algún aporte estudiando el rendimiento de distintas redes neuronales y su impacto en el desempeño del algoritmo genético que impulsa la evolución. Las diferencias entre ellas se basarán fundamentalmente en las arquitecturas de las mismas.

La arquitectura de una red neuronal es vital, impactando directamente sobre sus capacidades de procesamiento. Una tarea compleja, no podrá llevarse a cabo con una arquitectura de baja conectividad, y en el otro extremo, una red de muy alta conectividad no tendrá la buscada capacidad de generalización.

Estudiaré el comportamiento de arquitecturas feedforward –propagación hacia delante– y recurrentes –con ciclos–, con conexión de tendencia y sin ella, con función de transferencia fija y ajustada por evolución. Así de la combinación de estas tres características –con dos variantes cada una– resultarán 8 tipos diferentes de redes neuronales.

Además, en este trabajo, pretendo analizar el rendimiento de arquitecturas y estrategias neuroevolutivas sobre una clase especial de problemas conocida como “*sequential decision task*”. Los problemas de este tipo se caracterizan por la necesidad de realizar una secuencia de decisiones antes de poder establecer cuál ha sido el efecto de cualquiera de ellas, y aún así, muchas veces no queda claro cuáles decisiones fueron las responsables, y en qué medida, de lo acontecido. Complica la situación el hecho que frecuentemente las decisiones con buenos resultados a largo plazo tengan consecuencias inmediatas negativas. La decisión de caminar en sentido contrario a la salida para poder alcanzarla en un laberinto es un ejemplo de esta situación.

El conjunto de problemas elegido en este trabajo, relacionado con el control de agentes que se mueven en un entorno interactuando con obstáculos y objetivos aunque puede considerarse un *problema de juguete* cae en la categoría de *problemas de control de procesos*, y presenta las mismas dificultades que un problema real. Además es típicamente un problema de la clase *sequential decision task*.

Debido a la complejidad inherente a esta clase de problemas muchas tareas del mundo real permanecen sin automatizar o a lo sumo parcialmente automatizadas. Por lo tanto abordar el estudio de estrategias de aprendizaje máquina –*learning machine*– para conseguir sistemas que resuelvan este tipo de problemas, siempre resulta atractivo.

En un artículo de 1996 el Departamento de Ciencias de la Computación de la Universidad de Texas presentó la primera versión de SANE –Symbiotic Adaptive NEuroevolution– introduciendo en el área un nuevo concepto y demostrando que la evolución de soluciones parciales –nodos de la red– que cooperen en la composición de la solución completa –red neuronal– es más eficiente que el enfoque tradicional de la Neuroevolución. A partir de entonces SANE se ha convertido en un método frecuentemente citado y sobre él se han propuesto variantes y extensiones como ESP –Enforced subpopulations– desarrollado por el mismo laboratorio.

Pretendo que el presente trabajo sea en parte, un estudio sobre el rendimiento de distintas estrategias neuroevolutivas aplicadas a un conjunto de problemas bien definido. Para ello he implementado 6 estrategias distintas:

1. Neuroevolución Simple: Se utiliza el Algoritmo Genético canónico para evolucionar una población de cromosomas que codifican redes neuronales completas.
2. Neuroevolución Simple con Algoritmo Genético Elitista: Igual a la anterior, salvo por la utilización de un algoritmo genético elitista que mantiene dos copias del mejor individuo para la próxima generación.
3. SANE: Estrategia neuroevolutiva que coevoluciona dos poblaciones, una de unidades – neuronas– y otra de blueprints –redes neuronales– que determina la forma de construir las redes a partir de la población de unidades.
4. ESP: Variante de SANE que mantiene una subpoblación separada por cada una de las neuronas que componen la red neuronal evolucionada.
5. ESP con población de blueprints: Hibridación de las dos anteriores propuesta aquí.
6. Grupos Neuronales Evolutivos -GNE-: Nueva estrategia neuroevolutiva propuesta aquí.

Sin embargo, sin minimizar lo antes expuesto, mi mayor expectativa, está centrada en aquellos aspectos novedosos introducidos por la nueva estrategia neuroevolutiva implementada, a la que denominé “Grupos Neuronales Evolutivos” –GNE– , y en este sentido, pretendo que este trabajo de grado se constituya en la presentación, evaluación y análisis de este método.

GNE está inspirado en dos aspectos :

- 1) *Simulación de la organización funcional del cerebro*: Los estudios en el ámbito de las neurociencias han demostrado una gran localización funcional en zonas cerebrales precisas. Esto hace difícil concebir al cerebro como una única gran red neuronal. Más bien, se asemeja a un grupo de circuitos o redes neuronales relacionados con funciones cognitivas, sensoriales y motoras concretas.

En cierto sentido, resolver un problema utilizando una única red neuronal artificial –como se hace habitualmente–, implica estar emulando el funcionamiento de una región cerebral, más que del cerebro mismo. Debido a la enorme cantidad de neuronas que lo conforman, – alrededor de 100.000 millones–, y al alto grado de conectividad de las mismas –entre 10.000 y 100.000 sinapsis cada una– se hace imposible, al menos con la tecnología del momento, construir un sistema artificial remotamente cercano a este magnífico órgano biológico. Sin embargo, podríamos aproximarnos, aunque sea conceptualmente, utilizando un grupo de redes neuronales artificiales para llevar a cabo una tarea en base a la autoorganización y distribución funcional entre sus miembros.

Esta es la idea principal subyacente en GNE, donde se utilizan algoritmos genéticos para evolucionar grupos de varias redes neuronales artificiales. Estas últimas aprenden a especializarse en distintos aspectos de un problema y a delegar en el resto del grupo cuestiones ajenas a su incumbencia. Así, de su accionar coordinado dentro del grupo debe surgir la resolución al problema planteado en forma más eficiente.

- 2) *Simulación del carácter incremental del aprendizaje*: Las habilidades y conocimientos complejos no se adquieren instantáneamente. Por el contrario, es necesaria una serie de instancias de aprendizajes más simples de complejidad incremental. Es posible emular esta situación en los terrenos de la neuroevolución transformando el objetivo de una tarea compleja en una secuencia ordenada de objetivos parciales de complejidad creciente.

Básicamente son dos resultados pretenciosos los que deseo conseguir utilizando GNE:

1) *Mejores soluciones*: Aspiro a evolucionar controladores neuronales¹ que sean más eficientes que cualquiera generado con otro método neuroevolutivo en la ejecución de tareas con cierto nivel de dificultad.

2) *Menor tiempo de evolución*: En problemas que no representen grandes dificultades, dónde otras estrategias alcancen soluciones óptimas, pretendo con GNE alcanzar soluciones de igual rendimiento pero en menor cantidad de generaciones.

Logrado estos dos objetivos, un beneficio lateral importante surgirá de la utilización de GNE como metodología para conseguir la adaptación de un controlador neuronal frente a un cambio repentino en las condiciones del ambiente. GNE se desempeñará con mayor eficiencia que cualquier alternativa que implique atacar el cambio como si fuese un nuevo problema puesto que, gracias a las particularidades de la estrategia incremental, aprovechará lo que aún la red neuronal a ser reemplazada puede resolver con eficiencia.

Finalmente, pretendo con este trabajo mostrar indicios de que plantear la solución a un problema por medio de un grupo de redes neuronales artificiales, en lugar de con una sola, puede convertirse en una línea prometedora de desarrollo y con varios aspectos sobre los cuales hacer investigación.

Para completar esta introducción comentaré la forma en que están organizados los contenidos del presente trabajo de grado.

Organización de contenidos

En el capítulo titulado “Evolución y genética. Una mirada desde la biología” pretendo brindar los elementos necesarios para comprender cómo y a partir de qué surgieron los algoritmos evolucionarios. Se abordará entonces la temática de la evolución biológica comentando las teorías más importantes y me detendré en la obra de Darwin para detallar su teoría general de la evolución por selección natural. Ya en el ámbito de la ciencia iniciada por Gregor Mendel – aunque él nunca conociera la importancia de su trabajo –, se expondrán los mecanismos de la herencia a través de un recorrido por los principios de la genética mendeliana, de la genética de poblaciones y de la importante teoría cromosómica de la herencia. Esta última representa un nexo conceptual con el siguiente capítulo pues expone los principales puntos en que se inspira la Computación Evolutiva.

Luego, en el capítulo “Computación Evolutiva” se abordará la evolución desde el punto de vista de esta disciplina informática, considerada como perteneciente a la vertiente subsimbólica de la Inteligencia Artificial. Introduciré los conceptos básicos de este paradigma, dirigiendo la atención a los Algoritmos Genéticos como rama más representativa y con mayor desarrollo teórico de la Computación Evolutiva. Se presentarán variantes y alternativas al esquema general, sin dejar de lado la fundamentación teórica que descansa en el *teorema de esquemas* y la *hipótesis de bloques constructivos*. Tampoco omitiré las críticas que recientemente han surgido atacando estos basamentos teóricos desde algún sector de la Inteligencia Artificial. Entre otros temas se abordarán aspectos prácticos en relación a la codificación, frecuencia de operadores, la importancia de la diversidad genética, variantes

¹ Utilizaré el término genérico “*controlador neuronal*”, para referirme tanto a una única red neuronal, como a un grupo de ellas.

para asegurarla, y una breve evaluación del desenvolvimiento de los Algoritmos Genéticos en distintas clases de problemas.

Me apartaré, ligeramente y sólo por un momento, del enfoque exclusivamente informático en el capítulo “Redes Neuronales. Fundamentos biológicos”. He notado que en la literatura especializada en Redes Neuronales Artificiales, el apoyo conceptual desde el punto de vista de la neurobiología es escaso o inexistente. Así que destinaré unas 40 páginas a contrarrestar esta deficiencia informativa. Acercaré al lector algunos conocimientos que hoy poseen los científicos sobre el funcionamiento del cerebro. Realizaré una breve descripción de las células que componen el sistema nervioso deteniéndome, como es de esperar, en las neuronas. Se abordará el tema de los circuitos neuronales y de la organización funcional del cerebro que muestra cómo ciertas zonas se asocian con distintas capacidades sensorio-motriz y cognitivas. Esta localización cerebral ha sido importante en la concepción del modelo de neuroevolución propuesto –GNE–, pues sugiere que este enigmático órgano no es una gran red neuronal sino por el contrario un conjunto de ellas que responden selectivamente a los estímulos recibidos. Finalmente se hará un recorrido sobre las principales funciones cerebrales superiores – percepción, atención, sentido común, lenguaje, memoria–. Se revisarán las ideas sobre el aprendizaje de Donal O. Hebb, que ha influido de una u otra forma en las grandes corrientes de pensamiento en neurobiología de la memoria y en el campo informático de las Redes Neuronales Artificiales.

Ya en el capítulo “Redes Neuronales Artificiales y Neuroevolución” describiré algunas de las características principales del paradigma conexionista, enfatizando el contraste existente con los sistemas de cómputo tradicionales. Para ello será necesario estudiar el primer modelo de neurona artificial y algunas arquitecturas neuronales específicas como el Perceptrón, el Adaline, y la Backpropagation. Todas ellas relevantes, por sus implicaciones históricas o por el éxito en aplicaciones concretas. Retomando la perspectiva general describiré distintas clasificaciones y tipos de aprendizajes y comentaré algunas aplicaciones en las que las redes neuronales han demostrado ser eficientes. Luego se dedicarán unas cuantas líneas al tratamiento de la Neuroevolución, paradigma que surge de la combinación de las redes neuronales con la Computación Evolutiva. Se presentarán distintas estrategias de evolución: de pesos de conexión, de arquitectura, de función de transferencia, de arquitectura y pesos simultáneamente y de reglas de aprendizaje.

En el capítulo titulado “Criaturas virtuales comandadas por RNAE”, Comentaré las particularidades del dominio de problemas aquí planteado: control de agentes para esquivar obstáculos y alcanzar objetivos. Se expondrán los pormenores subyacentes en la interacción con el medio –campo de prueba, obstáculos y objetivos–. Luego presentaré las variantes de arquitecturas neuronales implementadas y finalmente las seis estrategias de neuroevolución desarrolladas. Primero, un tipo de neuroevolución simple en la que se utiliza un algoritmo genético convencional junto a una representación cromosómica tradicional. Luego la misma estrategia mejorada por la utilización de la técnica conocida como elitismo. Seguiré con SANE y ESP, dos estrategias desarrolladas por el Departamento de Ciencias de la Computación de la Universidad de Texas, que serán estudiadas junto a las extensiones necesarias que he debido implementar para soportar toda la gama de arquitecturas neuronales desarrolladas para este trabajo. Una hibridación de estos dos últimos métodos será presentada bajo el título de “ESP con población de blueprints”. Y para finalizar, se expondrá GNE – Grupos Neuronales Evolutivos–, la nueva estrategia que he propuesto para atacar la solución a problemas complejos por medio de más de una red neuronal y de un mecanismo de evolución incremental. Se comentarán algunos artículos, el planteo general, la versión genérica del método y su adecuación al dominio de problemas específico aquí presentado.

En el capítulo “Herramienta de software desarrollada. ENE.” se presentará el software que acompaña este trabajo que ha sido utilizado como la herramienta fundamental para analizar y evaluar el rendimiento de distintas arquitecturas neuronales y de las 6 estrategias neuroevolutivas implementadas. Se realizará una descripción del programa con el que básicamente se puede: llevar a cabo los ensayos neuroevolutivos, visualizar en simulaciones el accionar de los agentes, inspeccionar en cualquier momento el estado interno de las redes neuronales y analizar estadísticamente en forma de gráficos el desempeño de las evoluciones.

En el capítulo “Evaluación y Análisis” se definirán varios casos de pruebas tendientes a evaluar el rendimiento de las distintas clases de redes implementadas y de las 6 estrategias neuroevolutivas desarrolladas, enfatizando el análisis sobre SANE, tomado como referente comparativo, y sobre “ESP con población de Blueprints” y GNE por aportar aspectos novedosos que se desean evaluar. Se realizará una descripción de los experimentos ensayados argumentando su motivación y acompañados por los gráficos de curvas necesarios desde los que se elaborarán los resultados. Finalmente, para terminar el capítulo, se presentarán en de forma ordenada las conclusiones que se desprenden directamente de la experimentación.

Por último, el presente trabajo de grado finaliza con las “Conclusiones Finales”.

Antes de comenzar con el primer capítulo quiciera observar que, en ningún momento he dejado de cumplir cada uno de los puntos detallados en la propuesta de trabajo de grado presentada. Quizás existan algunas páginas dedicadas a temas que no están explícitamente indicados en la propuesta, pero que juzgué pertinente incorporarlos a los efectos de convertir este trabajo en una presentación razonablemente autocontenida, sin dejar demasiados huecos o verdades a medio explicar que disminuyan el interés del lector.

Evolución y genética. Una mirada desde la biología

Introducción

El término evolución en el contexto de las ciencias naturales, está íntimamente ligado a la figura de Charles Darwin. Éste fue el iniciador de una de las revoluciones del pensamiento más grande de la historia. Su obra ha influenciado considerablemente el desarrollo de toda la filosofía. Hoy se designa con el término Darwinismo al cuerpo doctrinal de carácter naturalista¹, sociológico y psicológico que explica las transformaciones graduales del mundo sobre la base de la selección natural, que lleva a la supervivencia, propagación y combinación de los caracteres más adecuados.

Darwin tuvo muchos opositores, tanto religiosos como científicos. Algunos ataques a su teoría se originaron en el desconocimiento de la época sobre la transmisión individual de los caracteres hereditarios, descubierta por Gregor Mendel, el primer genetista de la historia.

Los avances en la ciencia de la genética, han permitido confirmar la teoría de Darwin y han establecido los detalles que explican cómo se transmite y recombina el material hereditario de generación en generación.

Este capítulo está organizado de la siguiente forma. Inmediatamente se abordará la temática relativa a la evolución biológica. Se expondrán las teorías más importantes al respecto, sin dejar de lado a aquéllas que versan sobre el origen mismo de la vida. Se hará especial énfasis en la obra darwiniana detallando su teoría general de la evolución y selección natural. También se dedicarán algunas líneas a la llamada selección sexual propuesta por Darwin y generalmente poco citada.

La segunda mitad del capítulo está dedicada a la ciencia de la genética. En ella se verán los conceptos necesarios para comprender cómo la Naturaleza ha implementado los mecanismos de la herencia y evolución en los seres vivos. Luego de una breve reseña histórica se analizarán someramente los principios de la genética mendeliana y de la genética de poblaciones –neodarwinismo–, para finalizar con la teoría cromosómica de la herencia. Esta última de suma importancia en el contexto del presente trabajo.

El objetivo aquí planteado es el de brindar los elementos necesarios para comprender cómo y a partir de qué surgieron los *algoritmos evolucionarios* que se expondrán en el próximo capítulo. Comprender la teoría de Darwin permitirá apreciar la analogía existente entre la evolución de los seres vivos y los problemas de búsqueda y optimización informáticos². Manejar los conceptos de la teoría cromosómica de la herencia servirá para comprender la inspiración de John Holland, quién en la década del 70 presentó los Algoritmos Genéticos como un método adecuado para la resolución de aquellos problemas.

¹ El presente capítulo se interesa sólo en el aspecto naturalista del Darwinismo (ver “Darwinismo” –pág. 7–)

² Para obtener información sobre este tipo de problemas ver “Métodos de búsqueda y optimización” –pág. 43–.

Evolución biológica

Definición 1: Descendencia con modificaciones, proceso por el que todos los seres vivos de la Tierra han divergido, por descendencia directa, a partir de un origen único que existió hace más de 3.000 millones de años. [Encarta98]

Definición 2: Conjunto de procesos caracterizados por cambios biológicos y orgánicos de los organismos por los que los descendientes llegan a diferenciarse de sus antecesores. [ENCICLONET]

Ambas definiciones hacen uso de términos tales como *proceso*, *modificaciones*, *cambios*, *descendencia*. Efectivamente, por evolución se entiende algo que se desenvuelve o desarrolla, un cambio ordenado y gradual de un estadio a otro.

Una vez formulado como hipótesis coherente, el concepto de evolución, necesitó ser demostrado como cualquier otra teoría científica. Desde los tiempos de Darwin, diversas disciplinas biológicas se han encargado de suministrar estas pruebas. Hoy día, la bioquímica y genética que estudian la herencia y la variación, no sólo han aportado nuevas pruebas, sino también los elementos necesarios para comprender los mecanismos del proceso evolutivo.

En la antigua Grecia ya existía la noción de que las especies se transformaban en otras especies; también se elaboraron teorías de cómo se transmitían los caracteres de padres a hijos¹, sin embargo, estas ideas al igual que otras tantas esgrimidas tempranamente por los griegos, estuvieron dormidas por más de dos mil años, hasta el siglo XVIII en que fueron retomada por grandes pensadores como Pierre de Maupertuis, Erasmus Darwin y Jean Baptiste de Lamarck. Fue Charles Darwin –nieto de Erasmus– quien estableció finalmente la teoría de la evolución a través de la famosa publicación *Sobre el origen de las especies por selección natural o la supervivencia de las razas mejor dotadas en la lucha por la existencia* en 1859, conocida más comúnmente como *El origen de las especies*.

La biología molecular moderna hace difícil dudar que el origen de todas las especies se remonta a un antecesor común único, ya que todas las formas de vida conocidas comparten el mismo código genético² y es muy improbable que tal situación se alcanzara de forma independiente. A este ancestro primigenio se lo llama *cenancestor*.

¹Las primeras teorías sobre la herencia fueron expuestas por Hipócrates (460–377 a.C.), el médico más importante de la antigüedad, considerado el padre de la medicina, quien explica el parecido de los hijos a sus padres por el accionar de una especie de semillas distribuidas por todo el cuerpo, que son transmitidas a los hijos en el momento de la concepción. Aristóteles rechazó estas teorías un siglo después. Según él, el semen de los machos podía contener partículas heredadas de generaciones pasadas; en la fecundación se producía una mezcla del flujo masculino con lo que él llamó el *semen femenino* (flujo menstrual), y a partir de esa mezcla se formaba la carne y la sangre de los individuos.

²Hay excepciones, se conocen por lo menos unos cuantos casos en los que la transcripción de la información del ADN en información de proteína en una mitocondria (organoide que se encuentra dentro de las células e interviene en la respiración celular) utiliza un código diferente del utilizado por los genes del núcleo de esta misma célula. Esto sugiere una separación evolutiva de los códigos genéticos de las mitocondrias y de los núcleos, y concuerda con la idea de que las mitocondrias fueron antes organismos libres que se incorporaron a la célula en una relación simbiótica hace miles de millones de años. [Sagan_1]. La célula anfitriona ofrece protección a la invasora (el citoplasma, protegido por una membrana) mientras que la huésped trabaja para la célula produciendo energía por medio de la oxidación de carbohidratos para convertirlos en dióxido de carbono y agua.

Los orígenes de la vida y el comienzo de la evolución

Las estrellas y planetas, la topografía terrestre, los compuestos químicos del universo e incluso las partículas subatómicas, han pasado por alteraciones más o menos lentas a las que se denomina evolución inorgánica. Los astrónomos modernos han podido explicar de manera convincente, la evolución de la materia a partir del “*Big ban*” y la manera en que el Universo se ha ido poblando de materiales pesados sintetizados en el interior de las estrellas y esparcidos por medio de las explosiones estelares.

Por otra parte, el principio de la evolución orgánica –la que interesa en el presente capítulo– sostiene que todos los seres vivos de hoy día han descendido de especies más simples por modificaciones graduales que se han fijado y acumulado en generaciones sucesivas.

Si se acepta este principio de la evolución orgánica, surge inmediatamente el siguiente doble interrogante: ¿Cuál fue entonces la primera especie de la cuál han descendido todas las demás y cómo fue su particular origen pues no pudo descender de ninguna anterior? En otras palabras ¿Cuál fue el *Big ban de la vida* en el planeta Tierra? Aunque resulte por demás interesante, no se pretende profundizar aquí demasiado en esta cuestión, pues escapa a los objetivos de este capítulo en el que se intenta brindar un marco de referencia sobre las teorías en las que se inspiró Holland al desarrollar los Algoritmos Genéticos. Sin embargo, no es posible desentenderse totalmente del tema. La evolución de los seres vivos está supeditada a la existencia de la vida misma y por ello se abordará el origen de ésta en la Tierra, aunque más no sea de manera superficial.

Generalmente resulta fácil asociar el término evolución con ejemplos ampliamente difundidos tales como los cambios sufridos por el caballo que al migrar a la llanura se adaptó haciéndose más alto y fuerte¹, o el de alguna mariposa que debió cambiar el color de sus alas para camuflarse con la corteza de un árbol oscurecida por la contaminación ambiental². Pero, más difícil resulta imaginar que la evolución alcanza también a formas mucho más primitivas tales como organismos unicelulares, bacterias, virus etc. Más aún, los científicos han observado, que los principios establecidos por Darwin, que se verán más adelante –pág. 7– pueden aplicarse también a entidades todavía más primitivas, que seguramente hayan estado presente en la génesis misma de la vida: las primeras moléculas orgánicas autorreplicables.

Es probable que al principio, hace más de 4.000 millones de años, la atmósfera de la Tierra contuviera metano, amoníaco, dióxido de carbono y otros gases que abundan también en otros planetas del sistema solar. Los químicos han reconstruido en los laboratorios estas condiciones primitivas en el ámbito experimental. Si se mezclan los gases adecuados con agua en un matraz, y se añade energía mediante una descarga eléctrica –simulando la iluminación primitiva–, se sintetizan de forma espontánea sustancias orgánicas. Entre éstas se cuentan, en una proporción significativa, aminoácidos –unidades que construyen las proteínas,

¹ El mamífero extinto llamado Eohippus, del tamaño de un perro, no tiene gran semejanza con los caballos actuales. En la naturaleza evolucionó poco a poco hasta convertirse en un rápido corredor, relativamente grande y fuerte, adaptado a vivir en las llanuras (el caballo que hoy conocemos). Luego el hombre, al domesticarlo, continuó con una selección artificial produciendo una gran variedad de razas con propósitos específicos.

² Es particularmente conocido y citado en mucha biografía el caso de una especie de mariposas nocturnas de Inglaterra que se posan en el tipo de abedul que existe en este país, donde la coloración blanca del insecto le procura un camuflaje protector. Cuando a raíz de la revolución industrial la corteza del abedul empezó a cubrirse de hollín una mutación que aportaba más melanina hasta el momento perjudicial se transformó en beneficiosa y con el correr del tiempo, la mayor parte de las mariposas adquirieron la pigmentación negra, rasgo hereditario transmitido a las futuras generaciones.

incluyendo todas las enzimas importantes que controlan los procesos químicos de la vida—, purinas y pirimidinas —unidades que forman el ARN y ADN—¹. Probablemente al principio de la existencia de la tierra sucediera algo similar. Por consiguiente, el mar podría haber sido un “caldo” de compuestos orgánicos prebiológicos.

Aunque parezca asombroso que los científicos hayan logrado sintetizar moléculas orgánicas, en realidad, no es tan significativo. La vida es mucho más que aminoácidos y proteínas². El paso más importante de la Naturaleza, en la génesis de la vida, fue la aparición de moléculas que se autorreplicaban, capaces de producir copias de sí mismas, cosa que aún los científicos no han podido lograr en sus experimentos. Hoy, la molécula más conocida que se autorreplica es el ácido desoxirribonucleico —ADN—. No obstante, seguramente no ha sido la primera, ya que su replicación depende demasiado de estructuras muy especializadas que no pudieron existir antes del inicio de la propia evolución. El ADN es una molécula de “alta tecnología” que seguramente haya aparecido algún tiempo después del origen de la vida. Tal vez la molécula con la que está emparentada, el ácido ribonucleico —ARN—, que aún desempeña varias funciones vitales en las células vivas, fue la molécula autorreplicativa original. O tal vez ésta fue un tipo de molécula diferente.

La forma de vida más simple que se conoce hoy, en el límite con lo inerte, son los viroides. Aún más simples que los virus no son más que un simple filamento de ARN con una geometría o bien lineal o bien circular y cerrada y se componen de menos de 10.000 átomos. También son los responsables de varias enfermedades. Los virus son un poco más complejos pues presentan un recubrimiento de proteínas que los viroides no poseen. Tanto virus como viroides carecen de vida independiente pero se pueden replicar en el interior de las células vivas, a las que parasitan apoderándose de su maquinaria molecular perjudicándolas en el proceso. Es posible que las primeras manifestaciones de vida fuesen muy similares a nuestros viroides actuales y aparecieran hace unos 4.000 millones de años.

Posiblemente la selección natural expuesta por Darwin, a la que haremos referencia más adelante en este capítulo, haya entrado en juego en el mismo momento que aparecía la incipiente forma de vida conformada por las moléculas autorreplicativas. Errores de copiado de estas moléculas producirían variaciones de la misma. Las variantes con una replicación especialmente buena habrían predominado automáticamente en el caldo primitivo, mientras que aquellas que no se replicaron, o que lo hicieron de forma errónea, estarían en una proporción relativamente menos numerosa. Por lo tanto, una forma de selección natural molecular condujo a una eficacia mayor entre las moléculas que se replicaban.

Esta competitividad entre las moléculas debió llevar a que algunas desarrollaran aquellas estructuras previas a las membranas que proporcionaron espacios circunscritos donde incluir

¹ El primero en realizar este experimento fue Stanley Miller, trabajando como tesario en la Universidad de Chicago con Harold Urey —premio Nobel de química— en 1953. Tal experiencia mostró que la frontera entre lo inerte y lo vivo no implicaba un salto cualitativo como muchos lo creían en ese entonces. Significó una demostración empírica convincente de la *continuidad* de la materia, que va de lo más sencillo, como los átomos de hidrógeno, oxígeno, nitrógeno y carbono, a lo más complejo, pasando por los aminoácidos y las proteínas, hasta llegar en último término a la complejidad de las estructuras del cerebro. [Gutierrez01]

² La materia viva está formada por una serie de componentes químicos, pero es interesante hacer observar que ninguno de dichos componentes está dotado de vida; por lo tanto, podemos decir que la materia viva es un sistema organizado de sustancias que por sí mismas no son vivientes. Sin embargo, a partir de un determinado nivel de organización de tales componentes químicos, aparece la manifestación vital. Así, pues, la materia viva no puede ser considerada como una simple “sustancia”, sino como una “organización” muy especial de la materia en constante estado de actividad física y química, que confiere a aquella unas propiedades características, de las que carecen los seres inertes. [ENCICLONET]

las reacciones químicas, vislumbrándose la aparición de los primeros organismos unicelulares. Pudo haber sido poco después de este estadio cuando aparecieron las criaturas simples del tipo de las bacterias hace más de 3.000 millones de años.¹

Los representantes del reino animal y vegetal se hicieron esperar. Primero aparecieron los flagelados.² Los flagelados proporcionan indicios de cómo pueden haber evolucionado las formas de vida multicelulares, ya que presentan características comunes con las plantas, los animales y los hongos, y cuando forman colonias muestran una actividad coordinada. Mientras un grupo se parece a las plantas –contienen clorofila y realizan fotosíntesis– otro se parece a los animales –tienen cuerpo blando, son incoloros y se alimentan de otros organismos–.

Se considera que los animales han evolucionado de los flagelados. La mancha óptica que está en la base del flagelo y que es sensible a la luz, debió especializarse para ver mejor. Todos los vegetales son sensibles a la luz, pero su sensibilidad está repartida por casi todo el cuerpo. Para que mejore la sensibilidad a la luz se necesita que se concentre en un solo lugar. Se puede decir que desde el punto de vista biológico la diferencia entre el vegetal y el animal consiste en que el vegetal difundió su visión por todo el cuerpo; en cambio el animal concentró la visión en un órgano, que en los superiores se llama ojo. [ECNMAE]

Durante la mayor parte de los 4.000 millones de años transcurridos a partir del origen de la vida, la evolución avanzó extremadamente lenta, debido en gran parte a la ausencia de un mecanismo que facilitara la recombinación genética y que fuera inventado tardíamente por la Naturaleza, a saber, la reproducción sexual. Así, la evolución sólo se impulsaba por acumulación de mutaciones casuales en las instrucciones genéticas. Pero, gracias a la nueva invención desarrollada hace unos 2.000 millones de años, los organismos pudieron intercambiar y recombinar grandes cantidades de código genético produciendo grandes variedades y aumentando la posibilidad de conseguir una variante mejor, más hábil y mejor adaptada en menor tiempo.

El resto de la evolución puede ser considerada como una continuación de aquella selección natural de las moléculas replicativas, ahora denominadas genes, debida a su capacidad para construir por sí mismas estructuras eficaces –cuerpos celulares y multicelulares– para su propia supervivencia y reproducción.

La vida apareció tempranamente en el joven planeta Tierra de unos 800 millones de años de edad³, pero durante más de 3.000 millones de años no evolucionó más allá de las algas verdeazuladas, lo que sugiere que la evolución de formas vivas grandes con órganos especializados es difícil de conseguir, más difícil todavía que el origen de la vida misma.

¹ Los organismos independientes más pequeños que se conocen son mucho más complejos que los virus y viroides y están compuestos de aproximadamente unos 50 millones de átomos. Las bacterias son organismos unicelulares, que carecen de núcleo diferenciado y se reproducen por división celular sencilla. En el actual sistema de clasificación en cinco reinos, las bacterias, al igual que las algas verdeazuladas, pertenecen al reino Monera (o procariota).

² Son organismos unicelulares eucariotas –con núcleo celular definido– pertenecientes al reino Protista –un paso adelante en la evolución, por encima de las células procarióticas– cuya característica común es la existencia de unas prolongaciones en forma de látigo denominadas flagelos, que les sirven para desplazarse.

³ Distinta bibliografía señala la fecha de la formación del sistema solar y la Tierra con algunas diferencias. Para la realización de este trabajo se ha tomado el origen de nuestro planeta hace unos 4.800 millones de años.

Sin embargo, hace unos 600 millones de años se produjo una proliferación enorme de nuevas formas vivas, acontecimiento que se ha llamado la explosión del Cámbrico.

“...Después de la explosión del Cámbrico nuevas y exquisitas adaptaciones se fueron sucediendo con una rapidez relativamente vertiginosa. Aparecieron en rápida sucesión los primeros peces y los primeros vertebrados; las plantas que antes se limitaban a vivir en los océanos empezaron la colonización de la Tierra; evolucionaron los primeros insectos y sus descendientes se convirtieron en los pioneros de la colonización de la tierra por los animales; insectos alados nacieron al mismo tiempo que los anfibios, seres parecidos en cierto modo al pez pulmonado, capaces de sobrevivir tanto en la tierra como en el agua; aparecieron los primeros árboles y los primeros reptiles; evolucionaron los dinosaurios; emergieron los mamíferos y luego los primeros pájaros; aparecieron las primeras flores; los dinosaurios se extinguieron; nacieron los primeros cetáceos, antepasados de los delfines y de las ballenas y también en el mismo período nacieron los primates: los antepasados de los monos, los grandes simios y los humanos. Hace menos de diez millones de años evolucionaron los primeros seres que se parecían fielmente a seres humanos, acompañados por un aumento espectacular del tamaño del cerebro. Y luego, hace sólo unos pocos millones de años, emergieron los primeros humanos auténticos.”

Carl Sagan, COSMOS

Así doy por finalizada esta pequeña introducción al origen de la vida, y con ella de la evolución biológica, en la Tierra. Veamos entonces la teoría de Charles Darwin, quien fue capaz de descubrir el mecanismo que la gobierna y al que denominó *selección natural*.

Evolución por selección natural –Charles Darwin–

La comprensión del concepto de evolución puede surgir lógica y naturalmente de los conocimientos actuales de genética. Pero no fue así como surgió históricamente. Se obtuvo como resultado de numerosas observaciones de similitudes y diferencias en estructuras y funciones de las diversas clases de animales y plantas en diferentes partes del mundo.

Antes de Darwin –Juan Bautista de Lamarck–

La teoría original de la evolución –hoy desacreditada– expuesta en forma lógica se debe a Juan Bautista de Lamarck, zoólogo francés, cuya *Filosofía Zoológica* se publicó en 1809. Lamarck suponía que los seres vivos están animados por una fuerza innata con la cual luchan frente al antagonismo del ambiente. Aceptaba también que las adaptaciones a ese ambiente, una vez fijadas, se propagan a las generaciones sucesivas, o sea que los caracteres adquiridos se heredan. Al desarrollar el concepto que aparecen nuevos órganos como respuesta a las necesidades de la lucha con el medio, dedujo que su tamaño e importancia se relaciona con la ley del "uso y la falta de uso", lo cual también se hereda en el curso de las generaciones.

Lamarck afirmaba que el cuello largo de la jirafa evolucionó como resultado de varias generaciones de jirafas que tenían que estirar sus cuellos para alimentarse de las hojas de los árboles. Cada generación transmitió a sus descendientes el pequeño incremento en la longitud del cuello ocasionado por el continuo estiramiento. No obstante, hoy la evidencia no apoya la teoría lamarckista de la herencia de las características adquiridas.

A pesar de los intentos repetidos para probar que los cambios corpóreos adquiridos por un individuo pueden ser transmitidos a sus descendientes, no se ha podido descubrir evidencia alguna. Los primeros experimentos efectuados para tratar de resolver el problema consistieron en remover quirúrgicamente alguna parte de un cuerpo; por ejemplo, la cola de un ratón. Aún

después de haber efectuado tal operación a través de varias generaciones, los ratones nacían siempre con cola, la cual continuaba siendo tan larga como de costumbre.

Hoy se sabe que para que los cambios efectuados en el cuerpo de los padres pudieran ser transmitidos a las descendencias, tendrían que ser incorporados en los gametos, puesto que estos constituyen el único eslabón entre los cuerpos de los progenitores y los cuerpos de los descendientes. Quizás podría lograrse tal resultado si las células especializadas del cuerpo sobre las cuales pudiera efectuarse alguna alteración, produjeran luego los gametos. Pero desde hace muchos años se sabe que en los animales las células del cuerpo que producen gametos se diferencian en las primeras etapas del desarrollo embrionario. De hecho, una niña recién nacida ya ha formado y comenzado la primera división meiótica¹ de donde provendrán todos y cada uno de los óvulos maduros que algún día producirá.

Inconcebible o no, la evidencia está en contra de la teoría de Lamarck, y si bien explicaría la adaptación de muchos vegetales y animales al medio, en definitiva es inaceptable, puesto que las pruebas genéticas de hoy día son decisivas en el sentido de que los caracteres adquiridos no se heredan².

Darwinismo

“Probablemente todos los seres orgánicos que hayan vivido nunca sobre esta tierra han descendido de alguna única forma primordial, a la que se infundió vida por primera vez... Esta opinión sobre el origen de la vida tiene su grandeza... porque mientras este planeta ha ido dando vueltas de acuerdo con la ley fija de la gravedad, a partir de un inicio tan sencillo han evolucionado y siguen evolucionando formas sin fin, las más bellas y las más maravillosas.”

Charles Darwin, *El origen de las especies*, 1859³

La contribución de Charles Darwin a los conocimientos científicos fue doble: presentó las pruebas para demostrar que la evolución había ocurrido, a la vez que formuló una teoría, la de la selección natural, para explicar el mecanismo de la evolución. La publicación de Darwin, en 1859, de *El Origen de las Especies* es un hito, no sólo en la historia de la biología sino, asimismo, en la del pensamiento humano, puesto que dicho libro, aportando una demostración positiva de la doctrina evolucionista, ha ejercido una considerable influencia en el desarrollo de toda la filosofía. La teoría de evolución por selección natural es un concepto filosófico muy poderoso. Sin embargo, permaneció impensado por los más de dos mil años de historia de la filosofía occidental que comenzara con los antiguos griegos. Es una idea genial que reúne en sí al mismo tiempo una gran simplicidad y un enorme poder explicatorio. [Gutierrez97_1]

En su viaje alrededor del mundo, que comenzara en 1831 y se prolongara durante cinco años, Darwin reunió gran cantidad de observaciones interesantes, estableció analogías, meditando

¹ La meiosis y la mitosis son dos tipos de división celular. Mientras que las células somáticas se dividen por mitosis, las sexuales se originan a través de la meiosis. (Ver “La meiosis” –pág. 26–).

² Los algoritmos genéticos se inspiraron en la moderna teoría evolutiva de los seres vivos, no obstante, no deben limitarse a ella. La herencia tal cual la concebía Lamarck puede resultar válida en el contexto de tales algoritmos. Un ejemplo de ello lo podemos encontrar en [Castillo] donde se evolucionan redes neuronales del tipo Perceptrón multicapa –para más información sobre esta arquitectura vea “Perceptrón multicapa” en la pág.136–, utilizando el entrenamiento de las redes como un nuevo operador genético. De esta manera un descendiente codifica genéticamente información extragenética de su progenitor obtenida con un algoritmo estándar de aprendizaje para redes neuronales de su tipo.

³ Citado por Carl Sagan en su libro COSMOS [Sagan_1]

sobre grandes cuestiones, tales como la adaptación de los seres vivos, la diversidad de las especies y sus mutuas relaciones, la lucha por la existencia, la formación de las islas volcánicas. A su vuelta Darwin se dedicó a redactar su *Diario de viaje*; publicó diversos trabajos de geología, en especial sobre la formación de los corales y de las islas volcánicas. Veinte años después de su regreso a Inglaterra publicó *El Origen de las especies*. Escribió luego numerosos libros tales como *La Variación de los animales y de las plantas*, y *La Descendencia del hombre*.

En 1858 Darwin recibió un manuscrito de Wallace, joven naturalista que entonces estaba estudiando la distribución de las plantas y animales en la India y la Península Malaya. En su obra, Wallace formulaba la idea de la selección natural, a la cual había llegado sin conocer la obra darwiniana. Por acuerdo mutuo, Darwin y Wallace presentaron en colaboración un informe sobre su teoría a la Sociedad Linneo de Londres, aquel mismo año¹.

Selección natural

La explicación propuesta por Darwin y Wallace respecto a la forma en que ocurre la evolución, puede resumirse en la forma siguiente:

- La posibilidad de variación es característica de todas las especies de animales y plantas.
- De cualquier especie nacen más individuos de los que pueden obtener su alimento y sobrevivir. Sin embargo, como el número de individuos de cada especie sigue más o menos constante bajo condiciones naturales, debe deducirse que perece un porcentaje de la descendencia en cada generación.
- Sentado que nacen más sujetos de los que pueden sobrevivir, tiene que declararse una lucha por la existencia, una competencia en busca de espacio y alimento. Esta lucha es directa o indirecta, como la de los animales y vegetales para sobrevivir ante condiciones de falta de agua o de bajas temperaturas o a otras condiciones desfavorables del medio ambiente.
- Aquellas variaciones que capacitan mejor a un organismo para sobrevivir en un medio ambiente dado favorecerán a sus poseedores sobre otros organismos menos bien adaptados. Las ideas de la "lucha por la supervivencia" y "supervivencia del más apto" son la esencia de la teoría de la selección natural, de Darwin y Wallace.
- Los individuos supervivientes originarán la siguiente generación, y de este modo se transmiten variaciones "aventajadas" a la siguiente generación y a la siguiente.

Debe destacarse que Darwin fue capaz de demostrar cómo se poblaron continentes y mares de vida por medio de un procedimiento mecánico que nada tiene que ver con la inteligencia, voluntad o propósito de nadie, aunque en apariencia resulte lo contrario. Es probable en este punto que el lector esté pensando que estas palabras que aluden a comportamiento aparentemente inteligente generado por un procedimiento puramente mecánico se ajustan perfectamente a los programas de Inteligencia Artificial. En el próximo capítulo haré nuevamente referencia a esta relación en el apartado “Métodos de búsqueda y optimización” –pág. 43–.

¹ El nombre de Wallace ha quedado opacado ante la figura de Darwin a pesar de haber formulado la misma teoría y ser coautor de su principal obra. La razón de ello es la enorme cantidad de pruebas aportada por Darwin para sostener la teoría que lo posiciona como padre de la misma.

Estas ideas no fueron bien recibidas por la comunidad religiosa de la época y Darwin fue duramente criticado por la Iglesia y aún por algunos científicos que, siendo en su mayoría cristianos, no aceptaban la idea de una Creación sin el rol protagónico de Dios. Más adelante retomaré el tema de la oposición a la teoría darwiniana, pero ahora veamos unos conceptos más que forman parte de su idea sobre la evolución.

Selección sexual

Darwin hizo una distinción entre selección natural, que favorecía los órganos y estructuras orientadas a la supervivencia, y selección sexual que favorecía aquellos logros dirigidos a obtener pareja, por combate directo con los miembros de su propio sexo, o por su atractivo para el sexo opuesto.

Darwin quedó impresionado por el hecho de que, con frecuencia, las cualidades de atractivo sexual eran contrarias a aquellas que conducían a la supervivencia. Un ejemplo notorio son las colas llamativas e incómodas de las aves del paraíso, que deben estorbarles durante el vuelo y son visibles para los depredadores. Sin embargo, Darwin se dio cuenta que estos obstáculos podrían merecer la pena si también atraían a las hembras. Es probable que un macho que consigue persuadir a una hembra para que se aparee con él en lugar de con un rival contribuya con sus caracteres hereditarios a la población. Aquellas aves de colas con atractivo sexual tienen entonces una ventaja que compensa a las desventajas señaladas.

Darwin expuso la teoría de selección sexual de una manera algo tímida en su obra *El origen de las especies*, pero en 1867, escribió a Wallace diciendo: «Estoy casi decidido a publicar un breve ensayo sobre el origen del ser humano, y sigo convencido de que fue la selección sexual la fuerza principal que preformó las razas humanas». Cuando, en 1871, Darwin presentó su obra titulada *El origen del hombre y la selección natural en relación con el sexo*, causó con ella nueva indignación. No afirmaba en ella que la humanidad descendiera de los monos actuales, como sus opositores se empeñaban en ver, sino que decía, por el contrario que el antecesor del hombre fue, según su expresión, «un antiguo miembro del subgrupo antropoide», definición que aún vale. Darwin creía que este lejano antecesor había vivido en Africa, y los descubrimientos científicos de las últimas décadas permiten opinar que así fue, en efecto.

Wallace, el codescubridor de la selección natural, disentía de Darwin respecto a la selección sexual. Él creía que detrás de todas las características aparentemente ornamentales existía una función útil que se debía buscar. Más tarde hizo la concesión de que algunos adornos eran utilizados por los machos para atraer a las hembras. Por el contrario, Darwin creía que los caracteres seleccionados sexualmente como las colas de las aves del paraíso eran inútiles excepto en la medida en que se adecuaban a los caprichos de la hembra. Consideró estos últimos como algo dado, algo que no necesitaba explicación.

Más tarde algunos creyeron que la selección natural podría actuar inevitablemente sobre las hembras para cambiar sus gustos, de modo que no fueran atraídas por cualidades que al heredar sus hijos sólo podían ponerlos en peligro. Tras la muerte de Darwin, la teoría de la selección sexual fue desprestigiada durante un tiempo.

En la actualidad la mayor parte de los biólogos aceptan este tipo de selección. Para que pueda existir deben cumplirse algunos requisitos en la población. El más importante es que haya una organización social diferente de la de tipo parental un padre y una madre, generalmente con mayor proporción de machos que de hembras o, en su defecto, que haya una clara condición de poligamia.

Radiación adaptativa

Debido a la constante competencia por el alimento y el espacio para vivir, cada grupo de organismo tiende a diseminarse y ocupar el mayor número posible de hábitat. Este proceso de evolución que parte de una sola especie ancestral y origina una variedad de formas que ocupan hábitat algo diferentes se denomina radiación adaptativa.

La esencia de la radiación adaptativa es la evolución a partir de una sola forma ancestral, dando una variedad de diferentes formas, cada una de las cuales está adaptada y especializada en cierta forma única a sobrevivir en un hábitat particular.¹

La radiación adaptativa que da origen a varios tipos de descendientes, adaptados a distintos ambientes, se denomina evolución divergente. El fenómeno opuesto, evolución convergente, ocurre también con bastante frecuencia; es decir, dos o más grupos poco relacionados pueden adquirir, al adaptarse a un medio ambiente similar, características más o menos similares. Por ejemplo, las alas han evolucionado no sólo en las aves, sino en los mamíferos –murciélagos–, en los reptiles –pterosaurios– y en insectos.

Especiación

La unidad de clasificación de plantas y animales es la especie. Una especie es una población de individuos con características estructurales y funcionales similares, que tienen un antecesor común y en la naturaleza sólo se aparean entre sí.

El problema de la definición de especie consistirá en observar cómo un determinado grupo de individuos pueden intercambiar caracteres hereditarios entre sí, y no con los demás, es decir, un sistema "abierto" de intercambio se convierte en un sistema "cerrado" que ya no intercambia con sus vecinos, formándose así una especie. El sistema es abierto si hay intercambio; las diferencias de caracteres hereditarios que puedan surgir se propagarán por todo el grupo. Si, por el contrario, el sistema es cerrado, cualquier cambio que ocurra no se transmitirá a los vecinos, y las diferencias se harán progresivamente mayores. Así las especies se van separando cada vez más.

La oposición a las ideas de Darwin

Ya se ha mencionado que muchas personas se opusieron a la idea de evolución y selección natural, sin embargo, los ataques a las ideas de Darwin que encontraron mayor eco no provenían de sus contrincantes científicos, sino de sus oponentes religiosos. Su teoría ya no necesita el protagonismo de un Dios creador y diseñador del universo ya que todos los aspectos del “maravilloso diseño” en la Naturaleza pueden explicarse a través de un proceso mecánico, algorítmico, conocido como selección natural y ejecutado sin que medie inteligencia o voluntad alguna. La idea de que todos los organismos hubiesen sido meticulosamente contruidos por un Gran Diseñador proporcionaba a la Naturaleza significado y orden, y al género humano una importancia que, aún hoy le gustaría poseer. La Biblia, en su relato de la creación, otorga esta situación de privilegio al Hombre, quien no sólo es el broche final que dio cierre a la obra de Dios, sino que fue hecho a imagen y semejanza del Creador para dominar el resto de lo creado². Sin embargo, los restos fósiles, tal cual lo

¹ Pueden observarse algunos ejemplos más adelante en “Anatomía comparada” –pág.12–.

² Al final de su libro escribía Darwin: “Mucha luz va a arrojarse sobre el origen del hombre y la historia de su evolución”. fueron sobre todo estas palabras las que hicieron que la Iglesia vertiera su ira contra Darwin, un hereje que ponía en descrédito el relato bíblico sobre la Creación.

advirtiera Darwin, presuponen un proceso de tanteo y una incapacidad de anticipar el futuro lo cual no concuerda con un Gran Diseñador eficiente, pero sí con uno de carácter más distante e indirecto.

Darwin conocía algunas pruebas fósiles y las utilizó para demostrar el hecho de la evolución, aún cuando los geólogos de su época no fueron capaces de adjudicar fechas exactas a dichos fósiles. Pero además utilizó otra prueba menos directa, aunque en muchos sentidos más convincente, para demostrar el hecho de que la evolución había tenido lugar. Las modificaciones que habían sufrido los animales y plantas domesticados eran una prueba persuasiva de que las variaciones evolutivas eran posibles, y de la eficacia del equivalente artificial del mecanismo de evolución propuesto por Darwin, la selección natural. Por ejemplo, la existencia de razas locales aisladas tiene una explicación fácil en la teoría de la evolución; la teoría de la creación sólo podría explicarlas si se asumen numerosos “focos de creación” esparcidos por toda la superficie terrestre. La clasificación jerárquica en la que se distribuyen de forma natural los animales y las plantas sugiere un árbol familiar; la teoría de la creación tiene que establecer suposiciones complejas y artificiales acerca de las ideas que cruzaban la mente del creador.

Finalmente se puede decir que la prueba de que el proceso de la evolución ha existido consiste en un gran número de observaciones detalladas que, en conjunto, adquieren sentido si asumimos la teoría de la evolución, pero que sólo podrían ser explicadas por la teoría de la creación si suponemos que el creador dispone cuidadosamente las pruebas para confundirnos.

Aunque menos encarnizada, Darwin también tuvo oposición en sus colegas científicos. En 1862, el eminente físico lord Kelvin inquietó a Darwin al demostrar en su calidad de autoridad, y hoy sabemos que se equivocó, que el Sol, y por tanto la Tierra, no podía tener una antigüedad superior a 24 millones de años. Aunque esta estimación era mucho más acertada que la fecha de 4004 a.C. que en aquel entonces apoyaba la Iglesia para la creación, no concedía el tiempo suficiente que necesitaba la evolución que Darwin proponía. Kelvin utilizó esta estimación y su inmenso prestigio científico como herramientas en contra de la teoría de la evolución. Su error estaba basado en la presunción de que el Sol liberaba calor mediante combustión, en lugar de por fusión nuclear, algo difícil de saber en aquella época.

Darwin también tuvo muchas dificultades surgida de las ideas que existían en aquella época sobre la naturaleza de la herencia. En el siglo XIX se asumía casi de forma universal que la herencia era un proceso combinado. En esta teoría, los descendientes no sólo tienen un carácter y apariencia intermedia, producto de la combinación de la de sus padres, sino que los factores hereditarios que transmiten a su propia descendencia son en sí mismo combinaciones intermedias debido a que se produce una irreversible fusión. Por ejemplo, la descendencia de un animal de pelo blanco con otro de pelo negro sólo podría ser gris, y su progenie también lo sería pues, al mezclar el material hereditario blanco y negro, éste ya no podría volver a separarse. Se puede demostrar que si la herencia es de tipo combinada es imposible que la selección natural darwiniana actúe, ya que la variación disponible se divide a la mitad en cada generación. Esto se expuso en 1867 y preocupó a Darwin lo suficiente como para conducirlo hacia el Lamarckismo. La solución al problema que tanto inquietó a Darwin ya se había expuesto, y descansaba en la teoría de la herencia particular desarrollada por Gregor Mendel¹

¹ El fraile austriaco Gregor Mendel es el padre de la moderna genética. Mediante la experimentación consiguió descubrir las leyes sobre la heredabilidad de diversas cualidades. (Ver “Las leyes de Mendel” –pág. 20–)

y publicada en 1865, pero que desafortunadamente no fue leída por Darwin, ni prácticamente por nadie, hasta después de su muerte.

Las huellas de la evolución

¿Cuáles son los indicios estudiados por los científicos que dan crédito a las teorías evolutivas? Existen muchas pruebas que dan testimonio sobre la evolución de la vida en nuestro planeta. Sin apartarme excesivamente del objetivo expuesto tempranamente en este capítulo, detallaré a continuación las evidencias más sobresalientes, procurando así, no dejar huecos importantes para el lector interesado.

Pruebas taxonómicas

Las especies se agrupan en categorías superiores, los géneros, y estos, a su vez, muestran semejanzas y se agrupan en familias, etc. Estas agrupaciones cada vez más amplias son las categorías taxonómicas.

La clasificación natural condujo de manera casi inevitable al planteamiento de la evolución, pues la explicación más clara que se podía dar a las semejanzas entre los diversos organismos era suponer que procedían de antepasados comunes a partir de los cuales han ido diferenciándose.

Todos los vertebrados son semejantes porque descienden de los mismos antepasados y conservan su tipo de organización. A partir de estos antepasados fueron apareciendo las distintas categorías, como, por ejemplo, las clases de las aves, reptiles, etc. A su vez, estos nuevos individuos continúan diferenciándose y dan origen a las categorías inferiores hasta llegar a la especie, en que las semejanzas de todos los individuos agrupados en ella son máximas. Cada categoría taxonómica significa, por tanto, un grado de parentesco.

Anatomía comparada

La anatomía comparada busca semejanzas homológicas entre los organismos para establecer parentesco evolutivo.

Las semejanzas homológicas son las estructurales, a diferencia de las analógicas que tienen que ver con la apariencia. Por ejemplo las aletas de los delfines y ballenas se asemejan mucho a la de los peces –semejanza analógica–, no obstante, si se observa su estructura interna se ve claramente cuánto difieren. El esqueleto de las aletas de las ballenas y los delfines sólo difiere en pequeños detalles del de cualquier mamífero terrestre típico. En cambio, el esqueleto de la aleta de los peces varía enormemente. Las semejanzas externas entre las aletas del pez y las de la ballena se explican fácilmente como una adaptación funcional a la natación. Pero esta semejanza no es la relevante si de establecer parentesco se trata.

Otro ejemplo, las alas de aves y murciélagos son análogas; sin embargo, las alas de los murciélagos y las extremidades pentadáctilas de los demás mamíferos son homólogas.

Según Cuvier –anatomista comparativo francés especializado en la reconstrucción de fósiles–, la homología muestra la pertenencia al "tipo estructural"; Sin embargo, estos hechos se explican con mayor claridad desde el punto de vista evolutivo.

El "tipo estructural" corresponderá ni más ni menos al tipo de organización que presentaba un antepasado común, y los diversos órganos homólogos se habrían ido diferenciando a partir de éste. Es posible suponer, en el caso citado, que los primeros mamíferos poseían una extremidad pentadáctila típica, a partir de la cual se fueron diferenciando los diversos tipos de

extremidades: las aletas en los mamíferos acuáticos, las alas en los murciélagos, la extremidad con un único dedo funcional en los caballos, etc. Estas modificaciones se producen como respuesta a la expansión de estos animales a los diversos medios: el mar, el aire, los grandes prados, etc.

En definitiva, la Anatomía comparada tiene como objeto el estudio de las homologías, y éstas vemos que sólo pueden interpretarse correctamente aceptando que los seres vivos descienden unos de otros.

Órganos vestigiales

Darwin utilizó como prueba de su teoría el hecho de que algunos órganos observados en adultos y embriones parecían ser sólo vestigios de lo que pudo haber sido un órgano funcional en algún antepasado. Por ejemplo, inmersos en la masa muscular, ciertas ballenas poseen una serie de huesos a los que no es posible atribuir ninguna función. Estudiados con detalle, se puede advertir su semejanza con la pelvis y los huesos de las extremidades posteriores del resto de los mamíferos; sin embargo, dado que las ballenas nadan gracias a su potente cola, estas estructuras óseas son completamente inútiles, lo que se manifiesta también por su falta de conexión con la columna vertebral.

Estos huesos son un claro ejemplo de lo que se denomina "órganos vestigiales" y constituyen los residuos de órganos normales en otras especies anteriores, y que al ocupar dichas especies nuevos ambientes en los que resultaban completamente inútiles experimentaron un proceso de degeneración.

En el caso citado de la ballena, los huesos de las extremidades posteriores nos confirman que ésta desciende de mamíferos terrestres tetrápodos y que sólo su adaptación a la vida marina ha hecho que las extremidades posteriores entraran en regresión.

El número de casos de órganos vestigiales es inmenso; sólo en el hombre se han catalogado casi un centenar. Entre los más conocidos está el apéndice vermiforme, que no tiene utilidad. En cambio, en otros mamíferos está mucho más desarrollado y actúa como un saco para que los fermentos puedan influir sobre la celulosa y otras sustancias cuya digestión requiere largo tiempo. Esto nos indica que el hombre desciende de mamíferos en cuya dieta se incluía alimentos que ya no aprovechamos. Otro caso curioso son los músculos que permiten mover las orejas –atrofiados en el hombre– y las muelas de juicio, totalmente funcionales en otros primates.

Pruebas embriológicas

Los peces asimilan el oxígeno disuelto en el agua por medio de branquias. Para hacer circular el agua poseen unas "hendiduras branquiales", que ponen en comunicación la faringe con el exterior. Si nos fijamos ahora en los anfibios y escogemos como ejemplo la rana, se observa que en la respiración, adaptada a aprovechar el oxígeno atmosférico, se sustituyen las branquias por pulmones y desaparecen las hendiduras branquiales. Sin embargo, en las fases larvianas, los renacuajos poseen, al igual que los peces, branquias y hendiduras branquiales.

Las tres clases restantes de vertebrados, es decir, reptiles, aves y mamíferos respiran por pulmones en todas sus fases de desarrollo. Ahora bien, los embriones de todas estas clases de animales presentan todos en diversos estadios hendiduras faríngeas que a lo largo del desarrollo del embrión, originarán diversos órganos que nada tienen que ver con la respiración; es como si todos los animales superiores pasaran, en uno u otro estadio de su desarrollo, por una fase pisciforme.

Muchos sistemas de órganos presentan fenómenos idénticos; las estructuras embrionarias o las fases de desarrollo se asemejan básicamente a la estructura de organismos más primitivos.

El embrión humano tiene un rudimento de cola como los demás mamíferos; las ballenas, que carecen de dientes en estado adulto, los presentan durante las fases embrionarias.

Como si el desarrollo de cada organismo fuera una especie de compendio de la línea evolutiva que le ha dado origen. Así, en los anteriores ejemplos, los reptiles, aves y mamíferos poseen hendiduras faríngeas porque todos estos grupos proceden de los peces a lo largo de la evolución; de la misma manera que los esbozos de dientes de los embriones de las ballenas nos indican claramente que proceden de antecesores provistos de dientes, como corresponde al tipo normal de mamíferos. Por último, la cola de nuestras primeras fases atestigua nuestro parentesco con los simios y demás mamíferos.

Esta teoría fue expuesta por primera vez por el anatomista del siglo XIX Ernest Haeckel y la llamó *recapitulación* –el embrión recapitula la secuencia evolutiva de sus antecesores– y ha conocido distintas fases de aceptación y rechazo entre los medios científicos. Las causas explicativas de la recapitulación pueden entenderse del siguiente modo. La selección natural opera en los individuos, y escasamente sobre los huevos o los fetos. Puede que el feto presente determinados rasgos, como las ya citadas hendiduras branquiales de los mamíferos, que perjudicarían la adaptación del individuo al medio tras su nacimiento, pero mientras no causen daños al feto y desaparezcan antes del parto, el organismo las tolera sin dificultad. Nuestras hendiduras branquiales no son vestigios de los antiguos peces, sino de los primitivos embriones de peces. Muchos órganos de nueva génesis se constituyen por transformación de órganos primitivos en otros nuevos como por ejemplo aletas que se transforman en patas o arcos branquiales que se transforman en huesos del oído.[Sagan_2]

Pruebas biogeográficas

El estudio de la distribución geográfica de las especies aportó una de las primeras pruebas de la evolución y fue el que indujo con mayor fuerza a Darwin a considerar que las especies podían haberse originado unas de otras por causas naturales.

Las aves de las islas Galápagos¹, por ejemplo, como la mayor parte de la fauna y la flora insular, se dice que son endémicas, es decir, características de un determinado lugar geográfico y, por tanto, de un área de distribución muy pequeña.

Los casos de endemismo sólo tienen una explicación racional desde un punto de vista evolutivo. En el caso de las islas, tal como lo describe Darwin, suponemos que son especies que se han diferenciado a partir de los escasos ejemplares de especies próximas del continente que lograron migrar hasta ellas de forma accidental.

En el caso de los continentes sólo puede haber dos explicaciones posibles. O bien se trata de una especie joven que todavía no ha aumentado su área de distribución o, en el caso más frecuente, corresponden a restos de una especie que en otras épocas poseía una distribución mucho más extensa y que, al cambiar las condiciones ambientales, se ha visto reducida a ocupar unas pequeñas áreas.

¹ Las Islas Galápagos, al Oeste de Ecuador, que Darwin visitó en su viaje alrededor del mundo, son un lugar legendario. Darwin demostró particular interés por los numerosos pinzones, que varían de isla a isla. Unos son granívoros y otros insectívoros. El estudio de estos pájaros fue decisivo para la elaboración de su teoría sobre la evolución.

La Paleontología demuestra que en la mayor parte de los casos las especies endémicas continentales corresponden a especies de amplia distribución en épocas anteriores.

El aislamiento geográfico¹ también es un factor importante. Tal es el caso de las islas de Oceanía que prematuramente se separaron del continente y en donde la vida evolucionó aislada de otras formas. Es por ello que la flora y fauna de estas islas sea tan peculiar y única en el mundo.

Pruebas bioquímicas

Los datos acerca de relaciones evolucionistas pueden obtenerse por similitudes y diferencias funcionales y químicas, así como similitudes y diferencias morfológicas. Por ejemplo, el grado de similitud entre las proteínas del plasma de varios animales se demuestra por la técnica antígeno-anticuerpo.

Miles de pruebas con diferentes animales han revelado la semejanza fundamental entre las proteínas sanguíneas de los mamíferos. El "parentesco sanguíneo" más allegado al ser humano es el de los grandes monos.

Las investigaciones referentes al orden sucesivo de aminoácidos en las cadenas alfa y beta de hemoglobinas de diferentes especies han revelado grandes similitudes y también diferencias específicas, cuyo tipo general revela el orden que debe haber presidido a las mutaciones subyacentes; es decir, cambios en los pares de bases de nucleótidos², durante la evolución.

Las relaciones evolutivas susceptibles de deducirse de estos estudios concuerdan en absoluto con las basadas en estudios anatómicos.

Pruebas genéticas

La selección y entrecruzamiento de los animales domésticos y de los vegetales cultivados durante los últimos milenios nos proporcionan modelos de cómo obran las fuerzas evolutivas. Todas las variedades de perros conocidas en la actualidad descienden de una o pocas especies afines de perro salvaje o lobo, a pesar de que varían en muchos aspectos.

Los expertos en botánica y agricultura lograron las presentes variedades de plantas cultivadas también por selección y cruzamiento a partir de antecesoras comunes. La col silvestre, por ejemplo, es la antecesora, no sólo de las especies cultivadas, sino de otras hortalizas tan diferentes como la coliflor, col de Bruselas, colirrábano y otras variedades regionales.

También se han logrado muchas variedades de trigo por selección, todas adaptadas a distintas condiciones ambientales. Así como a la sequía y a la resistencia contra parásitos.

¹ El concepto de aislamiento geográfico también resulta particularmente útil en el contexto de los algoritmos genéticos. Como se verá en el próximo capítulo la convergencia prematura de una población a un subóptimo (óptimo local) es un acontecimiento indeseable. La manera de evitarlo es asegurando la diversidad genética, que puede lograrse fácilmente si se divide la población en subpoblaciones aisladas, impidiendo que individuos de una isla interactúen con los de otra. Esta aproximación es utilizada en [Gomez97] y [Bruce01]. Varias formas de migración pueden ser implementadas entre las islas para permitir un intercambio genético controlado. (ver "Diversidad Genética y Convergencia" –pág. 64– y "Aislamiento geográfico" –pág. 73). En algunas estrategias neuroevolutivas desarrolladas en el presente trabajo también se utiliza el concepto de aislamiento geográfico –ESP, ESP con población de blueprints– (ver "Estrategias de Neuroevolución implementadas" en la pág. 194).

² Los nucleótidos son subunidades muy pequeñas que componen los ácidos nucleicos y están formados por un azúcar de cinco carbonos, ácido fosfórico y una base nitrogenada. Para más información ver el apartado "Los cromosomas" –pág.25–.

El número de cromosomas y la estructura que les corresponde en especies afines pueden estudiarse mediante procedimientos citológicos. Así se han conseguido pruebas útiles respecto a la evolución de la mosca de la fruta y muchos otros animales y vegetales.

Finalmente para concluir este apartado y con él esta modesta introducción al concepto de evolución biológica, comentaré que las pruebas a favor de la teoría evolucionista de Darwin no han cesado de acumularse desde que ésta fuera revelada a la comunidad científica. Hoy día es ampliamente aceptada a tal punto que el filósofo norteamericano Daniel Dennett afirma que cualquiera que dude de que la variedad de la vida en este planeta fue producida por un proceso de evolución es simplemente ignorante¹. Los científicos han logrado confirmar y clarificar cada vez más el paradigma darwiniano en la medida que fueron venciendo cada uno de los retos que intentaban cuestionarlo.

...Como lo razonaría Karl Popper, una idea que fuera en el fondo falsa, habría sucumbido muchas veces, ante el embate inmisericorde y concertado de muchos intentos de refutación a lo largo de siglo y medio de práctica científica.

...Darwin escribió su obra para resolver un problema biológico relativamente modesto: el origen de las especies. Para ello, tuvo que describir un proceso que llamó "selección natural", y que consiste en una secuencia de hechos completamente ciega o mecánica es decir, que no persigue ningún propósito y que, en efecto, es del todo incapaz de concebir ningún propósito o meta. Desde el punto de vista filosófico, sin embargo, la idea de selección natural resultó ser la contestación a una pregunta mucho más importante, a saber, cómo es que el diseño, la capacidad de proponerse metas, ha aparecido en el mundo.

Claudio Gutiérrez. "La selección natural"

En el intento por presentar un marco de referencia donde pueda apoyarse la aproximación a los Algoritmos Genéticos que realizaré en el próximo capítulo, la teoría de evolución por selección natural constituye sólo una cara de la moneda. La otra cara está representada por los avances de una ciencia relativamente joven, a saber, la Genética. A continuación dedicaré unas páginas a exponer los principios fundamentales de esta ciencia. Además, ya se ha hecho mención a la teoría de la herencia particular desarrollada por Gregor Mendel y aún merece una explicación. No sólo haré referencia a la genética mendeliana, sino también a la genética de poblaciones –conocida también como neodarwinismo– y la teoría cromosómica de la herencia, todos ellos, temas relevantes en este trabajo, pues su esencia está presente de forma directa o indirecta en la obra de Holland, el padre de los Algoritmos Genéticos.

Genética

Genética: estudio científico de cómo se transmiten los caracteres físicos, bioquímicos y de comportamiento de padres a hijos. Este término fue acuñado en 1906 por el biólogo británico William Bateson. Los genetistas determinan los mecanismos hereditarios por los que los descendientes de organismos que se reproducen de forma sexual no se asemejan con exactitud a sus padres, y las diferencias y similitudes entre padres e hijos que se reproducen de generación en generación según determinados patrones. La investigación de estos últimos ha

¹ Comentario recogido de Darwin's Dangerous Idea: Evolution and the Meanings of Life (1995) por Claudio Gutiérrez en [Gutierrez97_1]

dado lugar a algunos de los descubrimientos más importantes de la biología moderna.
[Encarta98]

Introducción histórica

Los antiguos egipcios ya sabían cómo producir frutos por fecundación artificial, mediante el cruzamiento de las flores masculinas de una palmera datilera con las femeninas de otras. Desde épocas muy remotas, el hombre ha manipulado animales domésticos y plantas.

Los filósofos griegos tenían varias ideas. Como ya se mencionó a comienzos de este capítulo en una nota al pie, las primeras teorías sobre la herencia fueron expuestas por Hipócrates (460 – 377 a.C.) quien especuló, que unas "semillas" se producían en diferentes partes del cuerpo y se transmitían a los hijos en el momento de la concepción. Aristóteles pensó que el semen masculino y el semen femenino –así se llamaba al flujo menstrual– se mezclaban en la concepción. Algunos pensaban que ni siquiera este tipo de mezclas eran necesarias, las formas "simples" –gusano, moscas...– nacían por generación espontánea. Así, los ratones se desarrollaban a partir de los granos húmedos y los piojos de las plantas se condensaban a partir de una gota de rocío. Esta teoría permaneció vigente hasta que en 1864 Pasteur – contemporáneo de Darwin y Mendel–, demostró que los microorganismos aparecían por causa del aire contaminado y no espontáneamente, como sostenían sus opositores.

Anton van Leeuwenhoek (1632–1723), descubre "animáculos" en el esperma humano y de otros animales. Algunos de los que miraban por los primeros microscopios soñaron ver un "pequeño hombrecillo" dentro de cada espermatozoide. Sostuvieron que la única contribución de la hembra para la próxima generación era proveer el ambiente para su desarrollo. En oposición la escuela de los ovistas creía que el futuro hombre estaba en el óvulo, y que el espermatozoide solo lo estimulaba.

La pangénesis sostenía la idea que machos y hembras forman "pangenes" en cada órgano. Estos "pangenes" se movían a través de la sangre a los genitales y luego a los recién nacidos. El concepto, originado en los griegos influyó a la biología hasta hace solo unos 100 años. Los términos "sangre azul", "consanguíneo", "hermano de sangre", "mezcla de sangre", y otros similares surgen de estos conceptos. Francis Galton, un primo de Charles Darwin, desechó experimentalmente la pangénesis.

La teoría de la mezcla suplantó a la de los espermistas y ovistas durante el siglo XIX. La mezcla de óvulos y espermatozoides daban como resultado la progenie que era una "mezcla" de las características de los padres. Las células sexuales se conocían colectivamente como gametos. De acuerdo con la teoría de la mezcla, cuando un animal de color negro se cruzaba con uno blanco la progenie debía ser gris y, a menudo, este no era el resultado. La teoría de la mezcla obviaba, entre otras cosas, explicar el salto de generación de algunas características.

Charles Darwin en su teoría de la evolución, se vio forzado a reconocer que la mezcla no era un factor –o al menos no el factor principal– y sugirió que la ciencia, no tenía la respuesta correcta al problema. La respuesta vino de un contemporáneo, un monje austriaco llamado Gregor Mendel, pero lamentablemente, como ya dijimos antes, Darwin nunca conoció el trabajo de Mendel.

A pesar de, o quizá debido a la descripción de gran número de cruzamientos experimentales, que le permitió expresar numéricamente los resultados obtenidos y someterlos a un análisis estadístico, el trabajo de Mendel no tuvo trascendencia alguna en los siguientes 35 años a partir de su publicación. Sólo obtuvo el debido reconocimiento en 1900, de manera más o menos independiente, por parte de tres investigadores, uno de los cuales fue el botánico

holandés Hugo de Vries, y sólo a finales de la década de 1920 y comienzos de 1930 se comprendió su verdadero alcance, en especial en lo que se refiere a la teoría evolutiva. Como resultado de años de investigación en el campo de la genética de poblaciones, se pudo demostrar que la evolución darwiniana podía describirse en términos del cambio en la frecuencia de aparición de pares de genes mendelianos en una población a lo largo de sucesivas generaciones.

Puede decirse que la ciencia de la genética nació en 1900, cuando se redescubrió el trabajo de Mendel en el que describía los patrones de la herencia en función de siete pares de rasgos contrastantes que aparecían en siete variedades diferentes de la planta del guisante –*Pisum sativum*–. Observó que los caracteres se heredaban como unidades separadas, y cada una de ellas lo hacía de forma independiente con respecto a las otras. Señaló que cada progenitor tiene pares de unidades pero que sólo aporta una unidad de cada pareja a su descendiente. Más tarde, las unidades descritas por Mendel a las cuales él denominó “Elemente” recibieron el nombre de “genes”.

Mendel es el verdadero fundador de la genética y sus experimentos de hibridación realizados en el jardín de un monasterio del que llegó a ser abad, llevaron a una nueva comprensión del mecanismo de la herencia biológica y al nacimiento de la genética como ciencia. Así, el concepto de mezcla fue reemplazado por el concepto de unidad.

A partir del redescubrimiento del trabajo de Mendel, y gracias a los avances en microscopía, en algunas disciplinas como la citología, se realizaron grandes progresos; de tal forma que durante este período, se descubrieron los cromosomas y también se observaron sus primeros movimientos durante la división celular –mitosis–, así como el proceso por el cual se forman los gametos, que son los portadores de la información genética que se trasmite a la siguiente generación.

En 1902, Boveri y Sutton se percataron de la existencia de un estrecho paralelismo entre los principios mendelianos recién redescubiertos y la conducta de los cromosomas en la meiosis¹, propiciando así una colaboración cada vez más estrecha entre la genética y la citología.

En 1906 Bateson bautiza a la nueva ciencia de la herencia como genética, quedando así acuñado definitivamente el nombre y viendo la luz el primer libro de texto sobre el tema.

Los experimentos realizados en la mosca de la fruta –*Drosophila melanogaster*– llevados a cabo por Thomas Hunt Morgan y sus colaboradores, ayudaron a establecer la teoría cromosómica de la herencia. Estos investigadores propusieron que los factores hereditarios se disponían de forma lineal en los cromosomas. Posteriores investigadores contribuyeron a la afirmación de que los genes están en los cromosomas y, por tanto, los genes que están en el mismo cromosoma tienden a heredarse juntos, por lo que se denominan genes ligados.

Posteriormente, se aceptó que los genes son los responsables de la síntesis de proteínas, sosteniendo la teoría “un gen–una enzima”, y por tanto, al alterar la secuencia de nucleótidos de un gen faltaría una enzima determinada².

¹ La meiosis y la mitosis son dos tipos de división celular. Mientras que las células somáticas se dividen por mitosis, las sexuales se originan a través de la meiosis. Para más información ver el apartado “La meiosis” –pág. 26–.

² Una enzima es una proteína que interviene en distintas reacciones químicas indispensables para la vida acelerándolas o haciéndolas posibles. A esta función se la llama catalizadora o enzimática.

En 1953, Watson y Crick descubren la estructura del ADN, que representa el soporte del material hereditario. A partir de este momento se realizan numerosos estudios sobre replicación, biosíntesis de proteínas, biosíntesis de ARN, etc. Posteriormente, Francis Jacob y Jaques Monod demostraron la existencia de un sistema de regulación genética, en el que intervienen los denominados genes estructurales, genes reguladores y genes operadores.

El aporte de los conocimientos genéticos combinados con los de otras ciencias, como la bioquímica y la fisiología, han permitido avances importantísimos en la comprensión del fenómeno de la vida y de la evolución de los seres vivos. La genética molecular y la ingeniería genética están haciendo posible modificar a voluntad el material genético de un organismo vivo.

El denominado “Proyecto Genoma Humano” iniciado en 1990 por numerosas instituciones con el fin de obtener el mapa genético completo de cada uno de nuestros cromosomas, representa una ambiciosa iniciativa que ha comenzado a arrojar importantes resultados. En abril del año 2000 ya se había conseguido secuenciar el genoma completo, formado por 50 millones de fragmentos de ADN. Ahora resta la tarea más difícil, que consiste en ensamblar esa enorme cantidad de fragmentos.

Genética mendeleana

Hoy día existen varias ramas de la genética, por ejemplo la genética cuantitativa, genética humana, genética de poblaciones, genética fisiológica, citogenética, genética bioquímica, etc. En particular existe una conocida como genética mendeleana que realiza un estudio científico de la herencia, en relación y concordancia con los descubrimientos de Mendel. Hoy siguen vigentes muchas de sus afirmaciones, si bien se han encontrado excepciones y variantes.

Recordemos que Mendel no conoció las estructuras biológicas responsables de la herencia; se conocieron años después, cuando la teoría cromosómica de la herencia vino a afirmar la responsabilidad de los genes en la transmisión de los caracteres hereditarios, y la localización de dichos genes en el interior de los cromosomas del núcleo celular. La relación que existe entre genes y cromosomas se verá más adelante (“Los cromosomas” –pág. 25–), por el momento no es necesaria para comprender los principios establecidos por Mendel.

A continuación resumiré los conceptos básicos de la genética mendeleana, cabe destacar que ésta es una disciplina moderna y ciertas nociones son posteriores a Mendel.

- El *gen* –elemento para Mendel– es la unidad estructural y funcional de la herencia transmitida de padres a hijos a través de los gametos, y regulan la manifestación de los caracteres heredables.
- Los *alelos* son distintas formas de un mismo gen. Si hay muchos, se tiene una serie alélica.
- Un *locus* es el lugar que ocupa un determinado gen dentro de un cromosoma.
- La constitución genética de un organismo se conoce como su *genotipo*; y sus características externas observables se conocen como su *fenotipo*.
- *Alelo dominante* y *alelo recesivo* son dos formas de un mismo gen que codifican para caracteres determinados. El alelo dominante es el que, estando en uno solo de los cromosomas de cada pareja de un individuo, se manifiesta en el fenotipo de éste. El recesivo es el alelo que debe estar presente en los dos cromosomas para que se manifieste en el fenotipo el carácter para el que codifica.

- *Individuo homocigótico* para un gen determinado es el que tiene el mismo alelo repetido en los dos cromosomas homólogos¹ –los dos del tipo dominante o los dos del tipo recesivo–.
- *Individuo heterocigótico* para un gen determinado es el que tiene distintos alelos en los cromosomas homólogos.

Las leyes de Mendel

Mendel utilizó el guisante, *Pisum sativum*, para sus experimentos de cruzamientos y comenzó por cultivar unas cuantas variedades en el jardín de su monasterio. No pretendo entrar aquí en demasiados detalles sobre los experimentos de Mendel por considerar que escapan a los objetivos de este capítulo, pero sí realizaré una breve descripción de sus observaciones y de la formulación actual de sus leyes.

Primer experimento y primera ley de Mendel

Al efectuar cruzamientos entre plantas que diferían en uno sólo de entre siete caracteres estudiados, homocigotas para este carácter, Mendel observó que el híbrido heredaba la característica de uno sólo de sus progenitores.

Mendel llamó *dominante* al carácter que pasaba sin cambios a la generación híbrida o primera generación filial F1, y *recesivo* al carácter que no se manifestaba en esa primera generación.

De este hecho se deduce la primera ley de Mendel, llamada *ley de la uniformidad*, que se puede formular de la siguiente manera:

"Todos los descendientes del cruce entre dos razas puras –ambos homocigotos para un determinado carácter– son iguales entre sí".

Mendel llegó a esta conclusión trabajando con plantas de guisantes que producían semillas amarillas y con una variedad que producía semillas verdes. Al hacer un cruzamiento entre estas plantas, obtenía siempre plantas con semillas amarillas.

Interpretación del experimento: Se simboliza con letras mayúsculas a los alelos dominantes para un determinado gen, y con minúscula a los recesivos. El polen de la planta progenitora aporta a la descendencia un alelo para el color de la semilla, y el óvulo de la otra planta progenitora aporta el otro alelo para el color de la semilla; de los dos alelos, solamente se manifiesta aquél que es dominante (A), mientras que el recesivo (a) permanece oculto.

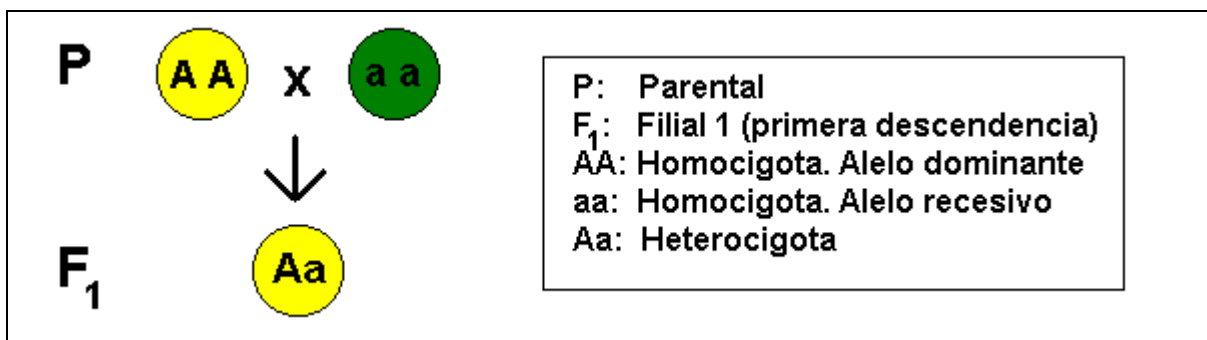


Ilustración 1 Interpretación del primer experimento de Mendel

¹ Los cromosomas homólogos tienen la misma estructura, pero suelen tener distintos alelos pues proceden de progenitores diferentes. Para más información ver "Los cromosomas" –pág. 25–.

La primera ley de Mendel se cumple también para el caso en que un determinado gen de lugar a una herencia intermedia –una mezcla en el fenotipo–, como es el caso del color de las flores del "dondiego de noche" –*Mirabilis jalapa*–. Al cruzar las plantas de la variedad de flor blanca con plantas de la variedad de flor roja, se obtienen plantas de flores rosas. La interpretación es la misma que en el caso anterior, solamente varía la manera de expresarse los distintos alelos. Estos resultados se producen porque ninguno de los alelos ejerce una dominancia completa sobre el otro.

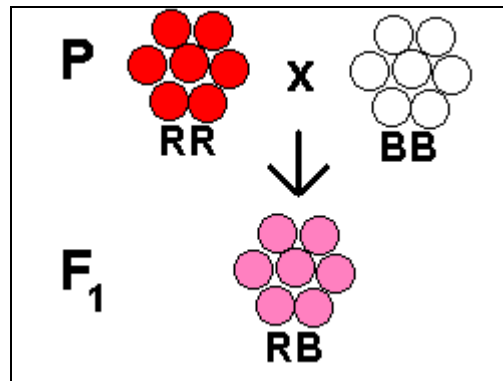


Ilustración 2 Herencia intermedia. Ningún alelo es completamente dominante. No obstante sigue cumpliéndose la primera ley de Mendel.

Es importante observar que la mezcla se produce en el fenotipo, y no en los alelos que se mantienen diferenciados. La transmisión de los alelos en forma diferenciada es deducida por Mendel a partir del siguiente experimento.

Segundo experimento y segunda ley de Mendel

La segunda generación filial F₂ se obtuvo por autofecundación de las plantas de la F₁ del primer experimento, y ya no era uniforme, sino que presentaba caracteres dominantes y recesivos en proporción 3:1. Así pues, aunque el alelo que determina la coloración verde de las semillas parecía haber desaparecido en la primera generación filial, vuelve a manifestarse en esta segunda generación.

A partir de estos resultados, dedujo Mendel la segunda ley, llamada *ley de la segregación* o de la disyunción, que se resume así:

"Los dos factores hereditarios que informan para un mismo carácter, no se fusionan o mezclan, sino que permanecen diferenciados durante la vida del individuo y se segregan, es decir, se separan y se reparten, en el momento de la formación de los gametos".

Los dos alelos distintos para el color de la semilla presentes en los individuos de la primera generación filial, no se han mezclado ni han desaparecido, simplemente ocurría que se manifestaba sólo uno de los dos.

Cuando el individuo de fenotipo amarillo y genotipo Aa, forme los gametos, se separan los alelos, de tal forma que en cada gameto sólo habrá uno de los alelos y así puede explicarse los resultados obtenidos. Al recombinarse al azar en la generación F₂ se forman cigotos de la siguiente forma AA, Aa y aa con probabilidades respectivas de 0.25, 0.50 y 0.25, lo que se traduce en la relación 3 : 1 de semillas amarillas: semillas verdes.

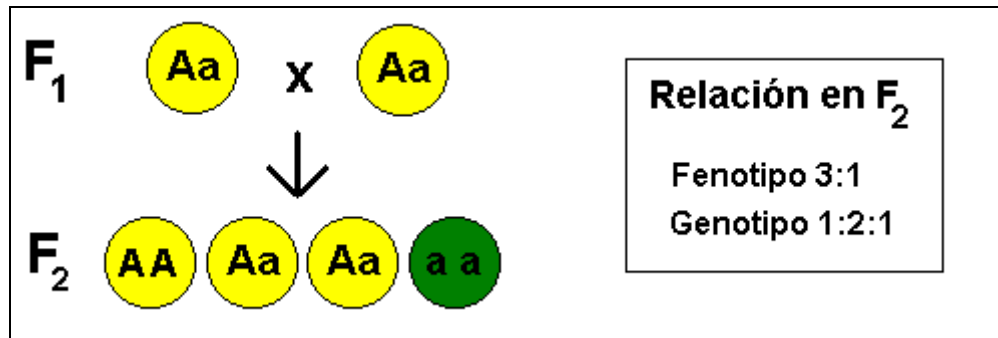


Ilustración 3 Interpretación del segundo experimento de Mendel.

En el caso de los genes que presentan herencia intermedia, también se cumple el enunciado de la segunda ley. Siguiendo el ejemplo de las flores del "dondiego de noche", si se toman dos plantas de flores rosas de la primera generación filial (F1) y se cruzan entre sí, se obtienen plantas con flores blancas, rosas y rojas, en la proporción 1:2:1 –tanto en el fenotipo como en el genotipo–. También en este caso se manifiestan los alelos para el color rojo y blanco, que permanecieron ocultos en la primera generación filial.

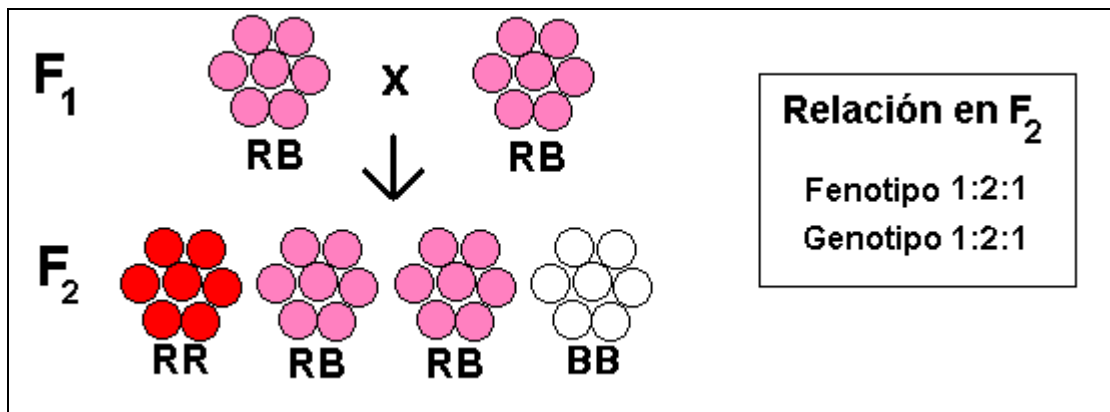


Ilustración 4 Herencia Intermedia. Sigue siendo válida la segunda ley de Mendel.

Tercer experimento y tercera ley de Mendel

Mendel cruzó plantas de guisantes de semilla amarilla y lisa con plantas de semilla verde y rugosa –Homocigóticas ambas para los dos caracteres–.

Las semillas obtenidas en este primer cruzamiento eran todas amarillas y lisas, cumpliéndose así la primera ley para cada uno de los caracteres considerados.

Estas plantas de la F1 se cruzan entre sí. Los resultados obtenidos para cada uno de los caracteres considerados por separado, responden a la segunda ley.

Basándose en estos experimentos, se deduce la tercera ley de Mendel llamada **ley de la independencia de los factores hereditarios**, que puede expresarse de la siguiente manera:

"Los factores hereditarios no antagónicos mantienen su independencia a través de las generaciones, agrupándose al azar en los descendientes".

Los resultados de los experimentos de la tercera ley refuerzan el concepto de que los genes son independientes entre sí, que no se mezclan ni desaparecen generación tras generación. Para esta interpretación fue providencial la elección de los caracteres, pues estos resultados no

se cumplen siempre, sino solamente en el caso de que los dos caracteres a estudiar estén regulados por genes que se encuentran en distintos cromosomas. Si por el contrario, los genes se ubican en el mismo cromosoma reciben el nombre de ligados¹ y suelen heredarse juntos.

Teoría de la genética de poblaciones –neodarwinismo–

Los estudios de Mendel demostraron lo que Darwin insinuó vagamente en cierta época, que la herencia es particular, no combinada. Sean o no los descendientes, formas intermedias entre sus dos padres, ellos heredan y transmiten partículas hereditarias separadas; que hoy en día se denominan genes. Esto marca la diferencia de la plausibilidad matemática de la teoría de la selección natural. Si la herencia es particular, la selección natural puede actuar.

La versión moderna del darwinismo, denominada teoría de la genética de poblaciones, fue elaborada entre los años 1920 y 1930 por los genetistas R. A. Fisher, J. B. S. Haldane y Sewall Wright, y consolidada con posterioridad en la década de los años cuarenta en la síntesis conocida como Neodarwinismo.

Esta teoría puede resumirse en lo siguiente: los genes de una población de animales o plantas que se entrecruzan sexualmente constituyen un “conjunto” de genes. Los genes compiten en este “conjunto”. En la práctica, la vida de los genes transcurre o asentándose en cuerpos individuales que ellos ayudan a construir, o transmitiéndose de un cuerpo a otro a través del espermatozoide o del óvulo en el proceso de la reproducción sexual. Ésta mantiene los genes mezclados y el hábitat a largo plazo de los genes es el “conjunto” genético. Cualquier gen que se origina en él es resultado de una mutación u error aleatorio en el proceso de copia de los genes. Una vez que se ha producido una mutación nueva, ésta puede extenderse a través del “conjunto” genético por medio de la mezcla sexual.

Como establecieron por primera vez el matemático británico G. H. Hardy y el científico alemán W. Weinberg en 1908, no existe una tendencia propia de los genes a desaparecer del “conjunto” de genes, salvo que actúen fuerzas como la selección natural, mutación o deriva genética. Por consiguiente, se puede decir que en poblaciones grandes, si no hay selección a favor de un alelo específico, si no ocurren mutaciones, si no hay migraciones y el apareamiento es al azar entonces se cumple que las frecuencias relativas de los genes en la población se mantendrán constantes a través de las generaciones. Esto es una consideración estadística que sólo tiene efecto para las poblaciones muy numerosas, si la población es pequeña, en ausencia de fuerzas selectivas, es posible aún que las frecuencias de los distintos genes varíen ampliamente de generación en generación por un simple efecto del azar –hechos fortuitos–, y que la población acumule unos u otros genes independientemente de su valor selectivo, a este efecto o fuerza se lo conoce con el nombre de deriva genética.

En esta teoría la evolución es considerada como el resultado de varias fuerzas que actúan sobre las poblaciones: la mutación, la selección, la deriva genética y la migración, produciendo un cambio gradual en las frecuencias de genes de una población.

Ya se ha mencionado que la mutación fue el primer mecanismo de variación genética que existió en la Naturaleza. La reproducción sexual y la recombinación genética debida al

¹ Para más información ver el apartado “Ligamiento” –pág. 37–.

cruzamiento¹, fue el segundo, y permitió que la variación genética se distribuya con mayor rapidez en el “conjunto” genético.

La selección natural es fundamental para explicar la mejora de la adaptación, la compleja organización funcional de la vida y aquello que puede definirse como evolución. La dotación genética de los organismos influye sobre su propio desarrollo. Algunos tienen mejores cualidades para sobrevivir y reproducirse que otros. Los organismos que son buenos, es decir aquellos cuyas características para sobrevivir y reproducirse son mejores, tenderán a aportar más genes a los “conjuntos” genéticos del futuro que aquellos cuyas características sean malas para estos fines: los genes que tienden a formar organismos buenos serán predominantes en los “conjuntos” genéticos. La selección natural se traduce en el distinto éxito que alcanzan los organismos en la supervivencia y reproducción: esto es importante debido a las consecuencias que supone para la supervivencia de los genes en el “conjunto” genético.

No todas las muertes selectivas conducen a cambios evolutivos. Por el contrario, la mayor parte de la selección natural se denomina selección estabilizadora, por cuanto que elimina genes del “conjunto” genético que tienden a producir desviaciones de una forma que ya es óptima. Pero cuando las condiciones del medio cambian, bien por una catástrofe natural o por una evolución más perfecta de otras criaturas –depredadores, víctimas, parásitos y otros–, la selección puede conducir a una variación evolutiva.

Teoría cromosómica de la herencia

Es sobre todo en esta teoría en la que se han inspirado los Algoritmos Genéticos y por ello los conceptos aquí desarrollados adquieren relevancia en relación a los objetivos propuestos en este capítulo.

Las aparentes excepciones de la tercera ley de Mendel, que estudiaron investigadores como Bateson y Punnett, y que parecían apuntar hacia la idea que ciertos caracteres no se heredan de manera totalmente independiente, encontraron su explicación en la primera década del siglo XX, cuando Thomas H. Morgan elaboró la teoría cromosómica de la herencia.

En 1902, dos investigadores por separado, Sutton en Estados Unidos y Boveri en Alemania, tras observar el paralelismo que existe entre la herencia de los factores hereditarios y el comportamiento de los cromosomas durante la meiosis y la fecundación, propusieron que los factores hereditarios estaban en los cromosomas. Esta afirmación constituye la base de la teoría cromosómica de la herencia, que puede resumirse en los siguientes términos:

Los genes están en los cromosomas, su disposición es lineal, uno detrás del otro, y mediante el entrecruzamiento de cromátidas homólogas se produce la recombinación de los genes.

En los seres procariontes –virus, bacterias y algas verdeazuladas–, la información genética está contenida en forma de una molécula de ADN o ARN desnuda, no separada del citoplasma por ninguna membrana y a la que no se asocian proteínas. Este tipo de material genético se llama genóforo, para diferenciarlo del de seres eucariontes, en los que se encuentra organizado en forma de varias piezas llamadas cromosomas, que contienen ácido nucleico acompañado de proteínas, y que están dentro de un verdadero núcleo rodeado de una membrana nuclear.

¹ Más adelante se expondrá el concepto de cruzamiento (crossover o crossing over) al definir teoría cromosómica de la herencia y con más detalle en el apartado “La meiosis” –pág. 26–. Debe mencionarse que el crossover ha sido tomado por los Algoritmos Genéticos como un operador de suma importancia.

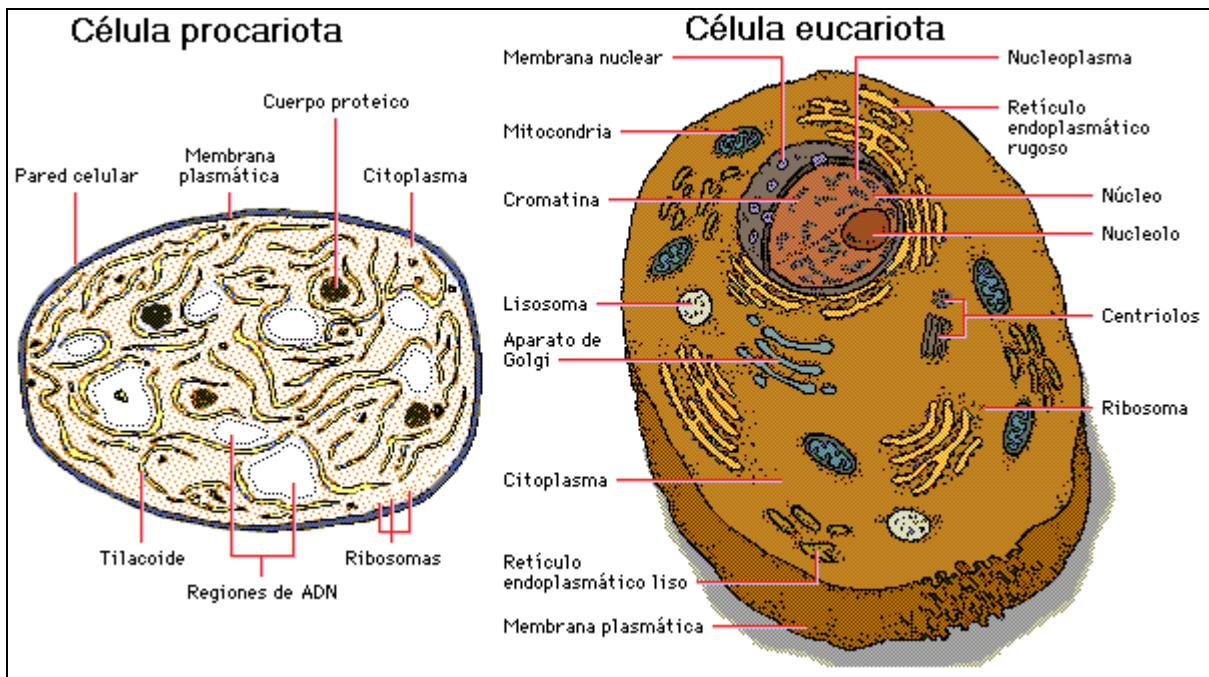


Ilustración 5 Diferencias entre las células procariotas y eucariotas.

Los cromosomas

Recibieron este nombre por su capacidad para ser teñidos con ciertos colorantes. Su estructura y composición, más detalladas, se han podido averiguar gracias a la aparición de técnicas como microscopía electrónica y procedimientos de biología molecular.

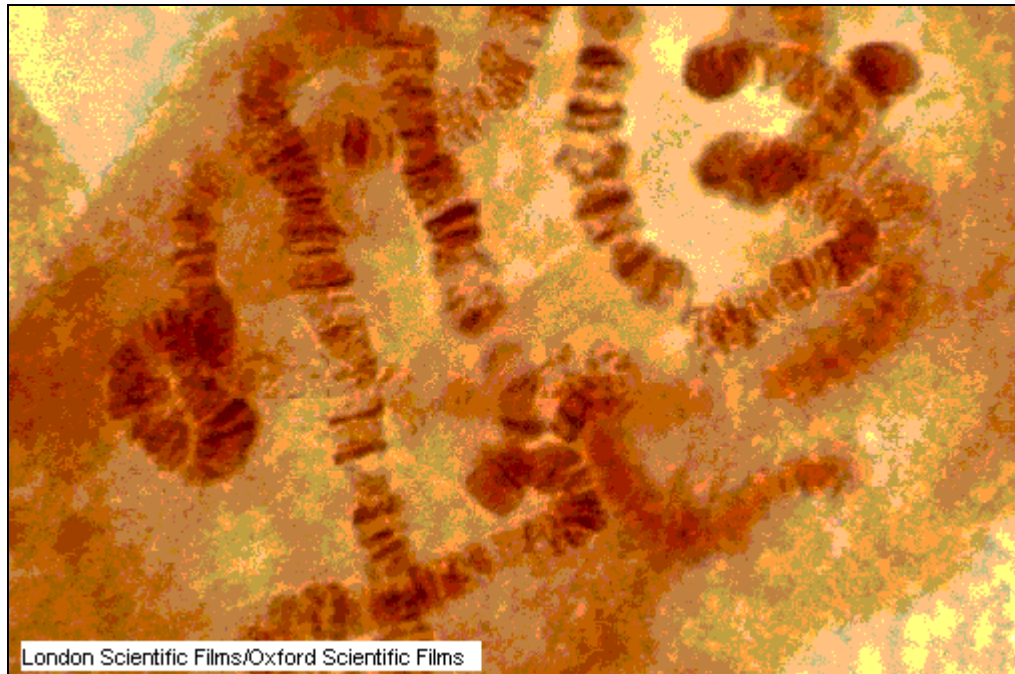


Ilustración 6 Foto extraída de [Encarta98]

El material del que están hechos los cromosomas es la cromatina, compuesta básicamente de un enrollamiento en hélice de ADN –ácido desoxirribonucleico– y proteínas. Debido a que el

ADN de cada cromosoma es una molécula continua, alargada, simple y delgada, los genes deben ser parte de ella; y como es una cadena de subunidades muy pequeñas que se conocen por nucleótidos¹, cada gen incluye muchos nucleótidos y su secuencia determina las propiedades del gen. El mapeo de los diferentes cromosomas ha servido para localizar los genes y para saber en donde se contiene la información para los distintos caracteres de un organismo.

Las células de los individuos de una especie determinada suelen tener un número fijo de cromosomas, que en las plantas y animales superiores se presentan por pares. El ser humano tiene 23 pares de cromosomas. En estos organismos, las células reproductoras tienen por lo general sólo la mitad de los cromosomas presentes en las corporales o somáticas. Durante la fecundación, el espermatozoide y el óvulo se unen y reconstruyen en el nuevo organismo la disposición por pares de los cromosomas; la mitad de estos cromosomas procede de un parental, y la otra mitad del otro. Los cromosomas del mismo par se denominan cromosomas homólogos –uno de cada progenitor– y juntos definen las características heredadas tal como se explicó en el apartado “Las leyes de Mendel” –pág. 20–.

El estudio de los cromosomas puede realizarse desde dos puntos de vista:

Externo, en el que se consideran la forma, el tamaño y el número de cromosomas en metafase –fase de la meiosis explicada luego–, por ser el estado de máxima condensación durante el ciclo celular y, consecuentemente, cuando se nos revelan las más importantes propiedades de los cromosomas.

Interno, que hace referencia a la organización estructural íntima del cromosoma y más correctamente al conocimiento de la organización supramolecular y diferenciación longitudinal del mismo.

La meiosis

La meiosis y la mitosis son dos tipos de división celular. Mientras que las células somáticas se dividen por mitosis, las sexuales se originan a través de la meiosis. Se puede considerar a la meiosis como un proceso de maduración celular sexual, en el que una célula diploide – n pares de cromosomas homólogos– forma cuatro células haploides – n cromosomas–. A través de la meiosis se producen los gametos en los organismos animales y esporas en los vegetales.

Las fases de la meiosis consisten en dos divisiones nucleares sucesivas que se designan meiosis I y meiosis II. En el contexto de este trabajo, resulta particularmente importante la primera división ya que allí es cuando ocurre la recombinación –crossover, o crossing over– del material genético, y constituye uno de los mecanismo más representativos tomado por la teoría de los Algoritmos Genéticos.

En la fase de reposo, la cromatina –la materia prima de los cromosomas– es amorfa y se encuentra dispersa al azar por todo el núcleo. Sin embargo, cuando las células se preparan para su división, la cromatina se condensa y se agrupa en un número de cromosomas bien definido.

En la *interfase*, etapa previa a la meiosis, cada cromosoma consistente en un único filamento llamado cromátida, se replica transformándose en dos cromátidas idénticas unidas entre sí por

¹ Cada nucleótido está formado por un azúcar de cinco carbonos, ácido fosfórico y una base nitrogenada. En cada cadena existen cuatro tipos diferentes de bases –adenina, guanina, citosina y timina– para más información ver “Genes y cromosoma” –pág. 32–

el centrómero y llamadas cromátidas hermanas. Es esta la forma más conocida de los cromosomas y con la que habitualmente se esquematizan.

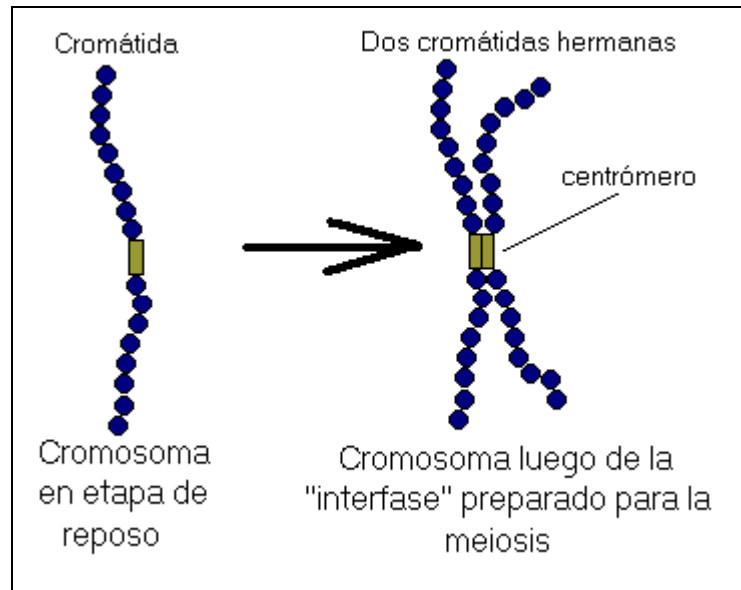


Ilustración 7 Esquema de un cromosoma antes y después de la interfase.

Primera división meiótica: Meiosis I

Se desarrolla a través de cuatro etapas denominadas profase, metafase, anafase y telofase; se las designa a todas con I para indicar que pertenecen a la meiosis I.

Profase I. Es una etapa crucial, varios eventos significativos ocurren aquí. El ADN comienza a torcerse y condensarse, haciéndose visible al microscopio. En esta etapa los cromosomas homólogos se aparean. Recuérdese que la carga genética de los cromosomas homólogos provienen de distintos parentales. En vista de que cada cromosoma consiste en dos cromátidas idénticas, en el apareamiento de los cromosomas homólogos en realidad intervienen cuatro cromátidas; este complejo constituido por un par de cromosomas homólogos se conoce como *cromotétada*. El punto donde dos cromátidas de distintos cromosomas –no hermanas– se entrelaza se llama *chiasma*. En este momento tiene lugar algunas veces, un entrecruzamiento –*crossing over*–, que es el intercambio de segmentos de un cromosoma con los segmentos correspondientes de su homólogo. Estos intercambios de material cromosómico son importantes fuentes de variación hereditaria.

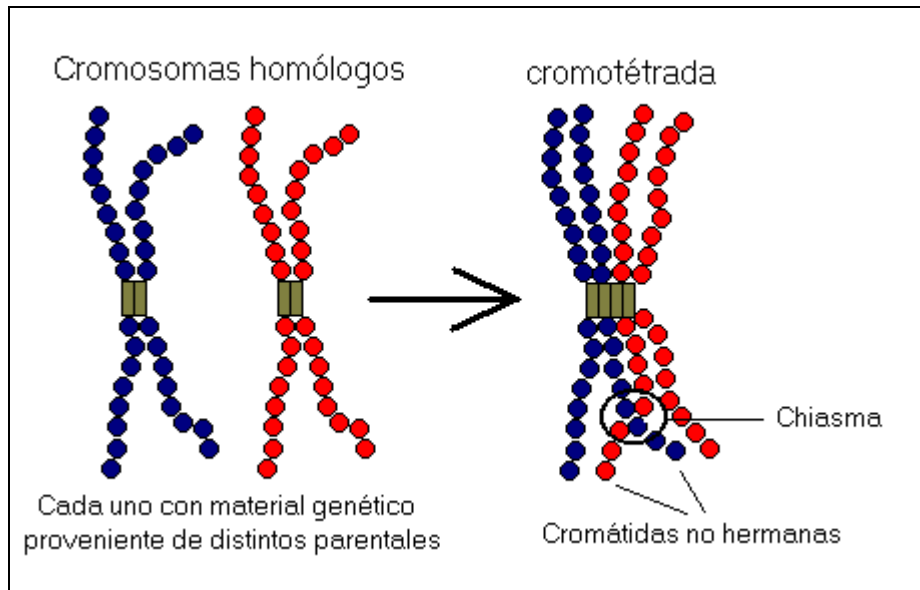


Ilustración 8 Meiosis. Profase I.

Hacia el final de la profase I los centriolos –un tipo de organoide de la célula– se duplican y comienzan a separarse, dirigiéndose hacia los polos. Se forma entre ellos el huso acromático, la membrana nuclear se hace más difusa y desaparece el núcleo.

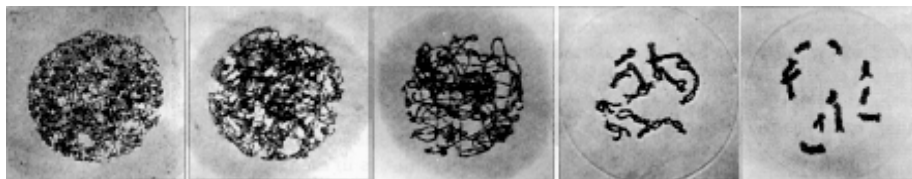


Ilustración 9 fotografías obtenidas de pág. Web <http://www4.ncsu.edu/unity/users/b/bnchorle/www/proi.htm>

Metafase I. En esta etapa los pares de homólogos se disponen en el plano ecuatorial de la célula. Cada cromosoma ha alcanzado su máxima densidad y cuando mejor se observan al microscopio. La región del centrómero –constricción del cromosoma que mantiene a las cromátidas unidas– se ha duplicado y las fibras del huso se asocian con los cinetocoros –región proteica del centrómero–.

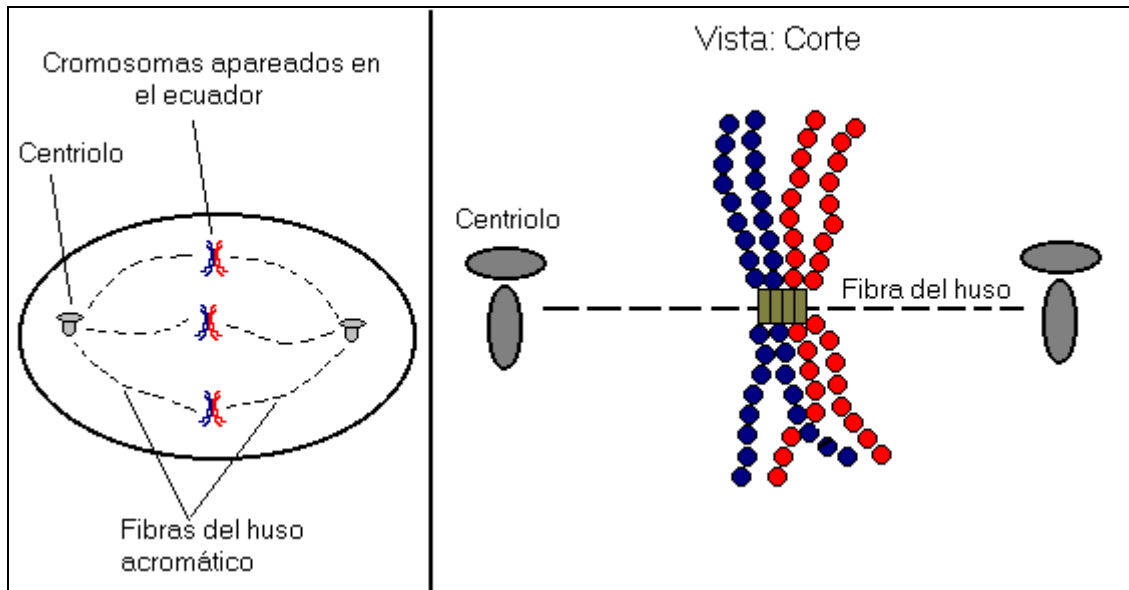


Ilustración 10 Meiosis. Metafase I.

Anafase I. Durante esta etapa los homólogos, cada uno formado por dos cromátidas hermanas, se separan como traccionados por las fibras del huso y se mueven hacia polos opuestos. A diferencia de la mitosis¹, las dos cromátidas hermanas de cada cromosoma no se separan.

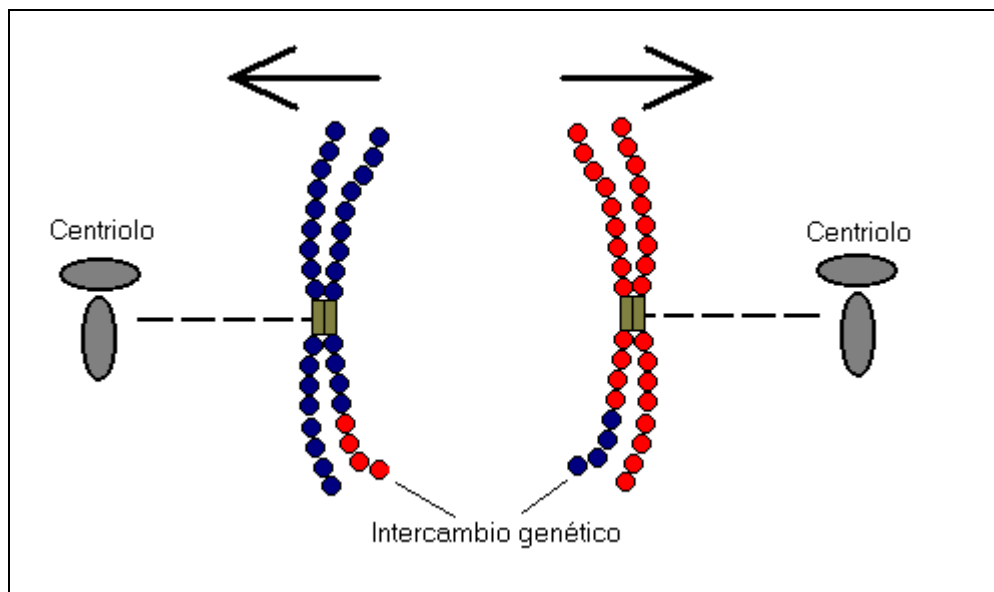


Ilustración 11 Meiosis. Anafase I.

Telofase I. En la última fase de la primera división meiótica, los homólogos se han trasladado a los polos. Ahora cada grupo sólo contiene la mitad de los cromosomas que había en el núcleo original, pero cada uno con dos cromátidas unidas por el centrómero. Además, estos cromosomas pueden ser distintos de cualquiera de los que había en la célula original, a causa

¹ No se expondrán en este trabajo los detalles de la división celular conocida como mitosis pues son ajenos al objetivo de este capítulo, simplemente debe considerarse que por medio de ella una célula se divide en dos células hijas con exactamente la misma información genética (no hay recombinación genética ni cambio en el número de cromosomas).

de los intercambios que tienen lugar durante el entrecruzamiento. Se forman las membranas nucleares y el citoplasma se adelgaza por el medio hasta estrangularse completamente formando así dos células separadas. Cada una de estas células realizará la segunda división meiótica.

Segunda división meiótica: Meiosis II

Al comienzo de esta división los cromosomas dispersos se condensan nuevamente. Consta de cuatro etapas: Profase II, metafase II, Anafase II y telofase II. Siguiendo el ejemplo

Profase II. En esta etapa las envolturas nucleares se desintegran, los centriolos vuelven a duplicarse y comienzan a separarse moviéndose hacia los polos. Se forma entre ellos el huso acromático, la membrana nuclear se hace más difusa y desaparece el núcleo.

Metafase II. Durante esta etapa los cromosomas –formados ahora por dos cromátidas no necesariamente iguales unidas por un centrómero– se alinean en el plano ecuatorial y las fibras del huso se vuelven a asociar con los cinetocoros

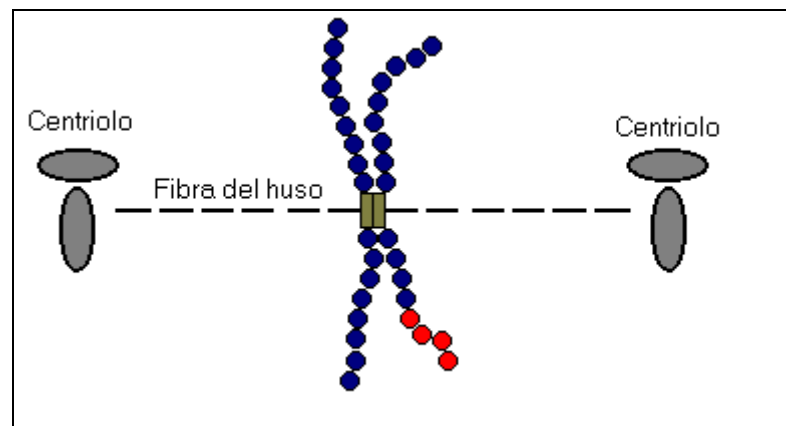


Ilustración 12 Meiosis. Metafase II.

Anafase II. Es la etapa meiótica en la que, al igual que en la anafase mitótica, las cromátidas se separan y cada cromosoma, compuesto ahora por una sola cromátida, se desplaza hacia uno de los polos.

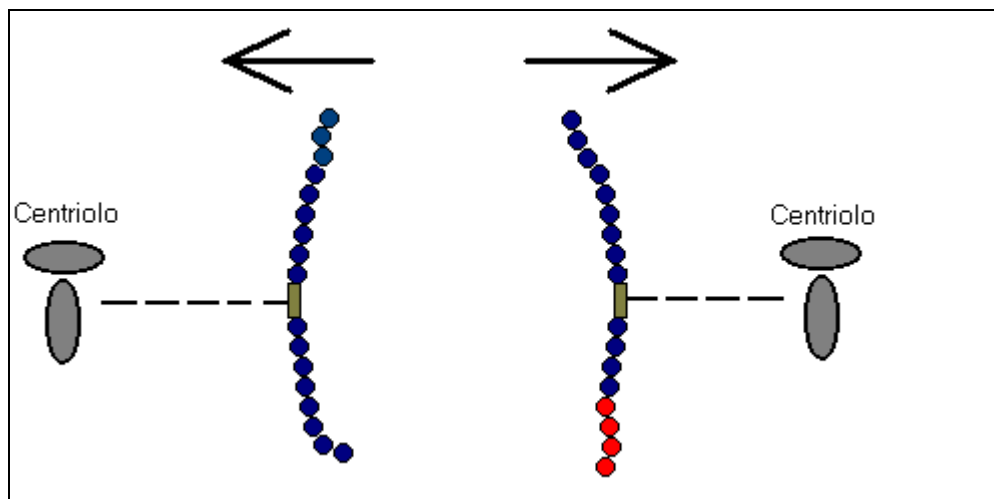


Ilustración 13. Meiosis. Anafase II.

Telofase II. En la telofase II los husos desaparecen y se forma una envoltura nuclear alrededor de cada conjunto de cromosomas. Ahora hay cuatro núcleos en total y cada uno contiene el número haploide de cromosomas. Se produce entonces la división citoplasmática –

citocinesis– de la misma manera que después de la mitosis, y las células empiezan a diferenciarse en esporas –en las plantas– o en gametos –en los animales–.

Por lo tanto, si comenzó el proceso con una célula que tenía 46 cromosomas –23 pares de homólogos, como ocurre con el hombre– terminó con cuatro células que contienen 23 cromosomas cada una –donde no hay pares homólogos–. De esta forma el número de cromosomas se ha reducido de diploide a haploide.

Características estructurales externas de los cromosomas

Como decíamos anteriormente, la cromatina –material con el que se componen los cromosomas– es amorfa y se encuentra dispersa al azar por todo el núcleo, la mayor parte del tiempo. Sin embargo, cuando las células se preparan para su división, la cromatina comienza a condensarse y se hace visible al microscopio. En la metafase de la meiosis los cromosomas alcanzan su mayor grado de definición. Por ello, se aprovecha esta etapa para el estudio de la estructura de los mismos considerando sus formas, tamaños y constitución. A continuación se presenta una breve descripción estructural de los cromosomas en metafase.

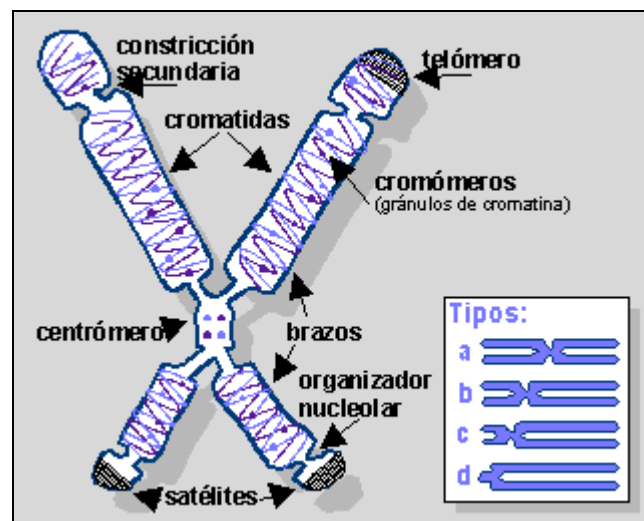


Ilustración 14 Detalle de un cromosoma metafásico y tipos de cromosomas.

El cromosoma está compuesto por dos elementos visibles que se llaman cromátidas hermanas o fraternales, unidas por un centrómero, que es una constrictión de posición primaria constante en cada cromosoma. Existen cuatro tipos de cromosomas según la posición del centrómero:

Metacéntrico: los brazos son aproximadamente iguales.

Submetacéntrico: los brazos originados por el centrómeros son ligeramente desiguales.

Acrocéntrico: los brazos son muy desiguales.

Telocéntrico: el centrómero es terminal o casi terminal.

Pueden existir además otras constrictiones llamadas secundarias o satélites.

Los cromosomas son de tamaños variados. En especies vegetales alcanzan los 30 micrones, mientras que en animales son de menor tamaño. El tamaño medio de los cromosomas humanos es de 5 micrones.

Los cromosomas puntuales o microcromosomas son unos cromosomas diminutos en los que no se aprecian las diferentes partes de la estructura. Un ejemplo es el cromosoma Y humano que interviene en la determinación del sexo.

En todas las células de un individuo, excepto en las germinales, y en las de todos los individuos de su misma especie, el número de cromosomas es constante. Recordemos que, como se expuso anteriormente, en las células germinales, tras la meiosis, el número de cromosomas es justo la mitad que en las demás células del individuo. Durante el proceso de la fertilización se fusionan el óvulo y el espermatozoide y se forma un cigoto con el número total de cromosomas.

Características estructurales internas de los cromosomas

El análisis químico de los cromosomas revela la existencia de ADN y ARN y de dos clases de proteínas en proporciones variables según el tejido y el organismo de procedencia –proteínas histónicas y no histónicas–.

Las cantidades de ADN y de histonas son constantes y semejantes, constituyendo el 50% del peso de la cromatina. El ARN se encuentra en menor proporción (un 10–15%), y las proteínas no histonas constituyen el resto.

Genes y cromosoma

Los genes son subdivisiones funcionales del ADN. En biología clásica se definían como una parte del cromosoma que determinaba o especificaba un carácter.

Mediante diversos procedimientos genéricos se han podido realizar "mapas" que muestran la disposición secuencial de muchos genes en los cromosomas. Se ha determinado que algunos genes están repetidos muchas veces en los cromosomas. Así, por ejemplo, los genes que codifican para las histonas tienen múltiples copias, ya que los embriones tempranos deben fabricar histonas muy rápidamente durante este período de rápido crecimiento.

Ahora bien, ¿Cómo se codifica la información genética en el ADN de los cromosomas? Para comenzar a responder esta pregunta transcribimos el siguiente párrafo de *Los Dragones del Edén* de Carl Sagan.

“...Dicha molécula de ADN está fragmentada en bloques o unidades autónomas unidos de forma similar a los peldaños de una escala de cuerda. Dichas unidades reciben el nombre de nucleótidos, de los que existen cuatro variedades. El lenguaje de la vida, el caudal de información hereditario, viene determinado por las diferentes disposiciones de los cuatro tipos de nucleótidos distintos. Así pues, podemos decir que el lenguaje de la herencia está escrito en un alfabeto de sólo cuatro letras”.

Carl Sagan, *Los Dragones del Edén*

En 1953, el genetista estadounidense James Dewey Watson y el británico Francis Harry Compton Crick lograron determinar la estructura y la forma en que se duplica la molécula de ADN. Descubrieron que esta macromolécula está formada por dos cadenas, o filamentos, alargadas que se enrollan formando una doble hélice, algo parecido a una larga escalera de caracol. Las cadenas, o lados de la escalera, están constituidas por nucleótidos. Cada nucleótido está compuesto por un grupo fosfato, un azúcar –desoxirribosa–, y una de las cuatro bases que contienen nitrógeno. Las cuatro bases nitrogenadas son adenina (A), timina (T), guanina (G) y citosina (C).

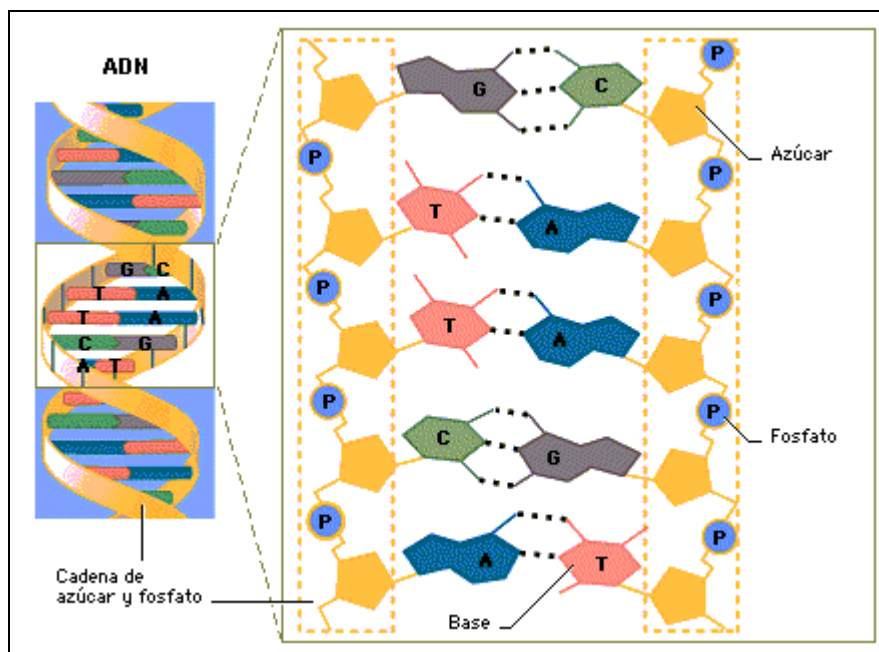


Ilustración 15 La molécula de ADN tiene la estructura de una escalera formada por azúcares, fosfatos y cuatro bases nucleóticas llamadas adenina (A), timina (T), citosina (C) y guanina (G). El código genético queda determinado por el orden de estas bases, y cada gen tiene una secuencia única de pares de bases.

Las bases nitrogenadas, dispuestas en parejas, representan los escalones. Cada base está unida a una molécula de azúcar y ligada por un enlace de hidrógeno a una base complementaria localizada en la cadena opuesta. La adenina siempre se vincula con la timina, y la guanina con la citosina. Para hacer una copia nueva e idéntica de la molécula de ADN, sólo se necesita que las dos cadenas se extiendan desenrollándose y se separen por sus bases –que están unidas de forma débil–. Una vez separadas y gracias a la presencia en la célula de más nucleótidos libres, se pueden unir a cada cadena separada bases complementarias nuevas, formando dos dobles hélices.

La estructura del ADN es en realidad mucho más larga que la del cromosoma, pero se halla muy condensada –Ilustración 16–. Ahora se sabe que este empaquetamiento se basa en diminutas partículas llamadas nucleosomas, sólo visibles con el microscopio electrónico más potente. El ADN está enrollado secuencialmente alrededor de cada nucleosoma formando una estructura en forma de rosario. Entonces la estructura se repliega aún más, de manera que las cuentas se asocian en espirales regulares.

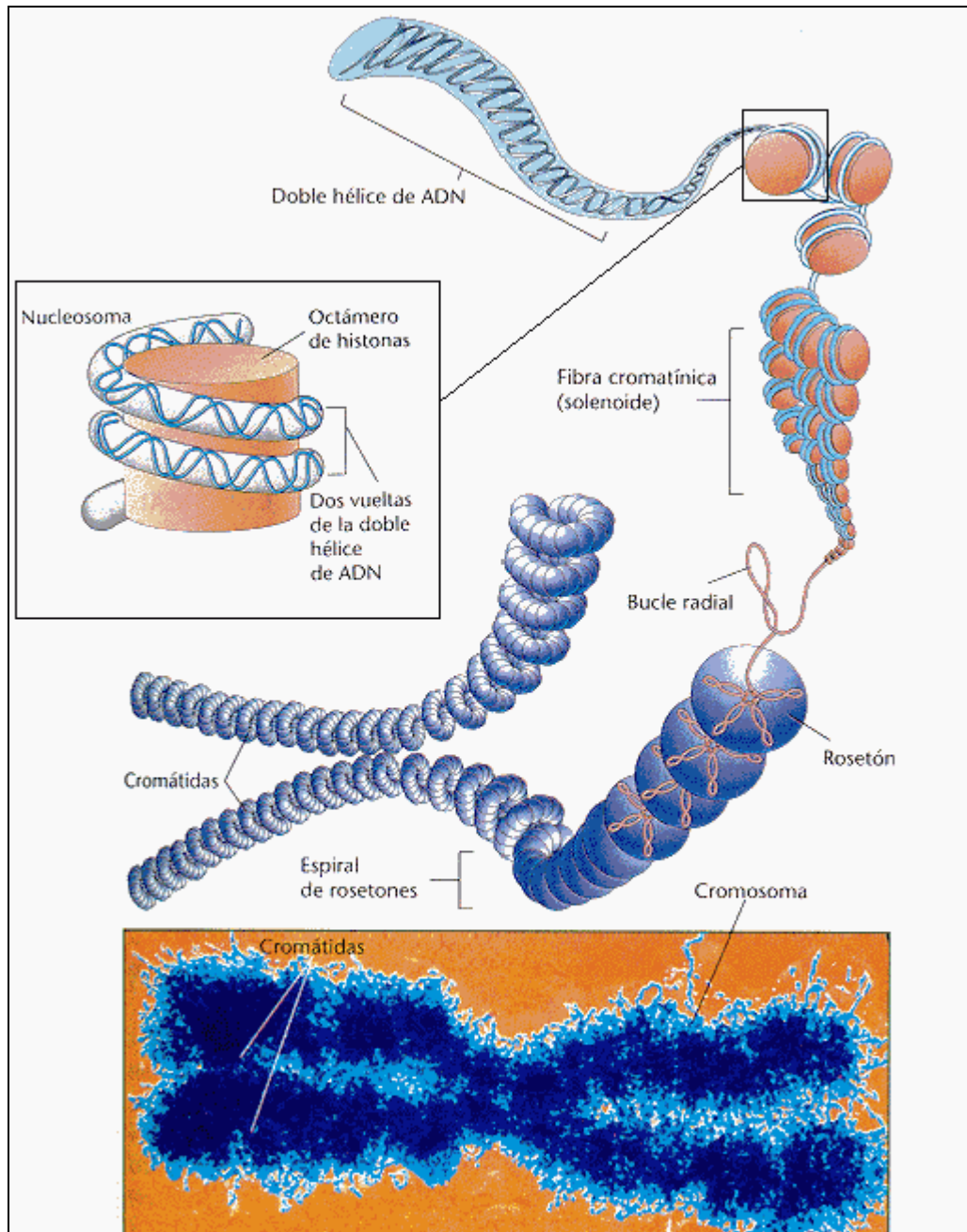


Ilustración 16 Gráfico editado a partir de varias imágenes extraídas de [AVB]

Tras los descubrimientos de Watson y Crick, quedó el interrogante de saber cómo el ADN dirigía la expresión de los genes. Diez años después el código genético fue descifrado y verificado. Los genes se expresan principalmente a través de la fabricación de proteínas – piezas y herramientas con que están contruidos los seres vivos– que tiene lugar en los ribosomas de las células. La molécula de ADN lleva escrito con palabras de tres letras – llamadas tripletas o *codones*– del alfabeto {A,T,C,G} cada uno de los aminoácidos y el orden en que deben enlazarse para formar las distintas proteínas.

De un modo general, es posible decir que las proteínas o bien cumplen funciones estructurales, para integrar membranas y otros micro órganos de la célula o bien poseen

funciones enzimática, para catalizar¹ las reacciones químicas en el funcionamiento del organismo.

Los aminoácidos son una clase importante de compuestos orgánicos de los cuales 20 sirven de ladrillos fundamentales para las proteínas. Esta claro que se necesitan palabras de código de al menos 3 símbolos pues 2 no son suficientes para identificar uno entre 20 aminoácidos ($4^2 = 16$). Pero sobran tripletas, pues existen $4^3 = 64$ codones distintos lo que implica que muchas palabras comparten el mismo significado. El hecho es que 61 tripletas codifican redundantemente los 20 aminoácidos, mientras que las otras tres constituyen delimitadores de los genes, señalan el final del código de una determinada proteína. Aunque algunos aminoácidos son especificados por más de un codón, ninguna tripleta especifica más de un aminoácido.

Las proteínas están compuestas por una o varias cadenas de entre 50 y varios cientos de subunidades de aminoácidos. Se fabrican en los ribosomas que se encuentran en el citoplasma de las células siguiendo estrictamente las instrucciones escritas en el ADN.

Una extraordinaria variedad de proteínas pueden ser producidas a partir de las mismas 20 subunidades. Las proteínas tienen la propiedad de doblarse sobre sí mismas de una manera peculiar, con lo que adquieren la forma tridimensional que las capacita para interactuar reactivamente con otras moléculas grandes como el ADN, y otras proteínas, para desempeñar su papel de artífices y materiales de la vida. Volveremos a comentar la relevancia de la forma tridimensional de las proteínas en el capítulo “Redes Neuronales. Fundamentos biológicos” en el apartado “Información molecular y reacciones químicas en la célula” –pág. 93– pues es de vital importancia en la comunicación sináptica entre neuronas por medio de los neurotransmisores.

Un descubrimiento reciente e inesperado es que, en los organismos superiores, los genes están interrumpidos. A lo largo de una secuencia de nucleótidos que codifican una proteína en particular, puede haber una o más interrupciones formadas por secuencias sin codificar. En algunos genes pueden encontrarse 50 o más de estas secuencias, o intrones. Las secuencias que corresponden a los intrones son ignoradas en la síntesis de proteínas. Se estima que en los seres humanos, de todo el ADN de las células sólo un 5% codifica para alguna proteína. [ACIENT]

Pero mientras que el ADN se encuentra encerrado en el núcleo de cada célula eucariota, los ribosomas y moléculas de aminoácidos flotan libres en el citoplasma. Un interesante mecanismo de comunicación ha evolucionado que permite llevar las instrucciones genéticas hasta las fábricas de proteínas. El mensajero lo constituye otro ácido nucleico, el ARN –ácido ribonucleico– en particular el ARNm –ARN mensajero–.

Las cadenas que forman la doble hélice del ADN se separan definitivamente para la replicación del mismo, o temporalmente para la expresión de los genes. En el segundo caso, una de las bandas sirve como patrón para crear una cadena de ARNm que es muy parecido al ADN y también tiene cuatro bases nitrogenadas salvo que en lugar de Timina posee Uracilo, un cambio del alfabeto {A,T,C,G} por este otro {A,U,C,G} que no compromete la información que transcribe. Una vez terminada la transcripción –en la que adecuadamente, no

¹ Se dice que una sustancia, a la que se denomina enzima, cataliza una reacción química si interviene en ella, haciéndola posible o acelerándola. Al terminar la reacción la enzima queda intacta, pues en el proceso se reconstituye y separa de los otros elementos de la reacción con los cuales se integra solo transitoriamente

se copian los intrones– el ARNm atraviesa la membrana nuclear y se dirige hacia los ribosomas donde se fabrica la proteína cuya codificación transporta. El ribosoma va ensamblando los aminoácidos en el orden en que los presenta el ARN mensajero y, conforme avanza a lo largo del mensaje, va produciendo la proteína. La señal de haber concluido el trabajo la da el mensaje mismo, gracias al signo de alto con que termina cada gen. Cuando la proteína está concluida, se desprende del ribosoma y se dobla sobre sí misma para tomar la compleja forma tridimensional que la caracteriza.

Tanto la duplicación del ADN como la transcripción al ARNm se lleva a cabo con la colaboración de una poderosa enzima –proteína de acción enzimática o catalizadora– llamada polimerasa, la cual realiza el complejo trabajo de atraer del ambiente celular los compuestos necesarios –azúcares, ácido fosfórico, bases nitrogenadas– e irlos armando progresivamente según el modelo ofrecido por la banda patrón y gracias a la complementariedad característica de las bases nitrogenadas.

Finalmente debe considerarse que no todas las proteínas son producidas por un único gen y que tampoco todos los genes dan origen a proteínas, algunos se transcriben en ARN que se pone directamente a trabajar en funciones especiales, como ser parte de un ribosoma, servir de agente transferente de aminoácidos durante la creación de una proteína, o alguna otra función necesaria en el citoplasma.[Gutierrez00_2] El ARN, al igual que las proteínas, también se encarga de catalizar algunas reacciones químicas, este hecho hace factible la hipótesis que antes de un mundo viviente basado en ADN existió uno más primitivo basado en ARN. Como ya se expuso tempranamente en este capítulo, la alta complejidad tecnológica de la molécula del ADN, junto con la dependencia de estructuras muy especializadas que no pudieron existir antes del inicio de la propia evolución hacen impensable su postulación como primera molécula autorreplicativa.

Recombinación genética

Si bien ya se han expuesto los conceptos de entrecruzamiento –crossover– y mutación, he aplazado deliberadamente el tratamiento de ciertos detalles hasta este punto del capítulo. La razón es simple: el crossover junto a la mutación son los operadores más importantes en la teoría de los Algoritmos Genéticos, y preparando la transición al próximo capítulo pretendo cerrar el presente con alguna discusión al respecto.

Para multiplicar aún más la enorme diversidad genética, existe el fenómeno de la recombinación, cuya existencia fue descubierta por Morgan en 1911 en unos experimentos con la mosca del vinagre –o de la fruta–, *Drosophila melanogaster*.

Ya se mencionó que durante la meiosis, cuando los cromosomas homólogos se aparean, se forman con frecuencia puentes entre dos regiones correspondientes del cromosoma. Estos puentes llamados chiasmas –o quiasmas– constituyen el punto de cruce, a partir del cual se produce el intercambio de fragmentos entre cromosomas homólogos.

Este proceso, llamado entrecruzamiento o *crossover*, ocurre con gran frecuencia en cada meiosis. El intercambio de fragmentos es la recombinación genética. También pueden ocurrir entrecruzamientos de dos puntos –dos chiasmas–. No obstante se ha determinado experimentalmente que la frecuencia de dobles entrecruzamientos es baja, porque resulta muy difícil separar un gen que se encuentra en el medio de otros dos. De esta forma, se ha deducido que los dobles entrecruzamientos se producen en menor proporción que la esperada

Cuanto más alejados estén dos genes en el mismo cromosoma, mayor probabilidad existe de que ocurra un entrecruzamiento simple en el espacio entre ambos. Este concepto está relacionado con la idea de ligamiento que se describe a continuación.

Ligamiento

Si se piensa en la gran cantidad de genes que determinan las características hereditarias de los seres vivos y en el número de cromosomas que hay en cada una de sus células, se evidencia inmediatamente que hay muchísimos más genes que cromosomas.

En la especie humana se estima que existen unas decenas de miles¹ de genes que se encuentran ubicados en cuarenta y seis cromosomas. Por lo tanto cada cromosoma debe portar más de un gen, formando así los llamados grupos de ligamiento que se transmiten juntos durante la formación de los gametos.

Los *locus* cercanos en un mismo cromosoma se dice que están "ligados" o que tienen un vínculo génico entre ellos, llamado ligamiento, porque si se produce recombinación, es muy fácil que permanezcan juntos en un mismo fragmento. Dicho de otra manera, la probabilidad de separar dos genes ligados por crossover es baja.

Estos grupos de genes no cumplen la tercera ley de Mendel (ver "Tercer experimento y tercera ley de Mendel", –pág.22–) ya que no hay transmisión independiente de factores mendelianos o genes. Las primeras evidencias de grupos de ligamiento fueron aportadas en 1906 por Bateson y Punnet, quienes hicieron varios cruzamientos entre plantas de flores color púrpura y grano de polen alargado, con plantas de flores rojas y de grano de polen redondo, sin obtener los resultados pronosticados por la tercera ley de Mendel en la proporción esperada. Estas pruebas permitieron inferir que los genes que controlan las dos características estudiadas se encontraban en el mismo cromosoma, y no había transmisión independiente de tales genes.

Pero el ligamiento no es completo, como vimos antes, los genes tienen la posibilidad de separarse a través del entrecruzamiento, por más cerca que estén unos del otro, existe una probabilidad de ser separados. Esto fue demostrado por primera vez por T. Morgan en sus experimentos con la mosca del vinagre.

La intensidad de la recombinación dependerá de la distancia a que se encuentran los genes dentro del mismo cromosoma. Si los genes están muy juntos se transmitirán casi siempre ligados; si se encuentran muy separados, se transmitirán con mucha frecuencia independientemente.

Mutaciones

Una mutación es un cambio hereditario en el material genético de un organismo. Junto con el crossover constituyen dos importantes conceptos tomados en la elaboración de las teorías de los Algoritmos Genéticos.

El material hereditario puede sufrir cambios en su estructura, secuencia o cantidad. Aunque las causas de las mutaciones en el hombre en gran parte son desconocidas, entre ellas se encuentran diversos factores ambientales como radiaciones, virus y productos químicos. No

¹ A la fecha de realización del presente trabajo, no hay certeza acerca de la cantidad de genes de la especie humana. A pesar de que en febrero de 2001, investigadores del Proyecto Genoma Humano y Celera Genomics presentaron conjuntamente los resultados de su trabajo que establecía en 30.000 los genes del ser humano, otros estudios posteriores han puesto en duda esta cifra sugiriendo que podrían ser muchos más. William Haseltine, fundador de Human Genome Sciences (HGS), afirma que tiene catalogados 90.000. Un estudio efectuado por un grupo de científicos de la Universidad de Ohio (Estados Unidos) también ha puesto en duda la cifra que ofrecieron en febrero los equipos del Proyecto Genoma Humano y de Celera Genomics. Los investigadores de Ohio analizaron la información genética con un supercomputador llegando a contabilizar 66.000 genes humanos. [ALDEAEDU]

se debe olvidar, también que las mutaciones pueden ocurrir por causas naturales. El procedimiento de réplica del ADN es complejo y pueden ocurrir errores durante la copia, de hecho ocurren con bastante frecuencia pero son corregidos –casi siempre– por una enzima especializada en ello¹.

Las mutaciones pueden producir una alteración visible de la estructura de un cromosoma, como una pérdida parcial o la translocación de una porción –mutaciones cromosómicas–, o causar un cambio muy leve en uno de los pares de bases de un gen –mutaciones puntuales–. En este último caso, el cambio más pequeño sería el correspondiente a un par de bases pero, a veces, se utiliza la expresión "mutación puntual" para el cambio que afecta a un cierto número de pares de bases adyacentes, o incluso la pérdida o intercalación de bases.

Morgan demostró que existen mutaciones de muy diversos grados, ya que los cambios provocados por ellas pueden incluso ser tan pequeños que no resulta fácil descubrirlos. El estudio de las mutaciones es de gran importancia porque constituyen una de las bases fundamentales del proceso evolutivo.

La parte negativa de las mutaciones es que muchas veces son responsables de determinadas enfermedades que no sólo van a afectar al organismo que sufre la mutación, sino que pueden afectar también a su descendencia. De manera fortuita, ciertas mutaciones beneficiosas suministran el material de base que impulsa la evolución biológica, pero ello no ocurre muy a menudo, pues la probabilidad de que un mecanismo de precisión mejore como resultado de un cambio al azar en las instrucciones para construirlo es baja. No obstante, 4.000 millones de años es mucho tiempo como para que la aparición de errores en el material hereditario haya permitido alcanzar la variabilidad de formas de vida que existen hoy.

Si el número de mutaciones es demasiado elevado, se perdería la herencia de 4.000 millones de años de lenta evolución. Si es demasiado bajo, no se dispondría de nuevas variedades para adaptarse a algún cambio futuro en el medio ambiente. La evolución de la vida exige un equilibrio más o menos preciso entre mutación y selección.

Los organismos de mayor tamaño como el hombre, sufren por término medio una mutación por cada diez gametos [Sagan_2] son mutaciones al azar y por consiguiente casi siempre nocivas. No obstante, por lo general las mutaciones son recesivas, sus efectos perjudiciales no se expresan a menos que dos de ellos coincidan para dar lugar a una situación homocigótica. Esto es más probable en la procreación consanguínea, en el apareamiento de organismos muy relacionados que pueden haber heredado el mismo gen mutante recesivo de un antecesor común. Por esta razón, en los seres humanos, las enfermedades hereditarias son más frecuentes entre los niños cuyos padres son primos que en el resto de la población. Otro dato a tener en cuenta es que la mutación podría manifestarse en alguna porción del ADN que no codifica para ningún gen –intrones–, en tal caso no tendrá efecto sobre el fenotipo.

¹ En la reproducción del ADN se distinguen tres fases. a) *Se desenrolla la hélice*: se rompen los puentes de hidrógeno y se separan las dos hebras. b) *Síntesis del ADN*: Los nucleótidos libres se acoplan a las bases de cada hebra formándose así dos moléculas de ADN cada una con una hebra de la molécula original y una nueva sintetizada a partir de aquella. c) *Corrección de errores*: existen enzimas capaces de detectar errores en la síntesis, de cortar segmentos incorrectos, de rellenar huecos dejados por nucleótidos y de unir extremos corregidos, a esta actividad, los biólogos moleculares la han denominado “corrección de pruebas” Un error en la corrección de pruebas, error raro pero factible, provoca una mutación.

Clásicamente se han venido distinguiendo varios tipos de mutaciones de acuerdo con la forma de sufrir cambios el material genético. Así, según Timoféeff-Resovsky, se pueden establecer los siguientes tipos:

Mutaciones espontáneas

Son mutaciones que tienen lugar por errores en el manejo interno del material genético en las células, es decir, que ocurren de un modo natural.

Tienen lugar en la naturaleza con frecuencia relativamente baja, ya que, a pesar de las continuas mutaciones espontáneas que sufren los seres vivos, la identidad de los organismos se transfiere de generación en generación con una fidelidad extraordinaria. Esto se consigue gracias a la notable capacidad que ya mencionamos, de aquellas enzimas encargadas de proteger y reparar la secuencia de bases específica de los cromosomas de cada organismo.

El papel de las enzimas reparadoras es muy importante, ya que la mayor parte de las mutaciones que se producen son automáticamente reparadas por ellas.

Mutaciones inducidas

Son aquellas que se producen por influencia de factores externos. Los principales son radiación ultravioleta, radiaciones ionizantes y mutágenos químicos.

Mutaciones cromosómicas

Caracterizadas por cambios en la estructura de los cromosomas. Tales cambios pueden consistir en la pérdida de un trozo –delección–, o en su adición, con lo cual existe un segmento repetido –duplicación– o bien en el traslado de una porción de un cromosoma a otro no homólogo –translocación–, e incluso que un trozo de cromosoma invierta su posición –inversión–.

Mutaciones cariotípicas o genómicas

También denominadas mutaciones numéricas, afectan al cariotipo o dotación cromosómica en su conjunto y se deben a la aparición de un número de series haploides distinto del normal – triploide, tetraploide, heteroploide, etc.– o aumento o disminución de algún cromosoma. Ejemplo clásico de una mutación correspondiente a esta categoría es la trisomía del cromosoma 21 humano, que da lugar al síndrome de Down o mongolismo.

Mutaciones génicas

En las cuales el cambio afecta a la constitución química de los genes. Lógicamente, por tratarse de cambios moleculares, este tipo de mutaciones a diferencia de las anteriores no son observables ni siquiera con el microscopio electrónico.

Mutaciones estructurales

Cuando el cambio estructural se verifica en un solo cromosoma, se llama *intracromosómico*. Si afecta a dos o más cromosomas, es *intercromosómico*. Existen varios tipos de mutaciones:

Delección. Es una mutación estructural en la que se produce pérdida de material génico. Su efecto depende de si el individuo es heterocigoto u homocigoto respecto a la variación estructural. En general, cuanto mayor es la deficiencia o delección, mayor es el desequilibrio génico producido, pudiendo llegar a ser inviable el gameto o cigoto portador del cambio. Si sobreviven, la delección se transmite a la descendencia como un factor mendeliano.

Duplicación. Es la presencia repetida de un segmento cromosómico. Puede ser directa o inversa según el segmento cromosómico se repita en el mismo orden o invertido. Los segmentos repetidos pueden estar más o menos distantes dentro del cromosoma, o bien seguidos. La duplicación puede producirse por una serie de roturas y reuniones. El efecto de

una duplicación depende de los genes a los que afecte. Los genes duplicados producirán generalmente las mismas consecuencias en el fenotipo que si estuvieran en su posición normal, y la transmisión de las duplicaciones es mendeliana.

Inversión. Se produce por un mínimo de dos roturas y dos reuniones. En las inversiones no hay ganancia ni pérdida de material genético y por lo tanto no hay alteración del equilibrio, pero existen inversiones con efecto fenotípico que en un principio se atribuyeron a pequeñas deleciones en los puntos de rotura. Luego, al conseguirse reproducir la inversión experimentalmente y su reversión a la estructura primitiva, se ha visto que se recupera el efecto fenotípico primitivo, por lo tanto no se había perdido material genético. Por eso se atribuyen los cambios fenotípicos a un efecto de posición.

Translocación. Son transposiciones de un segmento cromosómico que se coloca en otra posición dentro del mismo cromosoma. Se necesita un mínimo de tres roturas. La translocación puede producirse también entre dos cromosomas distintos. Las translocaciones, en el caso de que no sean letales, se heredan de forma mendeliana.

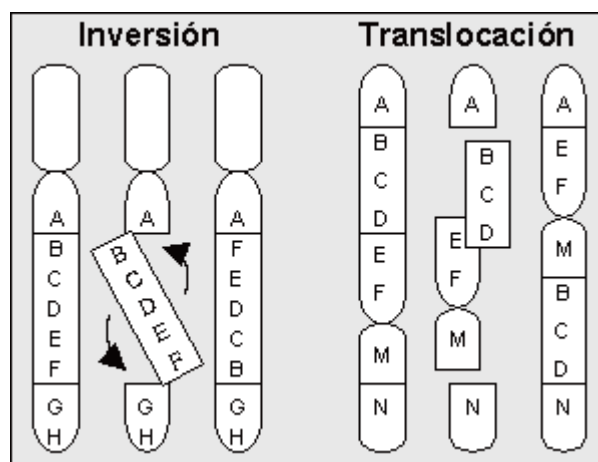


Ilustración 17 Mecanismo de una inversión y de una translocación.

Mutaciones puntuales

Son cambios que afectan a un gen o, como mucho, a varios genes vecinos al mismo tiempo. También se llaman mutaciones génicas. El gen primitivo se llama *alelo salvaje* y el mutado *alelo*.

En el hombre existen numerosas enfermedades cuya base genética es una mutación de este tipo. Ejemplos de ello los tenemos en la hemofilia, la talasemia y el albinismo.

Las mutaciones que se originan cuando una base es reemplazada por otra incorrecta se llaman mutaciones *por sustitución*. Cuando no producen cambios en las propiedades biológicas de la proteína para la que codifica el gen, la mutación se llama *silenciosa*.

Las mutaciones *por inserción* y *por supresión* son mucho más numerosas y letales.

Otros resultados de la teoría cromosómica de la herencia

Herencia ligada al sexo

Durante sus experimentos con la mosca del vinagre, Morgan descubrió que ciertos rasgos se manifestaban sólo en un determinado sexo, el otro no lo manifestaba, pero sí lo transmitía a su descendencia.

Observando los cromosomas de la mosca del vinagre al microscopio descubrió que en los machos, de los cuatro pares de cromosomas, uno de los cromosomas de una pareja era más

corto que el otro, como si le faltara un trozo, y llamó a esta pareja XY. Sin embargo en las hembras, a esta pareja de cromosomas no le faltaba ningún trozo, eran los dos cromosomas iguales, y llamó a esta pareja en las hembras XX.

Morgan pensó que los resultados extraños se debían a que el gen estudiado debía estar situado en la porción que faltaba del cromosoma Y, de forma que en las hembras, habría dos copias de este rasgo, heredándose de forma normal. Pero en los machos existiría una sola copia, por lo que el rasgo se manifestaría o no dependiendo de su presencia o ausencia.

En los seres humanos, por ejemplo, el daltonismo es una enfermedad ligada a los cromosomas sexuales. Un hombre con daltonismo, incapaz de distinguir el rojo del verde, porta un gen alterado y recesivo en su cromosoma X, y este lo transmitirá a todas sus hijas. Debido a que el gen es recesivo, estas hijas con una sola copia del gen alterado no presentarán la enfermedad, ya que su otro cromosoma X será normal, pero son portadoras. Ellas lo transmitirán a la mitad de sus hijas que serán también portadoras y a la mitad de sus hijos que serán daltónicos. Algo similar ocurre con la hemofilia. Existen también rasgos ligados al cromosoma Y, como por ejemplo los responsables de la aparición de vello en las orejas, o el factor de diferenciación testicular, que sólo aparecen en los hombres.

Interacciones entre genes no alélicos

Rara vez la acción de los genes es cuestión de un gen aislado que controla un solo carácter. Con frecuencia un gen puede controlar más de un carácter, y un carácter puede depender de muchos genes. Por ejemplo, es necesaria la presencia de al menos dos genes dominantes para producir el pigmento violeta en las flores de la planta del guisante de olor.

Se han descubierto varias formas en que genes que codifican para distintos rasgos son influidos unos a otros.

Epistasis: se llama así al enmascaramiento entre genes. Es decir, uno de ellos enmascara la expresión del otro.

Pleiotropía: el cambio de un solo gen afecta la expresión de muchos otros.

Genes transponibles: Son genes que se mueven de una posición a otra dentro del genoma, modificando la expresión de los demás genes, y modificando por tanto el fenotipo.

Herencia cuantitativa o poligénica

Los caracteres que se expresan como variaciones en cantidad o extensión, como el peso, la talla o el grado de pigmentación, suelen depender de muchos genes, así como de las influencias del medio. Con frecuencia, los efectos de genes distintos parecen ser aditivos, es decir, parece que cada gen produce un pequeño incremento o descenso independiente de los otros genes.

En este tipo de herencia, el ambiente suele jugar un papel fundamental, al contrario que en la herencia monogenética o mendeliana, donde el rasgo se expresa o no, dependiendo de la presencia o ausencia del gen. La combinación de influencias genéticas y del medio se conoce como herencia multifactorial, aquí la expresión del fenotipo depende de los estímulos ambientales. Por ejemplo, un niño subalimentado no crecerá tanto como le permite su herencia, ni su desarrollo alcanzará todo su potencial. La herencia genética proporciona el potencial, pero del medio ambiente dependerá el grado en que este se desarrolle.

Hasta aquí la mirada de la evolución y genética desde el punto de vista biológico. En el capítulo siguiente cambiaré el enfoque para abordar el paradigma informático conocido con el

nombre de “Computación Evolutiva”, del cual, los Algoritmos Genéticos constituyen la rama más representativa, por ser la más utilizada, completa, y con mayor base teórica.

Computación Evolutiva

Introducción

En el presente capítulo introduciré los conceptos básicos de este paradigma, focalizando especialmente en los Algoritmos Genéticos como rama más representativa de la Computación Evolutiva.

Una pequeña referencia a los métodos de búsqueda y optimización, antecede a un breve relato sobre los inicios de esta disciplina, cediendo el paso luego a la temática que ocupará más del 70 por ciento del capítulo: los Algoritmos Genéticos.

Siguiendo a Goldberg [*Goldberg89*] presentaré un ejemplo de Algoritmo Genético Simple para luego entrar en detalles sobre la anatomía de esta familia de algoritmos. Se presentarán variantes y alternativas al esquema general, sin dejar de lado la fundamentación teórica – *teorema de esquemas e hipótesis de bloques constructivos*– que los distingue de otros paradigmas de la Computación Evolutiva. Tampoco se omitirán las críticas que atacan estos basamentos teóricos desde algún sector de la Inteligencia Artificial.

Aspectos prácticos en relación a la codificación, frecuencia de operadores y variantes sobre todo para mantener la diversidad genética, serán presentados antes de realizar una breve evaluación del desenvolvimiento de los Algoritmos Genéticos en distintas clases de problemas.

Finalmente se cerrará el capítulo con una brevísima descripción de los paradigmas que incluye la Computación Evolutiva.

Antes de pasar a los orígenes de la Computación Evolutiva, y como parte de esta introducción, dedicaré unas líneas a caracterizar los tipos de problema en que una solución empleando algoritmos evolucionarios puede resultar adecuada.

Métodos de búsqueda y optimización

Una manera de resolver un problema es mediante un algoritmo, es decir un método que proporciona la solución al mismo en tiempo finito. El término proviene por varias traducciones y corrupciones del nombre del matemático persa Mûsâ al-Khowârizm, que escribiera un libro sobre procedimientos aritméticos por el año 835 y que fuera traducido al latín en el siglo XII. El concepto de algoritmo como un procedimiento seguro y en cierta forma mecánico de lograr algún resultado ha tenido vigencia por varios siglos hasta que Turing en 1936 lo definió formalmente en términos de la operación de una máquina abstracta que ahora lleva su nombre.[*Gutierrez97_2*] Se puede decir que un algoritmo es un conjunto finito de reglas que proporciona una prescripción exacta sobre la realización, en una secuencia determinada, de acciones conducentes en un número finito de pasos a la solución de una clase específica de problemas [*KVITCA*]. Ni más ni menos que un procedimiento factible de llevarse a cabo por una máquina de Turing que siempre se detiene alcanzando la solución al problema.

No obstante a veces existen razones por las cuales no es aconsejable o simplemente es imposible utilizarlo:

1. El problema en cuestión puede ser indecidible. En tal caso, no existe algoritmo alguno capaz de resolverlo en tiempo finito, es decir, no se puede construir una máquina de

Turing que pueda computarlo, garantizando que ésta no entre en un “loop” indefinidamente.

2. El tiempo para hallar la solución, a pesar de ser finito puede ser extremadamente grande.
3. Aunque exista y sea eficiente, puede ser muy complicado hallar el algoritmo adecuado para la resolución de ciertos problemas.

La Inteligencia Artificial encara, entre sus diversas áreas de aplicación, la forma de resolver problemas cuando no se tiene un algoritmo sistemático o directo para hacerlo. Para ello utiliza los denominados métodos débiles, que en su mayoría incluyen búsquedas en un espacio de soluciones.

Un ejemplo clásico de problemas de búsqueda es el de las 8 reinas, en el cual se deben colocar estas 8 piezas en los casilleros de un tablero de ajedrez de forma que ninguna amenace a otra. Este planteo puede resolverse fácilmente utilizando algoritmos recursivos que recorren, de ser necesario, todo el espacio de búsqueda hasta hallar la solución. A esta clase de métodos de búsqueda se los llama exhaustivos. Sin embargo no siempre se pueden utilizar, aún con espacio de búsqueda finito, determinados problemas requieren de algoritmos de complejidad temporal exponencial¹. Por ejemplo se podría computar un árbol de búsqueda completo para determinar la mejor próxima jugada en una partida de ajedrez, pues se trata de un espacio finito, pero seguramente el programador no viviría lo suficiente como para ver el primer movimiento de su algoritmo.

En estos casos, un método de búsqueda heurístico es más aconsejable, pues estimando la distancia a la meta, utiliza reglas para eliminar zonas del espacio de búsqueda consideradas “poco interesantes”. Estos métodos no son infalibles pues la mejor solución podría hallarse en un espacio no explorado, pero al igual que ocurre con la forma en que el ser humano se enfrenta a los problemas, estas técnicas permiten alcanzar de modo factible, soluciones que, aunque no sean las mejores teóricamente son aceptables.

En muchos casos, la búsqueda está guiada por una función que indica lo buena que es esa solución, o el costo de la misma, o lo cerca que se está de la solución final –heurística–, si es que se conoce; el problema se convierte entonces en un problema de optimización, es decir, encontrar la solución que maximiza la función objetivo, de evaluación o fitness² o minimiza el costo. En términos formales, dada una función F de n variables x_1, x_2, \dots, x_n , optimizar la función consiste en encontrar la combinación de valores de x_i tales que $F(x_1, x_2, \dots, x_n) = \text{Máximo}$. Se puede hablar de maximizar en vez de minimizar sin perder generalidad, ya que maximizar F equivale a minimizar $-F$. [Merelo97]

Como último comentario de este apartado me gustaría hacer una reflexión basada en el artículo “El algoritmo informático” de Claudio Gutiérrez [Gutierrez97_2] donde se presenta una interesante relación entre los algoritmos, los programas de Inteligencia Artificial –clásica o simbólica– y la teoría de evolución por selección natural que ya vimos en el capítulo anterior.

¹ Se dice que un problema es de orden temporal exponencial (no polinomial y pertenece al conjunto de problemas NP) si para resolverlo es necesario un algoritmo que requiera $O(c^n)$ instantes de tiempo para procesar un input de longitud n , con c constante. De esta manera, si la longitud del input aumenta, el tiempo necesario para computarlo aumenta en forma exponencial.

² La definición biológica de fitness no es otra cosa que el número posible de descendientes de un individuo (que no siempre es el mismo que el número *real* de descendientes). Generalmente así se calcula en los problemas de Biología que la Vida Artificial –una rama de la Inteligencia Artificial Subsimbólica– trata de resolver mediante simulaciones.

Como se indicó más arriba, los programas de Inteligencia Artificial basados en búsquedas heurísticas no tienen garantizado su éxito debido a que el espacio explorado generalmente es mucho más chico que el espacio total de búsqueda a veces infinito. Además, cualquiera que haya interactuado con alguno de estos programas, como por ejemplo uno que juegue ajedrez, puede dar fe de que su comportamiento aparente no es el de un procedimiento mecánico, sino todo lo contrario, llegando incluso a sorprendernos con soluciones que parecen creativas o novedosas. Estas dos afirmaciones contradicen el concepto de algoritmo presentado párrafos antes. Pero la contradicción es sólo aparente; pues estos programas están basados en algoritmos que realizan infaliblemente lo que están llamados a realizar: generar de manera más o menos arbitraria una serie de hipótesis plausibles y enseguida escoger mediante algún criterio –a veces con algún condimento aleatorio– aquella que resulte más prometedor.

Ahora bien, esta táctica denominada *generar y escoger* corresponde totalmente al proceso de selección natural que vimos en el capítulo anterior –pág. 8–. La etapa de *generar* está dada por la reproducción con eventuales mutaciones y recombinación genética por crossover en los organismos superiores, lo que produce suficiente variabilidad, para que la selección del medio ambiente escoja los individuos mejor adaptados a la situación reinante –etapa de *escoger*–. Según Claudio Gutiérrez la selección natural, con millones de años de historia constituye, sin lugar a dudas el más antiguo de todos los algoritmos.

Orígenes de la Computación Evolutiva

Ya en el año 1932 Canon concibió a la evolución natural como un proceso de aprendizaje, y Alan Turing en 1950 pensó que debía existir una conexión entre el aprendizaje máquina y la teoría de evolución darwiniana. También señaló que se podrían desarrollar programas para jugar a la ajedrez utilizando como modelo esta teoría.

No obstante, fue Von Neuman, científico al que se debe la arquitectura de las computadoras actuales, quien aún antes de que Watson y Crick descubrieran la estructura del ADN¹ concibió la idea que la vida debe estar sostenida por algún código capaz de crear seres vivos con capacidad de autorreproducirse, es decir, dicho código no sólo debe contener las instrucciones necesarias para crear un organismo, sino también para copiar dichas instrucciones a su descendencia. En la década del 40 desarrolló el modelo computacional de los Autómatas Celulares². El razonamiento de Von Neumann, se resume así según Langton³:

- La autorreproducción, se lleva a cabo por una máquina bioquímica altamente compleja.

¹ El bioquímico estadounidense James Watson y el biofísico británico Francis Crick publicaron la primera descripción de la estructura del ADN en el año 1953. Para más información acerca de la estructura del ADN consultar el apartado “Genes y cromosoma” –pág. 32–

² Los autómatas celulares (AC) son importantes en el contexto de la vida artificial (paradigma de la Inteligencia Artificial subsimbólica). Un AC es un modelo formal que está compuesto por un conjunto de entes elementales, cada uno de ellos susceptible de encontrarse en un cierto estado y de alterarlo de un instante al siguiente, asumiendo que el tiempo transcurre de forma discreta. La regla que gobierna la transición de estados en los entes es sensible a los estados de los demás elementos en su vecindad, siendo por tanto una regla de transición local. El aspecto que más caracteriza a los ACs es su capacidad para dotar al conjunto del sistema, visto como un todo, una serie de propiedades emergentes inducidas por la propia dinámica local. Los AC se popularizaron a través de “El juego de la vida” de John Conway que, a su vez, alcanzó popularidad gracias a Martin Gardner y se convirtió en un objeto de culto para una generación de jóvenes científicos en los años 70.

³ Christoffer Langton es el investigador que, siguiendo la línea de von Neumann, más ha estudiado los autómatas celulares como modelos de cómputo que pueden dar cuenta de la autorreproducción, característica básica de lo viviente, tratando de crear expresamente una base para la Vida Artificial dentro de la Informática Teórica. [Cases]

- Entonces, el comportamiento de esa máquina es descriptible como una secuencia lógica de etapas.
- Luego, si el algoritmo se puede llevar a cabo por alguna máquina, existe una máquina de Turing que hace lo mismo –tesis de Church–.
- Por lo tanto, se requiere modelar una máquina de Turing capaz de autorreproducirse. Si esta máquina de Turing autorreproductiva existe, es plausible que los procesos de autorreproducción biológicos sean algorítmicamente descriptibles, y por lo tanto la vida puede simularse en máquinas.

Von Neumann consiguió bosquejar una máquina de Turing autorreproductiva utilizando los Autómatas Celulares. A pesar de que era un físico y matemático excepcional, en su trabajo original hay una falta de consideraciones físicas; su interés estaba dirigido principalmente a dar una explicación reduccionista de ciertos aspectos de la biología. De hecho los mecanismos que propuso para alcanzar estructuras autorreplicantes se asemejan bastante a los que utilizan los organismos vivos.

En 1957 Gregor Box desarrolla la técnica denominada EVOP –Evolutionary Operation– para mejorar procesos industriales. La idea básica fue substituir la operación estática de un proceso por un esquema continuo y sistemático de perturbaciones leves en las variables de control. El efecto de estas pequeñas perturbaciones se evaluaba y el proceso se cambiaba en la dirección de la mejora. Es decir, estaba aplicando dos conceptos de la evolución natural: la mutación y la selección de las variantes más adecuadas –selección natural–, con el objeto de mejorar la calidad del proceso. Este procedimiento se aplicó con éxito a algunas industrias químicas.

En 1958, un científico de IBM llamado Friedberg y sus colaboradores procuraron solucionar problemas simples evolucionando pequeños programas –sólo 64 instrucciones– utilizando mutaciones sobre las instrucciones de 14 bits diseñadas especialmente. Aunque Friedberg nunca mencionó explícitamente estar simulando la evolución natural, esa es la interpretación más comúnmente aceptada de su trabajo. Según Coello [Coello97], a pesar de que tuvo cierto éxito, la comunidad de Inteligencia Artificial de la época no prestó atención a su trabajo, y en algunas ocasiones lo atacó duramente. Por ejemplo, Minsky lo criticó, argumentando que una búsqueda puramente aleatoria era mejor que el algoritmo de Friedberg.

También en el año 1958 Bremmerman empezó a considerar la evolución como un proceso de aprendizaje. Intentó resolver problemas de minimización de funciones lineales y convexas. Introdujo el importante manejo de un valor de aptitud, y definió a un individuo como una cadena de símbolos binarios. Advirtió, acertadamente, que la mutación jugaba un papel importante en la evolución, pues impedía el estancamiento en mínimos locales. Aunque no obtuvo buenos resultados, sus aportes han sido de gran valía. Su error fue el dominio de problemas elegido en el que los algoritmos evolutivos difícilmente pueden competir con las técnicas tradicionales de optimización.

Más tarde Reed, Toombs y Baricelli, evolucionaron un juego de cartas simplificado en el que desvalorizaron la importancia del crossover¹ –entrecruzamiento– y el operador primordial fue la mutación.

¹ Contrariamente a lo que ocurre en los Algoritmos Genéticos donde el crossover es el operador principal y la mutación ocupa un lugar secundario. Para un análisis del mismo desde el punto de vista biológico ver “La meiosis” –pág. 26–, más adelante en este capítulo se presenta desde el punto de vista de la Computación Evolutiva –pág. 49–.

Los posteriores trabajos realizados en la década del 60 ya corresponden a los algoritmos evolutivos modernos, y se han seguido investigando hasta nuestros días. Algunos de ellos son simultáneos a los Algoritmos Genéticos –de los que se ocupa en gran parte este capítulo–, pero se desarrollaron sin conocimiento unos de otros. Uno de ellos, la programación evolutiva de Fogel, se inició como un intento de usar la evolución para crear máquinas inteligentes, utilizando a tal efecto un autómeta celular.

Fogel trataba de hacer aprender a estos autómetas a encontrar regularidades en los símbolos que se le iban enviando. Una población de diferentes autómetas competía para hallar la mejor solución prediciendo cual iba a ser el siguiente símbolo de la secuencia. Los peores 50% eran eliminados de cada generación, y sustituidos por otros autómetas resultantes de una mutación de los existentes¹. De esta forma, se evolucionaron autómetas que predecían algunos números primos. Estos primeros experimentos demostraron el potencial de la evolución como método de búsqueda de soluciones novedosas.

A mediados de los años 60, Rechenberg y Schwefel describieron las “estrategias de evolución” como métodos de optimización paramétricos, que trabajan sobre poblaciones de cromosomas compuestos por números reales.

A fines de los años 60 John Holland desarrolló los planes evolutivos que se popularizaron con el nombre de Algoritmos Genéticos en el libro “*Adaptation in Natural and Artificial Systems*” publicado en 1975. Mientras que las técnicas antes mencionadas utilizan la mutación como motor principal de la evolución, los algoritmos de Holland se valen del crossover como operador principal siendo la mutación un operador secundario. No obstante, ambos operadores cumplen una función importante, consiguiéndose mejores soluciones al ofrecer un buen equilibrio entre la exploración y explotación del espacio de búsqueda (ver “Explotación vs. Exploración” –pág. 74–).

Aunque los Algoritmos Genéticos no garantizan encontrar la solución óptima, la experiencia acumulada hasta la fecha parece demostrar que, cuando se utilizan apropiadamente, pueden proporcionar soluciones muy aceptables y, en la mayoría de los casos, superiores a las encontradas con otras técnicas de búsqueda y optimización.

Aunque aún atacados por algunos sectores de la comunidad de Inteligencia Artificial, la efectividad en los resultados de los Algoritmos Genéticos los han hecho merecedores de reconocimiento como métodos viables para resolver problemas de búsqueda complejos.

Algoritmos Genéticos

Los Algoritmos Genéticos son métodos adaptativos muy flexibles y robustos que pueden usarse para resolver problemas de búsqueda y optimización.

Trabajan con una población de individuos –representados por sus cromosomas–, cada uno de los cuales constituye una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución –valor de fitness–. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material

¹ A este tipo de selección, en que los mejores individuos siempre pasan a la siguiente generación, hoy en día se la denomina elitismo. Para más información al respecto ver “Elitismo” –pág. 72–.

genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos –descendientes de los anteriores– los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones¹.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados –mejores soluciones–, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el Algoritmo Genético ha sido bien diseñado, la población convergerá, idealmente, hacia una solución óptima del problema.

El poder de los Algoritmos Genéticos proviene del hecho de que se trata de una técnica robusta –efectiva sin perder generalidad–, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. El hecho de trabajar sobre los cromosomas –genotipo²–, en lugar de hacerlo con el fenotipo, permiten una gran independencia del problema. Es más, podría implementarse un algoritmo genético casi sin conocimiento de cuál es la cuestión a resolver, simplemente manipulando el material genético de una población de cromosomas. Este hecho es también el responsable, según Goldberg [*Goldberg89*], de su eficiencia, ya que al operar en el nivel de código genético, son difíciles de engañar aun cuando la función que codifican los cromosomas sea difícil para los enfoques tradicionales.

Los Algoritmos Genéticos procesan similitudes en el genotipo junto con información sobre las capacidades de supervivencia en el entorno actual del fenotipo correspondiente. Al explotar una información tan fácilmente disponible, los Algoritmos Genéticos se pueden aplicar en prácticamente cualquier problema.

El vínculo con el problema particular a resolver está representado por la codificación genética adoptada, y prácticamente condensado en la implementación de una rutina capaz de construir el fenotipo a partir de la representación cromosomática. Muchas veces esta rutina forma parte de la función de evaluación o fitness, que debe ser diseñada especialmente para cada problema a resolver. Al ser la encargada de asignarle un valor de utilidad a cada solución representada por los cromosomas de la población, debe ser capaz de construir el fenotipo a partir de los cromosomas para así poder calificarlos.

Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima, del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

¹ Nótese que siempre se habla en términos de probabilidades, cabría esperar que un individuo mal adaptado no tenga ninguna chance de reproducirse. Esta sería una decisión desacertada, ya que ciertas particularidades de un individuo mal adaptado combinado con otros podría ser muy útil, quizá la llave para salir de una solución subóptima podría estar en algún segmento de un individuo con bajo fitness.

² Los algoritmos genéticos y la Computación Evolutiva en general, importan términos provenientes de otras áreas como la Biología y la Genética, estos términos ya han sido explicados en el capítulo anterior “Evolución y genética. Una mirada desde la biología”, recomendándose su lectura previa. Para un repaso rápido puede consultarse el apartado “Genética mendeliana” –pág. 19– que concentra gran parte de la terminología.

Representación –Codificación Genética–

Como se dijo anteriormente, la codificación de las potenciales soluciones en forma del material genético que manipulará el algoritmo, exige al diseñador del mismo concentrarse en las particularidades del problema a resolver.

Se asume que la solución potencial a un problema puede ser representada por un conjunto de parámetros [Beasley93]. Estos parámetros no serán otra cosa que los genes, los cuales, colocados uno a continuación de otro conformarán una secuencia de valores que, apropiadamente, recibe el nombre de *cromosoma*.

Existen varias cuestiones relacionadas con la codificación. Así como la Naturaleza utiliza un alfabeto de cuatro símbolos¹ para describir completamente un organismo, el diseñador de un algoritmo genético deberá adoptar también un alfabeto para codificar a los individuos –las posibles soluciones– como cromosomas. Holland y Goldberg aconsejan utilizar uno de baja cardinalidad, en lo posible binario. No obstante, hay también una tendencia fuerte a la utilización de otros alfabetos y existe un debate planteado al respecto. Más adelante en el apartado titulado “Codificación” –pág.65– se retomará el tema para ampliar este punto y presentar alternativas.

Crossover y Mutación

Para simular la reproducción sexual los Algoritmos Genéticos utilizan el crossover como operador principal y la mutación como secundario. Esto se debe a que la primera técnica favorece una rápida exploración en el espacio de búsqueda, mientras que la mutación provee una pequeña cantidad de búsqueda aleatoria ayudando a asegurar que ningún punto del espacio tenga probabilidad cero de ser analizado.

El crossover es el principal operador genético, hasta el punto que se puede decir que no es un algoritmo genético si no tiene crossover, y, sin embargo, puede serlo perfectamente sin mutación. No obstante, no es aconsejable diseñar un algoritmo genético sin algún tipo de mutación, ya que este operador permite alcanzar puntos del espacio de búsqueda que serían inaccesibles si no se implementara alguna forma de mutación. Ver “Más acerca de la Mutación” –pág. 57–.

Generalmente ambas operaciones se llevarán a cabo de acuerdo a una probabilidad de ocurrencia fijada previamente en el algoritmo.

El crossover es la operación más importante en el cruzamiento sexual, por la cual se realiza la importante acción de recombinación genética. Ya se mencionó en el capítulo anterior, que la evolución natural se vio acelerada considerablemente a partir del momento en que comenzó la reproducción sexual, permitiendo una proliferación enorme de nuevas formas vivas.

Es así que, inspirado en el entrecruzamiento de cromosomas que suele ocurrir en la división celular conocida con el nombre de meiosis² los Algoritmos Genéticos han adoptado el crossover como el operador más característico de esta familia de algoritmos.

¹ Las cuatro bases nitrogenadas Adenina (A), Citosina (C), Guanina (G), Timina (T) que componen los nucleótidos del ADN permiten especificar un organismo por medio de una palabra del conjunto {A,C,G,T}*. Para más información acerca de la estructura del ADN ver “Genes y cromosoma” –pág. 32–.

² En el capítulo anterior, bajo el apartado “La meiosis” –pág. 26– puede verse una explicación de este tipo de división celular, en donde a partir de una célula con número diploide de cromosomas se originan 4 células sexuales (número haploide de

La forma más básica de crossover se describe a continuación y corresponde al denominado crossover de un punto. Más adelante se presentarán otras formas de este operador (“Más acerca del Crossover” –pág. 56–).

En las células que se dividen por meiosis el entrecruzamiento se realiza a partir de un puente –chiasma– formado en un punto al azar a lo largo de los cromosomas homólogos. Este puente se produce a la misma altura en ambos cromosomas, así cada segmento homólogo posee la misma longitud. En el contexto de los Algoritmos Genéticos, el chiasma suele llamarse simplemente punto de cruce y también es generado al azar.

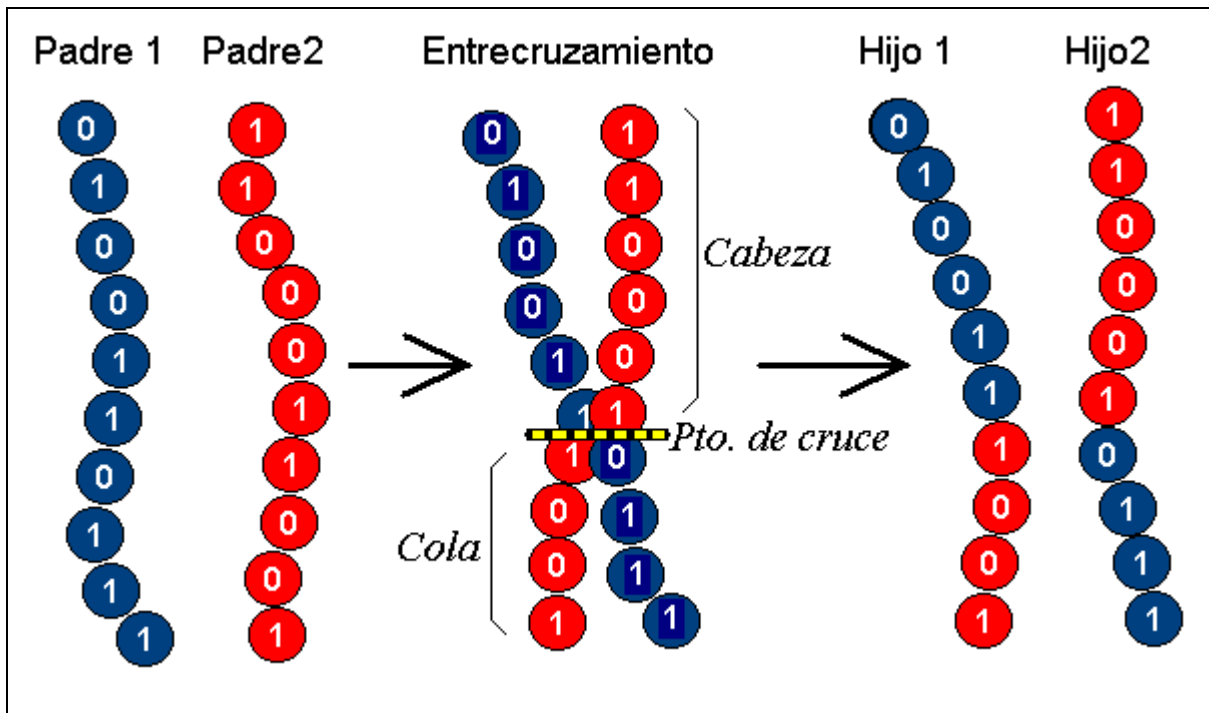


Ilustración 18 Crossover de un punto.

A partir del punto de cruce cada cromosoma padre se divide en dos segmentos: *cabeza* y *cola*. Cada segmento en un cromosoma posee la misma longitud del segmento correspondiente en el otro. Luego se intercambian los segmentos *cola* obteniendo así dos cromosomas hijos distintos.

En la naturaleza, durante el proceso de división meiótica, no siempre se produce el entrecruzamiento. En los Algoritmos Genéticos tampoco. Una vez elegidos los parentales, una elección probabilística determina si se cruzan o no. Esta probabilidad es fijada de antemano y típicamente varía entre 0.6 y 1.0. Si el crossover no se aplica, la descendencia se produce simplemente como copia de los padres. Esto da la posibilidad a cada individuo de pasar sus genes sin ninguna alteración a la próxima generación.

El otro operador importante en la reproducción es la mutación. Ésta se aplica –también en función de una probabilidad– a cada cromosoma hijo luego del crossover.

Consiste simplemente en la alteración de un alelo elegido al azar. En caso de una codificación binaria consiste simplemente en cambiar el valor 0 por 1 ó viceversa. Si no fuese una

cromosomas) en dos divisiones consecutivas. Los cromosomas homólogos (apareados y provenientes de distintos parentales) se entrecruzan en la primera división meiótica.

codificación binaria habrá que definir en qué consiste una mutación y el rango aceptable de valores al que se puede mutar.

Generalmente se implementa recorriendo uno a uno todos los genes de los cromosomas hijos alterándolos con una probabilidad muy baja, típicamente cercana a 0.001

Ejemplo de Algoritmo Genético Simple

A continuación se presenta un ejemplo simplificado de un algoritmo genético basado en Goldberg [Goldberg89]. Luego se generalizará la explicación describiendo la estructura básica de cualquier algoritmo genético junto a sus posibles variantes o extensiones.

Se pretende encontrar el valor de x que haga máxima la función $f(x) = x^2$ sobre los enteros $\{1, 2, \dots, 32\}$. Obviamente este problema puede resolverse fácilmente con un método de búsqueda exhaustivo –por la baja cardinalidad del espacio de búsqueda–, o analíticamente, pero resultará ilustrativo abordarlo con un algoritmo genético para clarificar su comportamiento.

Se comienza con la inicialización de la población, una vez determinado el tamaño de la misma, se procede a construirla generando los cromosomas aleatoriamente.

Se utilizará una codificación binaria, por lo tanto se construyen los cromosomas como una secuencia de 5 bits suficientes para representar los 32 puntos del espacio de búsqueda.

Para este ejemplo trivial se utilizará una población de tan solo 4 individuos. Supongamos que aleatoriamente –arrojando una moneda 20 veces– se generan los siguientes cromosomas: [01101]; [11000]; [01000] y [10011].

Se requiere ahora conseguir el valor de fitness de cada uno. Se debe encontrar una función que dado un cromosoma, devuelva una medida de que tan bien se adecua el individuo que representa a la resolución del problema. En este caso la función de fitness es trivial, simplemente se corresponde con la función a maximizar ($f(x) = x^2$).

En este ejemplo presentado por Goldberg [Goldberg89] es importante conocer cómo se reparte el fitness total logrado entre los individuos de la población, ya que será relevante al momento de seleccionar los parentales que se cruzarán. Para ello se presenta la siguiente tabla correspondiente a la población inicial considerada.

Generación 1

Nº	String	Valor x	Fitness x^2	% del total de fitness
1	01101	13	169	14.4
2	11000	24	576	49.2
3	01000	8	64	5.5
4	10011	19	361	30.9
Total			1170	100.0

Como puede verse el individuo 2 es quien posee el mayor fitness, consiguiendo por sí mismo casi la mitad del total alcanzado por la población. Esta información será utilizada para seleccionar los individuos a reproducirse. Presumiblemente los mejores tendrán más chance de ser elegidos más veces.

Goldberg en su libro utiliza para la selección el método denominado *la ruleta*. Este método es muy simple, y consiste en crear una ruleta en la que cada cromosoma tiene asignada una fracción proporcional a su valor de fitness. De la tabla anterior se puede conseguir la siguiente ruleta:

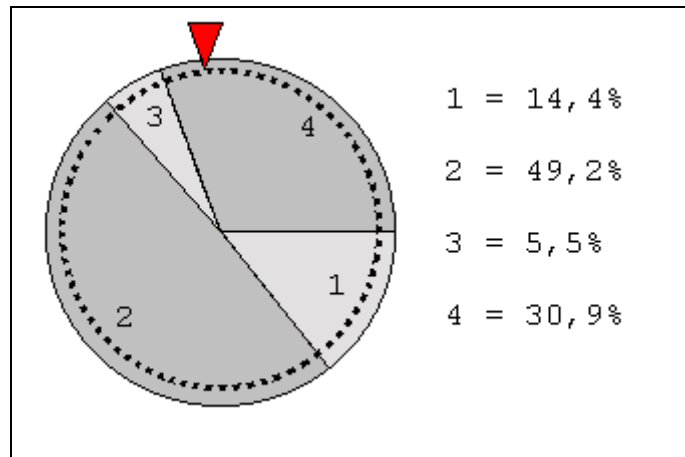


Ilustración 19 Rueda de ruleta. Selección proporcional al valor de fitness.

Se acciona la ruleta dos veces para elegir dos individuos –nótese que podría elegirse al mismo individuo ambas veces–. Una vez elegidos ambos parentales se reproducen obteniendo dos descendientes. Esta operación se repite hasta obtener tantos descendientes como cantidad de individuos haya en la población –se asume número par de individuos–. Una vez terminada esta operación se reemplaza completamente la población por la nueva descendencia producida.

Siguiéndose con el ejemplo supóngase que al accionar 4 veces la ruleta se obtienen los siguientes resultados en el orden indicado: 2,1,4,2. Esto significa que se cruzará el individuo 2 con el 1 y luego el 4 con el 2.

Aquí entra en juego un parámetro del algoritmo genético fijado previamente: la probabilidad de cruce. Un valor aceptable es por ejemplo 0.8 lo que indica que de 10 parejas elegidas 8 serán cruzadas para obtener su descendencia, mientras que 2 pasarán intactas a la población de descendientes.

Suponiendo en el ejemplo que se deben cruzar ambas parejas se elige al azar dos números entre 1 y 4 ($\text{longitud_Cromosoma} - 1$) para determinar el punto de cruce para los dos casos. Imaginemos que los números elegidos son 4 y 2, el crossover se realiza de la siguiente manera:

PadreID	Cromosoma	Pto. cruce		Hijos
2	11000	4	1100 0	11001
1	01101		0110 1	
PadreID	Cromosoma	Pto. cruce		Hijos
4	10011	2	10 011	10000
2	11000		11 000	

El proceso de cruce se completa con la operación de mutación aplicada con baja probabilidad a los bits en los cromosomas hijos. Supóngase en este ejemplo que ningún bit resultó seleccionado para la mutación.

La reproducción termina cuando la descendencia es tan numerosa como la población, en el ejemplo 4. Ahora se reemplaza íntegramente la población por los descendientes obtenidos arribando a la siguiente situación

Generación 2

Nº	String	Valor <i>x</i>	Fitness <i>x</i> ²	% del total de fitness
1	11001	25	625	35.6
2	01100	12	144	8.2
3	10000	16	256	14.6
4	11011	27	729	41.6
Total			1754	100.0

El ciclo se repite hasta que se cumpla alguna condición de finalización. Habitualmente no se conoce cuál es la mejor solución. Normalmente se usan 2 criterios principales de detención: correr el algoritmo genético durante un número máximo de generaciones o detenerlo cuando la población se haya estabilizado, es decir, cuando la mayoría de los individuos sean muy similares entre sí.

Luego de este pequeño, pero ilustrativo ejemplo, sobre cómo se comporta un simple algoritmo genético, se presenta a continuación el esqueleto básico de un algoritmo genético general, a partir del cual se explicitarán algunas variantes y extensiones.

Anatomía de un Algoritmo Genético

El esquema general de un algoritmo genético es sorprendentemente simple:

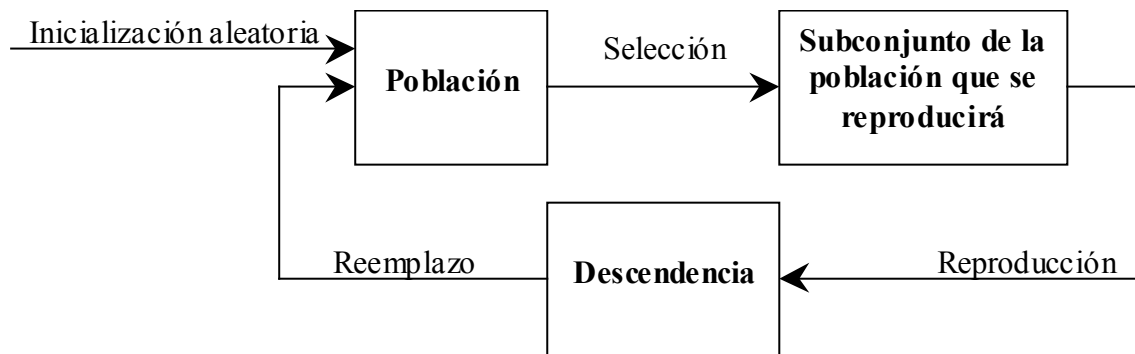


Ilustración 20 Esquema general de Algoritmo Genético

Pseudocódigo
<pre> Begin Inicializar(Poblacion); While not condicion_de_terminación do Begin EvaluarFitness(Poblacion); Parentales := seleccionar(Poblacion); Descendencia := reproduccion(Parentales); Poblacion := reemplazo(Poblacion, Descendencia); End; end; </pre>

Por lo general se comienza con una población de cromosomas generados al azar y se itera el bucle de *selección*, *reproducción* y *reemplazo* hasta que se cumpla alguna condición de finalización, pudiendo ser esta un número máximo de generaciones –muy utilizado–, un período de tiempo fijo, después de encontrar una solución de determinada calidad, cuando el

algoritmo ha convergido, es decir, un gran porcentaje de la población tiene características similares, o cuando el algoritmo no genera mejores soluciones durante una determinada cantidad de generaciones.

La selección se realiza en base a la función de fitness que califica las potenciales soluciones para que, aquellas con valores más altos, tengan mayor probabilidad de reproducirse más veces. Los operadores genéticos infaltables en la reproducción serán el crossover y la mutación. Por último, existen varios métodos de reemplazo para determinar quienes serán sustituidos por la nueva descendencia lograda para conformar la siguiente generación.

A continuación se verán más en detalles cada una de las fases por la que pasa un algoritmo genético.

Inicialización de la población

Una cuestión que uno puede plantearse es la relacionada con el tamaño idóneo de la población. Parece intuitivo que las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que el trabajar con poblaciones de gran tamaño puede acarrear problemas relacionados con el excesivo costo computacional.

Goldberg efectuó un estudio teórico, obteniendo como conclusión que el tamaño óptimo de la población para strings de longitud L , con codificación binaria, crece exponencialmente con el tamaño del string.

Este resultado traería como consecuencia que la aplicabilidad de los Algoritmos Genéticos en problemas reales sería muy limitada, ya que resultarían no competitivos con otros métodos de optimización combinatoria. No obstante, otros autores han concluido, basándose en evidencia empírica que un tamaño de población comprendida entre L y $2*L$ es suficiente para atacar con éxito una amplia gama de problemas. [Larrañaga]

Habitualmente la población inicial se escoge generando cromosomas al azar, pudiendo contener cada gen uno de los posibles valores del alfabeto con probabilidad uniforme. Otra opción es utilizar alguna técnica heurística para crear estos primeros individuos, esto es una forma de introducir en el algoritmo información sobre el problema particular a tratar. En general esta técnica acelera la convergencia del algoritmo hacia la solución. Sin embargo en algunos casos la desventaja resulta ser la prematura convergencia del algoritmo hacia óptimos locales –subóptimos–.

Selección

El proceso de selección consiste en la elección de un subconjunto de individuos de la población con fines reproductivos. La idea aquí es emular la selección natural que elegantemente expusiera Darwin y Wallace en su libro “*El origen de las especies*” –1859–. El principio de esta teoría –expuesta a lo largo del capítulo anterior y sobre todo en el apartado “Darwinismo” en la pág. 7– que interesa imitar aquí es el que sostiene que los individuos mejor adaptados son quienes tienen mayor probabilidad de reproducirse, consecuentemente serán los que más chance tengan de transmitir sus genes a la próxima generación.

Para poder llevar a cabo esta tarea de *selección artificial* en el algoritmo genético es necesario medir de alguna manera que tan adaptados están los individuos de la población. Dicho en otras palabras, qué tan buenos son como soluciones al problema que intentan resolver. Por lo tanto es necesario evaluar cada uno de los individuos de la población para calificarlos, acción que se realiza por medio de la función de fitness –también llamada función objetivo–

Función de fitness

Durante la evaluación previa a la selección, se decodifican los cromosomas, se obtienen entonces los parámetros que representan a la potencial solución del problema. Se le da una puntuación a esa solución en base a qué tan bien lo resuelve. A esa puntuación se le llama fitness.

La función de fitness debe ser diseñada especialmente para cada problema a resolver ya que de alguna manera está vinculando dos dominios diferentes, los códigos por un lado –genotipo– y la interpretación de los mismos por el otro –fenotipo–.

La determinación de una adecuada función de fitness o función objetivo es crucial en el comportamiento del algoritmo genético. Por cada cromosoma de la población debe devolver un valor numérico proporcional a la habilidad del individuo –utilidad de la solución– que representa dicho cromosoma.

Para muchos problemas la función de fitness es trivial, pues surge directamente de un único parámetro que se desea maximizar. No obstante este no es el caso más general, sino por el contrario, la mayor parte de las veces se necesita decidir qué parámetros y en qué porcentajes deben favorecer o perjudicar el valor de fitness que se otorgará.

En el caso de un problema en el que se desea minimizar algún parámetro, es conveniente, para no perder coherencia con la terminología, que el fitness aún sea proporcional a la utilidad de la solución, es decir, debe ser mayor cuando más chico sea el parámetro a minimizar.

Según Goldberg, también es conveniente que la función de fitness sea no negativa, propiedad de la que goza la definición de fitness biológica que no es otra que el número posible de descendientes de un individuo –que no siempre es el mismo que el número *real* de descendientes–.

Idealmente se deberían construir funciones objetivo con “ciertas regularidades”, es decir funciones objetivo que verifiquen que para dos individuos que se encuentren cercanos en el espacio de búsqueda, sus respectivos valores en las funciones objetivo sean similares. Por otra parte una dificultad en el comportamiento del Algoritmo Genético puede ser la existencia de gran cantidad de óptimos locales, así como el hecho de que el óptimo global se encuentre muy aislado.

Una técnica que se ha venido utilizando en el caso en que la computación de la función de fitness sea muy compleja es la denominada evaluación aproximada de la misma. En algunos casos la obtención de n funciones objetivo aproximadas puede resultar mejor que la evaluación exacta de una única función objetivo –supuesto el caso de que la evaluación aproximada resulta como mínimo n veces más rápida que la, evaluación exacta–.

Distintas estrategias de selección

Una vez evaluados y calificados con su fitness, un subconjunto de individuos es seleccionado de la población para obtener la descendencia que, por medio de un apropiado reemplazo contribuirá a la conformación de la siguiente generación.

En función del fitness se determinan los cromosomas que se van a reproducir –también aquellos que se van a eliminar en la etapa de reemplazo–, esta selección, y la consiguiente reproducción, se puede hacer de tres formas principales:

Basado en ranking. En este esquema se mantiene un porcentaje de la población para la siguiente generación. Se coloca toda la población por orden de fitness. Se obtienen M descendientes por reproducción de los N individuos mejor ranqueados. La descendencia reemplazará a los M peor calificados formando así la próxima generación. Una variante se

utiliza en SANE¹ donde una vez ranqueados se seleccionan uno a uno los individuos del 25% superior para cruzarse con individuos seleccionados al azar también de este 25% superior, reemplazándose con su descendencia el 50% inferior para dejar así constituida la población de la próxima generación.

Rueda de ruleta. Corresponde a la metodología explicada en el apartado “Ejemplo de Algoritmo Genético Simple” –pág. 51–. Es una de las formas de selección de padres más utilizada, y en la cual cada individuo tiene una probabilidad de ser seleccionado proporcional al valor de su función objetivo. Con esta metodología es necesario evitar que aparezcan superindividuos cuyo fitness sean muy superior a todos los demás, ya que ocupará un sector muy ancho de la ruleta, será elegido muchas veces provocando la pérdida de diversidad genética –ver “Diversidad Genética y Convergencia” en la pág. 64– ocasionando seguramente una convergencia prematura a un subóptimo local. Por ello suele escalarse la función de fitness para atenuar estas diferencias. Otra forma de atacar el problema es fijar un máximo de veces en que un individuo puede ser seleccionado para cruzarse, llegado el caso, dicho individuo se elimina del conjunto de selección.

Selección de torneo. La selección por torneo, constituye también un procedimiento de selección de padres muy extendido y en el cual la idea consiste en escoger al azar un número T de individuos de la población, donde T es el tamaño del torneo. Se selecciona el mejor individuo de este grupo, y se repite el proceso –nuevo torneo– para seleccionar el segundo padre, una vez elegidos los dos se reproducen obteniéndose dos descendientes. Habitualmente –al igual que con la rueda de ruleta– se consiguen tantos descendientes como tamaño de la población para reemplazarla íntegramente, no obstante puede haber variantes. Habitualmente el tamaño del torneo es 2, y en tal caso se ha utilizado una versión probabilística en la cual se permite la selección de individuos sin que necesariamente sean los mejores.

Reproducción

Ya se ha presentado en este capítulo una breve descripción de los operadores genéticos más utilizados en la fase de reproducción, con el objeto de simular la manera en que los organismos recombinan su material genético –Ver “Crossover y Mutación” en la pág. 49–. Ahora se presentarán algunas variantes.

Más acerca del Crossover

El Algoritmo Genético Simple descrito anteriormente utiliza crossover de un punto, en el cual los dos individuos seleccionados como padres, son recombinados por medio de la selección de un punto de corte, para posteriormente intercambiar las secciones que se encuentran a la derecha de dicho punto. Este tipo de crossover es el más común en la Naturaleza. –Ver “Recombinación genética” en la pág. 36–.

Se han investigado otras formas de crossover, habitualmente teniendo en cuenta más de un punto de cruce. La ventaja radica en que el espacio de búsqueda puede ser explorado más fácilmente, siendo la principal desventaja el hecho de aumentar la probabilidad de ruptura de buenos esquemas².

¹ SANE son las siglas de Symbiotic adaptative neuroevolution. Es un método para obtener Redes Neuronales evolucionando dos poblaciones simultáneamente, una de neuronas y otra de blueprints que describen las redes utilizando individuos de la primera población. Para más información ver “SANE –Symbiotic Adaptive NEuroevolution–” en la pág. 201.

² Por el momento considérese un esquema como una máscara que define un conjunto de cromosomas que poseen el mismo valor en ciertas posiciones, más adelante se formalizará el concepto (ver “Fundamento teórico” –pág. 59–).

Crossover de 2 puntos. Se eligen dos puntos de cruce al azar quedando así ambos parentales divididos en tres segmentos. Se consiguen dos cromosomas hijos intercambiando el segmento medio entre los padres. Ejemplo:

Parentales	Ptos. de cruce		Hijos
11000110	2,5	11 000 100	11101110
01101001		01 101 001	01000001

Nótese que el crossover de un punto puede verse como un caso particular del crossover de 2 puntos, donde uno de los puntos se encuentra fijo al final del cromosoma

Crossover de n puntos. El operador de crossover puede fácilmente generalizarse a n puntos de la siguiente manera: Se eligen al azar n puntos que dividirán a los cromosomas padres en $n+1$ segmentos. Se numeran los segmentos y se obtiene dos descendientes intercambiando los segmentos pares.

Se ha concluido experimentalmente que el cruce basado en dos puntos, representa una mejora mientras que añadir más puntos de cruce no beneficia el comportamiento del algoritmo. [Larrañaga]

Crossover uniforme. Esta es otra variante en la que se genera un patrón aleatorio de 1s y 0s, y se intercambian los bits de los dos cromosomas que coincidan donde hay un 1 en el patrón. O bien se genera un número aleatorio para cada bit, y si supera una determinada probabilidad se intercambia ese bit entre los dos cromosomas. Ejemplo –con máscara–:

Parentales		Hijos
Padre 1	1 1 0 0 0 1 1 0	0 1 0 0 0 0 0 0
Máscara	1 0 0 1 0 1 1 0	
Padre 2	0 1 1 0 1 0 0 1	1 1 1 0 1 1 1 1

Crossover especializados. En algunos problemas, aplicar aleatoriamente el crossover da lugar a cromosomas que codifican soluciones inválidas; en este caso se puede aplicar el crossover de forma que genere siempre soluciones válidas, adecuando este operador a las particularidades del problema. Como se verá más adelante, otra forma de resolver el problema de las soluciones inválidas es a través de una buena codificación. Por otra parte, la idea de que el cruce debería ser más probable en algunas posiciones del cromosoma ha sido descrita por varios autores, entre ellos el propio Holland.

Más acerca de la Mutación

Como se explicara anteriormente, la forma más común de mutación es la alteración de un alelo elegido al azar. No obstante pueden implementarse otros tipos.

Puede consultarse el apartado “Mutaciones” –pág. 37– del capítulo anterior, donde se exponen unas cuantas clases de mutaciones que sufren los seres vivos. No es difícil adaptar alguna de ellas –sobre todo las llamadas mutaciones estructurales– en la concreción de un algoritmo genético.

Goldberg en su libro [Goldberg89] presenta varias formas de mutación: inversión, reordenamiento, duplicación, delección que se comportan de la forma esperada emulando el comportamiento biológico de estas alteraciones genéticas.

Si bien los Algoritmos Genéticos admiten al crossover como principal responsable de efectuar la búsqueda a lo largo del espacio de posibles soluciones, también parece desprenderse de los experimentos efectuados por varios investigadores que el operador de mutación va ganando

en importancia a medida que la población de individuos va convergiendo. Esto se debe a que el crossover pierde eficacia entre individuos muy similares.

Según Larrañaga [*Larrañaga*] Schaffer postula que la denominada evolución primitiva, en la cual, el proceso evolutivo consta tan sólo de selección y mutación supera con creces a una evolución basada exclusivamente en la selección y el crossover. Otra conclusión de su trabajo es que la determinación del valor óptimo de la probabilidad de mutación es mucho más crucial que el relativo a la probabilidad de cruce.

La primera conclusión de Schaffer no debe sorprender. Si no hay mutación, no pueden aparecer nuevos genes. De todas maneras un ejemplo sencillo muestra que la mutación es imprescindible pues permite la exploración en regiones del espacio de búsqueda que serían inalcanzables sin ella. Supongamos la siguiente población de 4 cromosomas:

[01101000] [00101100] [01101110] [00001100]

Claramente existen puntos en el espacio de búsqueda que nunca serán examinados, si sólo se aplica crossover. Obsérvese que jamás podrá obtenerse ningún cromosoma que posea un 1 en cualquiera de sus extremos. Esto deja claramente establecida la necesidad de algún mecanismo de mutación. De hecho la evolución natural se impulsó en sus comienzos sólo por mutación –hace 4.000 millones de años–, mientras que el mecanismo del crossover fue inventado por la Naturaleza hace a penas 2.000 millones de años –Ver “Los orígenes de la vida y el comienzo de la evolución” en la pág. 3–.

La búsqueda del valor óptimo para la probabilidad de mutación, es una cuestión que ha sido motivo de varios trabajos. Así, De Jong recomienda la utilización de una probabilidad de mutación del bit de $(1/L)$, siendo L la longitud del string. Si bien en la mayoría de las implementaciones de Algoritmos Genéticos se asume que tanto la probabilidad de cruce como la de mutación permanecen constantes, algunos autores han obtenido mejores resultados experimentales modificando la probabilidad de mutación a medida que aumenta el número de iteraciones.

No hace falta decir que no conviene abusar de la mutación. Es cierto que es un mecanismo generador de diversidad y, por tanto, la solución cuando un algoritmo genético está estancado, pero también es cierto que reduce al algoritmo genético a una búsqueda aleatoria. Siempre es más conveniente usar otros mecanismos de generación de diversidad, como aumentar el tamaño de la población, o garantizar la aleatoriedad de la población inicial.¹

Reemplazo

Una vez obtenidos los individuos descendientes de una determinada población debe seleccionarse los individuos que serán reemplazados por esta descendencia para construir así la población de la próxima generación. Aquí se realiza un análisis para el caso de Algoritmos Genéticos tradicionales, con número de población constante. Existen variantes donde se permite que el tamaño de la población crezca o disminuya con el tiempo.

Supongamos que se tiene una población de N individuos y a través del proceso de selección y reproducción se generan n descendientes. El proceso de reemplazo puede hacerse de varias formas distintas:

¹ Puede consultar el apartado “Diversidad Genética y Convergencia” –pág. 64– para más información al respecto.

- Los individuos de la próxima generación se forman con los n descendientes, es decir se reemplaza completamente la población –sólo válido para $n=N$ –. Este es el enfoque presentado en “Ejemplo de Algoritmo Genético Simple” –pág. 51–.
- Los n descendientes sustituyen a sus respectivos progenitores. Para mantener el tamaño de la población constante debe darse el caso que los n descendientes se hayan formado por cruza entre un grupo de n parentales.
- Los n descendientes sustituyen a aquellos miembros de la población que más se les parezcan. Esto se denomina *crowding* y favorece la diversidad genética y aparición de especies –Ver más adelante “Diversidad Genética y Convergencia” en pág. 64–.
- Se juntan los n descendientes con los m progenitores en una sola población, y en ella se seleccionan N individuos –normalmente los mejores– para formar la próxima generación, los restantes son eliminados.
- Se seleccionan n miembros de la población –generalmente los de más bajo fitness– para ser reemplazados por los n descendientes –sólo válido para $n \leq N$ –
- Se seleccionan los N miembros mejor ranqueados del conjunto de descendientes y con ellos se reemplaza a toda la población –sólo válido para $n \geq N$ –.

Nótese que las dos últimas alternativas constituyen la primera si $n=N$.

Además de estas pueden existir infinidad de variantes todas ellas con comportamiento similares. Por ejemplo, y sólo a modo ilustrativo, Michalewicz introduce un algoritmo que denomina Algoritmo Genético Modificado, en el cual para llevar a cabo el reemplazo generacional, selecciona al azar n_1 individuos para la reproducción, y n_2 individuos –distintos de los primeros– para ser reemplazados. Estas selecciones aleatorias tienen en consideración el valor de fitness de cada individuo, de tal manera que cuanto mayor es su fitness, mayor es la probabilidad de que sea seleccionado para la reproducción, y menor es la probabilidad de que dicho individuo sea seleccionado para el reemplazo. A partir de los n_1 parentales se obtienen los n_2 descendientes que reemplazarán al grupo elegido para ello.

Fundamento teórico

Teorema fundamental –Teorema de esquemas

Proporciona el fundamento teórico de porqué los Algoritmos genéticos pueden resolver diversos problemas. En su análisis se considera el proceso de selección –proporcional al fitness– y los operadores crossover de un punto y mutación –que altera un sólo gen–.

Sin perder generalidad se considerará un alfabeto binario $V=\{0,1\}$. Para definir los esquemas es necesario agregar un nuevo símbolo (*) para representar un comodín que puede aparear ambos valores 0 ó 1. Por ejemplo el esquema *101* describe un subconjunto de cuatro miembros {01010,01011,11010,11011}.

Obsérvese que si el alfabeto a partir del cual se definen los cromosomas es de cardinalidad k el correspondiente a los esquemas será de cardinalidad $(k+1)$ debido al símbolo “*” agregado. Así para cadenas de longitud L existirán k^L cromosomas distintos y un número mucho mayor de esquemas igual a $(k+1)^L$. Por ejemplo para un alfabeto binario y cromosomas de longitud 5 tenemos: $2^5 = 32$ cromosomas distintos y $3^5 = 243$ esquemas distintos.

Existe una relación de “*muchos a muchos*” entre los cromosomas y esquemas. Así como un esquema representa varios cromosomas, un cromosoma está representado por varios

esquemas. Por ejemplo considérese el string 11111, existen 2^5 esquemas que lo representan, puesto que se pueden construir colocando en cada posición el bit correspondiente al cromosoma o el símbolo “*”. En general cualquier string de longitud L contiene 2^L esquemas asociados que lo representan.

El efecto de la selección sobre el número esperado de esquemas en la próxima generación es fácil de determinar. Supóngase que t es la generación actual. Denotaremos:

$m(H,t)$: número de individuos en la generación t que son representados por el esquema H .

$f(H)$: fitness promedio de los individuos representados por H en la generación t .

F : fitness promedio de toda la población en la generación t .

Suponiendo que la selección de individuos para la reproducción es proporcional al fitness, y que la descendencia se consigue simplemente como copias de los parentales –sin crossover ni mutación–, la cantidad de individuos representados por un esquema H en la próxima generación puede escribirse así:

$$m(H,t+1) = m(H,t) \cdot f(H)/F \quad (1)$$

En otras palabras el número de individuos asociados a un esquema crecerá de acuerdo a la proporción del fitness promedio del esquema con el fitness promedio de toda la población.

El efecto de la Reproducción –sin crossover ni mutación– sobre el número de esquemas de una población a través de las generaciones es claro. Si su fitness promedio es inferior al fitness promedio de la población ($f(H)/F < 1$) tiende a desaparecer. Por el contrario, si $f(H)/F > 1$ entonces tiende a crecer.

El análisis hecho arriba considera solo el paso de selección del algoritmo genético, el crossover y la mutación deben ser considerados también. En este teorema se examina sólo el efecto negativo de estos operadores sobre un esquema H –probabilidad de destruirlo–. Pero antes es necesario definir dos nuevos conceptos:

El *orden* de un esquema H , denotado por $o(H)$ es simplemente el número de posiciones fijas del esquema, es decir aquellas que no poseen el símbolo “*”. Ejemplos: $o(011*1**) = 4$; $o(100**11) = 5$; $o(******) = 0$.

La *distancia de definición* de un esquema H , denotada por $d(H)$ es la distancia entre el primero y el último valor distinto de “*”. Así $d(011*1**) = 4$ porque la primera posición fija es la 1 y la última la 5. $d(**1****) = 0$ ya que tanto la primera como la última posición fija del esquema es la número 3.

Para ver como afecta el crossover de un punto a los esquemas considérese el siguiente cromosoma de longitud $L=8$ y un esquema asociado:

A= 1 0 0 1 1 0 1 0
H= * 0 * * 1 * * *

| ↑ ↑ ↑ | | | –posibles puntos de cruce–

Claramente una operación de crossover sobre A no tendrá efecto sobre el esquema H si el punto de cruce elegido es el 1, 5, 6 ó 7 (| en el esquema) pero en el caso que el punto elegido

sea 2, 3 ó 4 (↑ en el esquema), H no sobrevivirá¹. Obsérvese que la cantidad de puntos que producen la destrucción del esquema puede calcularse como $d(H)$. Existen 3 ($d(H)$), entre 7 ($L-1$) puntos que destruyen H . Si el punto de cruce es elegido al azar con distribución uniforme el esquema H será destruido con probabilidad $p_d = 3/7$, más genéricamente:

$$p_d = d(H)/(L-1).$$

Si se denota con p_c a la probabilidad de realizar crossover definida en el algoritmo genético se tiene que:

$$p_d = p_c \cdot d(H)/(L-1).$$

Inmediatamente se deduce que la probabilidad de supervivencia de H frente a la operación de crossover sobre el cromosoma A será:

$$p_s = 1 - p_c \cdot d(H)/(L-1)$$

Combinando este resultado con (1) se tiene que el número esperado de un esquema particular H en la próxima generación utilizando selección y crossover –sin mutación– será:

$$m(H,t+1) \geq [m(H,t) \cdot f(H)/F] \cdot [1 - p_c \cdot d(H)/(L-1)] \quad (2)$$

El símbolo = a sido cambiado por \geq ya que el segundo término de la derecha representa una probabilidad, no un valor determinístico.

La expresión (2) nos indica que la cantidad de esquemas H crecerá o decaerá de acuerdo a dos cosas, primero si el fitness del esquema está más arriba o abajo del promedio de toda la población, y si el esquema posee una relativamente corta o larga distancia de definición. Claramente el número de aquellos esquemas con ambas características –fitness promedio por encima de la media de la población y longitud de definición corta– crecerá en forma exponencial.

Ahora se considerará el operador de mutación. La mutación se aplica independientemente sobre cada bit con probabilidad p_m . Así la probabilidad de un alelo de no ser mutado será $(1-p_m)$. Para que el esquema H sobreviva, es necesario que no sean mutados ninguno de los alelos del cromosoma A correspondientes a una posición fija (distinta de *) en H . La cantidad de posiciones fijas de un esquema corresponde a $o(H)$ que definimos anteriormente. Por lo tanto la probabilidad de que un esquema sobreviva a la mutación será $(1-p_m)^{o(H)}$ ya que la mutación se aplica independientemente a cada posición.

Así este teorema concluye que el número esperado de copias de un esquema particular H en la próxima generación, considerando los efectos de la selección, el crossover de un punto y la mutación puntual será:

$$m(H,t+1) \geq [m(H,t) \cdot f(H)/F] \cdot [1 - p_c \cdot d(H)/(L-1)] \cdot [(1-p_m)^{o(H)}]$$

La conclusión final indica que *los esquemas pequeños de bajo orden cuyo fitness está por arriba del promedio reciben un incremento exponencial de representantes en las siguientes generaciones de un algoritmo genético.*

En definitiva, como un esquema representa una región en el espacio de búsqueda –un conjunto de strings– se ve que el algoritmo genético, explota las regiones de más alto

¹ En realidad no es cierto, si el cromosoma que está asociado a este esquema es elegido para una operación de crossover con otro cromosoma que es idéntico en las posiciones fijas del esquema, éste sobrevivirá independientemente del punto de cruce elegido. Goldberg [Goldberg89] aclara en su libro que esta situación es ignorada, textualmente: “conservatively ignore”.

rendimiento del espacio de soluciones, porque las sucesivas generaciones producen un número creciente de ellas. De hecho, el número de cadenas de una región o esquema dado, aumenta de forma proporcional a la estimación que se ha hecho del grado de adaptación de esa región.

Paralelismo implícito y la Hipótesis de construcción de bloques

Cada individuo de la población pertenece a numerosos esquemas. En el apartado anterior se explicó que cualquier string de longitud L contiene 2^L esquemas asociados con él, ya que al construirlos, en cada una de las L posiciones se tienen 2 alternativas: colocar el valor correspondiente al cromosoma o el símbolo “*”. Una población de n individuos tendrá asociada una cantidad de esquemas que varía entre 2^L y $n2^L$ dependiendo de la diversidad genética de sus miembros.

Si se asume que los individuos pertenecientes a un mismo esquema tienen características similares –y por ende una calidad parecida–, cuando se evalúa un individuo se obtiene información relativa a todos los individuos que comparten pertenencia a los mismos esquemas. Bajo este supuesto, el algoritmo no procesa individuos, sino esquemas.

Sin embargo, se sabe que no todos los esquemas son procesados ya que, dependiendo de la longitud de definición y el orden, unos tendrán una mayor probabilidad de supervivencia a los operadores de cruce y de mutación. Se ha estimado que en una generación de n individuos se procesan del orden de $O(N^3)$ esquemas¹. Esta es una importante propiedad a la que J. Holland dio el nombre de *Paralelismo Implícito*. El Paralelismo Implícito es el que hace que un algoritmo genético que manipule una población de unos cuantos strings, realmente esté tomando muestras de un número de regiones o esquemas enormemente mayor. Tal paralelismo implícito –en el sentido de procesamiento paralelo–, proporciona al algoritmo genético una ventaja sobre otros métodos de búsqueda y optimización.

De acuerdo al teorema fundamental, los esquemas de corta longitud de definición, bajo orden y cuyo fitness sea mayor al promedio comenzarán a dominar rápidamente la población a medida que avance la evolución. Admitiendo entonces que juegan un papel importante en el desarrollo del algoritmo genético Goldberg prefiere darles un nombre especial: bloques de construcción –Building Blocks–. De acuerdo a este autor el funcionamiento del algoritmo genético será óptimo si es posible combinar estos esquemas cortos para formar soluciones cada vez mejores. A esta suposición se la denomina Hipótesis de los Bloques de Construcción –*Building Block Hypothesis*–. Así, los Algoritmos Genéticos son viables para resolver los problemas cuya solución pueda formarse con bloques constructivos. Por ejemplo, si el objetivo es encontrar una cadena de n bits con mayor cantidad de 1s, el problema parece ser muy apto para un algoritmo genético ya que los bloques constructivos son los bits con valor igual a 1.

De acuerdo con Beasley [Beasley93], una buena codificación debe animar a la formación de buenos bloques de construcción asegurando que:

1. Los genes relacionados estén juntos en el cromosoma.

¹ Puede consultarse [Goldberg89] para la demostración de este resultado.

2. Exista poca interacción entre genes –epistasia–¹.

En el contexto de los Algoritmos Genéticos epistasia es cuando la contribución de un gen al fitness depende del valor de otros genes en el cromosoma. Un ejemplo donde no hay epistasia lo constituye el Algoritmo Genético Simple presentado anteriormente donde se intenta maximizar la función $f(x)=x^2$ ya que aquí la contribución de cada bit al fitness es independiente del valor de los otros cuatro. Pero habitualmente en problemas reales es imposible asegurar su ausencia.

Tampoco la primera condición suele ser fácil de conseguir. Muchas veces los genes están relacionados de una manera que hace imposible colocarlos juntos en un arreglo unidimensional.

Críticas a la Base Teórica de los Algoritmos Genéticos

Si bien durante mucho tiempo el teorema fundamental expuesto por Holland fue considerado como una explicación convincente del funcionamiento de los Algoritmos Genéticos, hoy día existe una corriente crítica que al menos, plantea unas cuantas dudas al respecto. Según Cotta Porras [CottaPorras98] entre los argumentos esgrimidos por los autores que cuestionan la validez del teorema de esquemas figuran los siguientes:

- Desconexión entre el procesamiento de esquemas y la construcción de soluciones válidas –Altenberg, 1995–: el teorema afirma que los esquemas cuyo fitness sea superior al fitness promedio de la población aumentarán su número de ocurrencias en la próxima generación. Sin embargo, esto no guarda relación directa con que dichos esquemas permitan construir mejores soluciones. Por supuesto, la hipótesis de los bloques de construcción sí garantiza esto, pero dicha hipótesis es lógicamente independiente del teorema, el cual debe seguir verificándose incluso cuando la hipótesis no es válida.
- Inadecuación de los esquemas –Radcliffe, 1992–: Los esquemas representan regiones en el espacio de búsqueda, es decir, agrupan un conjunto de soluciones. Pero sólo agrupa soluciones con similitudes léxicas en sus cromosomas, mientras que el número de posibles agrupaciones relevantes para el problema es mucho mayor que el número posible de esquemas. Por lo tanto existen subconjuntos que no pueden ser representados por ningún esquema.
- Generalidad del teorema de los esquemas: el teorema de los esquemas continúa siendo válido si H es un subconjunto arbitrario de soluciones –conjunto de cromosomas cualquiera– en lugar de ser un conjunto de strings con semejanzas léxicas, asociados con un esquema –Vose, 1991–. Por lo tanto, la caracterización de un algoritmo genético como un manipulador de esquemas, según el autor, debe ser replanteada.

Otras críticas se han hecho también en relación a la robustez de los Algoritmos Genéticos²

¹ En la ciencia de la genética se llama epistasia a un tipo de influencia que se da entre distintos genes, en la cual el valor de un gen tiene efecto sobre otro. Para más información se puede consultar el apartado “Interacciones entre genes no alélicos” –pág.41–

² Cotta Porras, en su tesis doctoral [CottaPorras98] menciona el Teorema de No Free Lunch, desarrollado por Wolpert y Macready (1995; 1997) según el cual, Holland se equivoca al concluir la superioridad de un Algoritmo Genético sobre la búsqueda aleatoria en cuanto a que el algoritmo era capaz de aprovechar de manera óptima la información que iba obteniendo sobre la distribución de los valores de fitness de los individuos examinados. Este resultado ha sido reelaborado en el Lema de Conservación (English, 1997), el cual muestra la incapacidad de explotar información referente a la distribución de los valores de fitness en el espacio de soluciones, dado que dicha distribución es mutuamente independiente de la distribución de

indicando la necesidad de introducir conocimiento del problema en consideración dentro del algoritmo de búsqueda, ya que de ello dependerá el rendimiento de éste. Esta necesidad puede parecer obvia para los investigadores de numerosos campos afines como por ejemplo la investigación operativa, pero no lo ha sido para muchos investigadores en computación evolutiva. Dicha introducción de conocimiento se ha considerado siempre como algo adicional que puede contribuir a afinar el rendimiento del algoritmo, pero no a mejorarlo sustancialmente.

Aspectos Prácticos

Diversidad Genética y Convergencia

Si un algoritmo genético está correctamente implementado, la población irá evolucionando a través de las generaciones hacia el óptimo global. Un gen se dice que ha convergido cuando el 95% de la población comparte el mismo valor. La población se dice que ha convergido cuando todos los genes han convergido [Beasley93].

La convergencia, por lo tanto, está asociada a la resolución del problema. Una vez que la población ha convergido la evolución se estanca, pues la reproducción entre organismos muy similares producen más organismos similares. En tal caso sólo puede avanzar por medio de la mutación.

Muchas veces, la convergencia de la población, se toma como condición de finalización del algoritmo ya que resulta difícil que a partir de ese momento se pueda conseguir alguna mejor solución. La convergencia está asociada a la falta de diversidad genética. Se puede decir que la diversidad genética es el combustible del algoritmo genético que permite su accionar. Por ello existen varias técnicas tendientes a conservar siempre algo de diversidad, aún cuando el algoritmo se aproxime a una solución aceptable.

Un problema habitual en las ejecuciones de los Algoritmos Genéticos surge debido a la velocidad con la que el algoritmo converge. En algunos casos la convergencia es muy rápida, lo que suele denominarse convergencia prematura, en la cual el algoritmo converge hacia óptimos locales, mientras que en otros casos el problema es justo el contrario, es decir se produce una convergencia lenta del algoritmo. Una posible solución a estos problemas pasa por efectuar transformaciones en la función de fitness. El problema de la convergencia prematura, surge a menudo cuando la selección de individuos se realiza de manera proporcional a su función objetivo. En tal caso, pueden existir individuos con una adaptación al problema muy superior al resto, que a medida que avanza el algoritmo “dominan” a la población. Por medio de una transformación de la función de fitness, en este caso una compresión del rango de variación de la función objetivo, se pretende que dichos “superindividuos” no lleguen a dominar a la población.

El problema de la lenta convergencia del algoritmo, se resolvería de manera análoga, pero en este caso efectuando una expansión del rango de la función objetivo.

los valores de fitness de las soluciones visitadas. Esto contradice la suposición realizada por Holland que los Algoritmos Genéticos consiguen un equilibrio óptimo ente la exploración y la explotación (Ver “Explotación vs. Exploración” –pág. 74).

Una técnica tendiente a mantener la diversidad es el aislamiento geográfico que se expondrá más adelante en este capítulo –pág. 73–. En el ámbito de la neuroevolución, SANE –ver en pág. 201– también asegura diversidad.

También se ha utilizado una función de compartición o *sharing*, que indica cuán similar es un cromosoma al resto de la población. La puntuación de cada individuo se dividirá por esta función de compartición, de forma que se facilita la diversidad genética y la aparición de individuos diferentes.

Los denominados método de *nicho* también están encaminados a mantener la diversidad genética de la población, y están relacionados con el concepto de especiación –forma en que aparecen distintas especies a partir de una población original–. En Biología se entiende por nicho al rol que juega un organismo en un determinado hábitat –entorno–.

Según Goldberg [Goldberg89] Cavicchio presentó en 1970 uno de los primeros estudios en el que se intenta incluir el comportamiento de especiación en un algoritmo genético. Introdujo un mecanismo que denominó preselección, en el que un descendiente reemplaza al padre de más bajo fitness, siempre y cuando el fitness del hijo sea superior al del padre. De esta manera se mantiene la diversidad pues los strings tienden a reemplazar strings similares a ellos –uno de sus padres–.

De Jong en 1975 generalizó la técnica de preselección de Cavicchio en un esquema llamado *crowding*, donde los nuevos individuos reemplazan los strings existentes de acuerdo a sus similitudes. En realidad, para este esquema se escoge un *crowding factor*, *CF*. Cuando nace una nueva criatura, se seleccionan *CF* individuos de la población –generalmente los de peor valor de fitness–, y se reemplaza al más parecido a la nueva criatura, de esta manera se favorece la creación de especies. De Jong ha obtenido buenos resultados con este esquema sobre funciones multimodales utilizando $CF=2$ y $CF=3$. Resulta especialmente útil en problemas con muchas soluciones; puesto que idealmente, cada especie se dedicará a cada máximo.

Otra manera de conseguir la especiación es restringiendo la selección de los padres a aquellos cromosomas que sean similares, lo que se realiza por periodos que se intercalan con otros en los cuales la cruce es libre.

Codificación

Una de las tareas más importantes que tendrá el diseñador del algoritmo genético será la elección de una adecuada codificación genética que represente a las potenciales soluciones. Seguramente se enfrentará a problemas de digitalización y precisión. Los valores de aquellos parámetros que representan la solución al problema deben discretizarse –si es que no lo están– para poder codificarlos en los cromosomas. Debe asignárseles la cantidad necesaria de símbolos –espacio dentro del cromosoma– para asegurar la precisión requerida.

La codificación también debe contemplar el tratamiento de genotipos no válidos. Por ejemplo si un parámetro de la solución puede tomar valores entre 0 y 9 son necesarios al menos 4 bits para poder especificar todo su rango. ¿Pero qué hacer con las 6 representaciones sobrantes? Una manera de resolverlo –adoptada en este trabajo en todos los casos de neuroevolución implementados– es impedir la existencia de genotipos no válidos, es decir a todo cromosoma posible le corresponde algún fenotipo válido. Pudiendo ocurrir que más de un genotipo

produzca el mismo fenotipo¹. Por ejemplo, una secuencia de 4 dígitos binarios puede decodificarse como el resto de dividir por 10 al valor que representa en el sistema binario. En este caso hay que tener en cuenta que se favorecen ciertos valores respecto de otros puesto que algunos poseen dos representaciones en el cromosoma y otros sólo una². Otras maneras de tratar los genotipos no válidos es asignarles un valor cero como fitness, reemplazar los valores no válidos del cromosoma con algunos predefinidos, o simplemente borrarlo, es decir tomarlo como un acto fallido de reproducción.

La representación tradicional usada para codificar un conjunto de soluciones es el esquema binario. Esto se debe principalmente a la influencia de Holland y su desarrollo teórico que justifica la utilización de un alfabeto de baja cardinalidad. Aunque hoy día cuestionado por las críticas surgidas al teorema de esquemas (Ver en pág. 63) Holland argumentó que la codificación binaria posee un grado más elevado de “paralelismo implícito” porque permite manejar un número mayor de esquemas. Dicho de otra manera, el alfabeto binario es el que ofrece mayor número de esquemas posibles por bit de información si se compara con cualquier otra codificación posible.

El hecho de contar con más esquemas favorece la diversidad e incrementa la probabilidad de que se formen buenos “bloques constructores” en cada generación, lo que en consecuencia mejora el desempeño del Algoritmo Genético con el paso del tiempo de acuerdo al teorema de esquemas.

Por lo tanto, de acuerdo con Holland, es preferible tener muchos genes que puedan tomar unos pocos valores –pocos alelos–, que el caso contrario, es decir pocos genes pero muchos alelos por gen. La idea de un alfabeto de baja cardinalidad también tiene una justificación biológica, ya que en los organismos vivos lo usual es encontrar cromosomas con muchos genes y pocos alelos para cada gen.

Sin embargo los recientes cuestionamientos al teorema de esquemas y la evidencia empírica podrían inclinar la tendencia hacia otras representaciones, por lo menos en algunos casos concretos. El uso de un alfabeto binario puede ser desventajoso en problemas de alta dimensionalidad si se pretende trabajar con una buena precisión, lo que ocasiona cromosomas extremadamente largos y difícilmente llegue a producir resultados aceptables, a menos que se usen procedimientos y operadores especiales diseñados para el problema en cuestión –pérdida de robustez–.

Las principales razones por las que una codificación binaria puede no resultar adecuada son:

- Epistasis³: Utilizando sólo dos dígitos para la codificación es casi imposible evitar que el valor de un bit puede influir en las contribuciones de fitness de otros bits en el genotipo. Si se utilizan alfabetos de mayor cardinalidad pueden conseguirse genes más independientes uno de otros.

¹ Que más de un genotipo den como resultado un mismo fenotipo no es raro en la Naturaleza. La primera ley de Mendel presenta una justificación a ello (Ver “Las leyes de Mendel” –pág. 20–).

² En este ejemplo los valores entre 0 y 5 tendrán dos codificaciones, así el número 3 se escribe en el cromosoma como “0011” ($3 \bmod 10 = 3$) pero también “1101” ($13 \bmod 10 = 3$). Sin embargo los números 6, 7, 8 y 9 poseen una única forma de codificarse. Claramente esto provoca una diferencia probabilística de conseguir ciertos valores al azar.

³ Este concepto ya se definió en este capítulo (ver “Paralelismo implícito y la Hipótesis de construcción de bloques” –pág. 62–) y en el anterior desde el punto de vista biológico (ver “Interacciones entre genes no alélicos” –pág.41–)

- Representación natural: algunos problemas se prestan de manera natural para la utilización de representaciones de mayor cardinalidad que la binaria –por ejemplo, el problema del viajante parece resolverse adecuadamente utilizando permutaciones de enteros–.
- Soluciones ilegales: Como se exponía al comienzo de este apartado, la codificación debe abordar el problema de los genotipos inválidos. Los operadores genéticos utilizados pueden producir con frecuencia soluciones ilegales si se usa una representación binaria.

Códigos de Gray

Un problema que fue notado desde los inicios de la investigación en Algoritmos Genéticos fue que el uso de la representación binaria no mapea adecuadamente el espacio de búsqueda con el espacio de representación. Por ejemplo, si se codifican en binario los enteros 5 y 6, los cuales están adyacentes en el espacio de búsqueda, sus equivalentes en binario serán el 101 y el 110, los cuales difieren en 2 bits en el espacio de representación .

A este fenómeno se le conoce como el riesgo de Hamming y ha conducido a los investigadores a proponer una representación alternativa en la que la propiedad de adyacencia existente en el espacio de búsqueda pueda preservarse en el espacio de representación. La codificación de Gray es parte de una familia de representaciones que caen dentro de esa categoría.

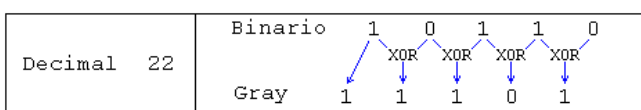
Los códigos de Gray poseen la propiedad de que números consecutivos varían a lo sumo en un bit. Es un código no ponderado, es decir que las posiciones de los bits no tienen un peso específico asignado. Por lo tanto no es apropiado para operaciones aritméticas.¹

Decimal	Código Gray	Decimal	Código Gray	Decimal	Código Gray	Decimal	Código Gray
0	0000	4	0110	8	1100	12	1010
1	0001	5	0111	9	1101	13	1011
2	0011	6	0101	10	1111	14	1001
3	0010	7	0100	11	1110	15	1000

Algunos investigadores han demostrado empíricamente que el uso de códigos de Gray mejora el desempeño del Algoritmo Genético. De hecho Mathias y Whitley encontraron que la codificación de Gray no sólo elimina los riesgos de Hamming, si no que también altera el número de óptimos locales en el espacio de búsqueda así como el tamaño de las buenas regiones de búsqueda. En su trabajo, Mathias y Whitley mostraron empíricamente que un mutador aleatorio de tipo “escalando la colina” es capaz de encontrar el óptimo global de la mayor parte de las funciones de prueba utilizadas cuando se emplea la codificación de Gray, a pesar de que algunas de ellas fueron diseñadas explícitamente para representar dificultades a los algoritmos de búsqueda tradicionales.

Otras codificaciones, además de las binarias han sido propuestas: números reales, vectores, estructuras de árboles, grafos etc. En la actualidad se distinguen dos escuelas, una que propone limitarse a strings binarios y la otra que utiliza estructuras de todo tipo. En el

¹ Cualquier número binario puede convertirse a su representación en código Gray como sigue: El primer bit del código Gray es el mismo como el primer bit del número binario. El segundo bit del código Gray es igual a la operación XOR del primer y segundo bits del número binario; El tercer bit del código Gray es igual al XOR del segundo y tercer bits del número binario y así sucesivamente. Ejemplo:



segundo caso se suele hablar de *Programas de evolución*¹ en lugar de *Algoritmos genéticos* y en la que deben redefinirse adecuadamente los operadores genéticos, normalmente crossover y mutación de acuerdo con la estructura elegida.

Codificando números reales

Los códigos de Gray pueden ser muy útiles para codificar enteros por las razones ya expuestas, no obstante si de números reales se trata presentan los mismos inconvenientes que la codificación binaria corriente. El problema más importante es la alta dimensionalidad. Si se tienen demasiadas variables reales, y se pretende una buena precisión para cada una de ellas, entonces las cadenas binarias necesarias son extremadamente largas, y el Algoritmo Genético tenderá a un desempeño pobre.

La utilización del formato binario estándar para representar números reales codificando mantisa y exponente no es recomendable puesto que un pequeño cambio en el exponente –por ejemplo una mutación en un solo bit– producirá grandes saltos en el espacio de búsqueda, mientras que perturbaciones en la mantisa pueden no cambiar de manera significativa el valor numérico codificado.

Mientras que Holland y otros han utilizado la codificación binaria por ser supuestamente superior a otros tipos de codificaciones, existen pruebas empíricas que para una cantidad significativa de aplicaciones del mundo real el uso directo de números reales en un cromosoma funciona mejor que la representación binaria tradicional.

El uso de números reales en una cadena cromosómica ha sido común en otras técnicas de Computación Evolutiva. Sin embargo, los teóricos de los Algoritmos Genéticos han criticado fuertemente el uso de valores reales en los genes de un cromosoma, argumentando que produce un comportamiento más errático y difícil de predecir sobre el algoritmo. Debido a esto, se han diseñado varios operadores especiales en los años recientes, para permitir que una codificación basada en números reales siga gozando de las mismas propiedades y características estudiadas para el caso binario que cuenta con mucha teoría desarrollada.

Se han utilizado también codificaciones alternativas para las representaciones de números reales en los cromosomas. Por ejemplo, el uso de enteros para representar cada dígito decimal del valor real se ha aplicado exitosamente a varios problemas de optimización. La precisión está limitada por la longitud de la cadena, y puede incrementarse o decrementarse según se desee. Los operadores de cruce tradicionales –un punto, dos puntos y uniforme– pueden usarse directamente en esta representación, y la mutación puede consistir en generar un dígito aleatorio para una cierta posición o bien en producir una cierta perturbación para evitar saltos extremadamente grandes en el espacio de búsqueda. Esta representación pretende ser un compromiso entre Algoritmos Genéticos con codificación real y una representación binaria de números reales, manteniendo lo mejor de ambos esquemas al incrementar la cardinalidad del alfabeto utilizado, pero manteniendo el uso de los operadores genéticos tradicionales casi sin cambios.

Representaciones de longitud variable

En algunos problemas el uso de alfabetos de alta cardinalidad puede no ser suficiente, pues además puede requerirse el empleo de cromosomas de longitud variable para lidiar con cambios que ocurran en el ambiente con respecto al tiempo. Por ejemplo, en el diseño de

¹ Más adelante se expondrán brevemente particularidades de ésta y otras metodologías relacionadas en el apartado “Otros Paradigmas de la Computación Evolutiva” –pág. 78–.

redes neuronales, si bien la cantidad de neuronas de entrada y salida suelen estar fijas por la naturaleza del problema que resuelven, no ocurre lo mismo con la cantidad de neuronas ocultas. El conocimiento exacto de cuántas neuronas ocultas se necesitan generalmente está ausente. Si se usa un locus para cada neurona de la capa oculta, un cromosoma de longitud variable permitirá también encontrar el número adecuado de ellas.

Algunas veces, puede ser posible introducir símbolos en el alfabeto que sean considerados como posiciones "vacías" a lo largo de la cadena, con lo que se permite la definición de cadenas de longitud variable aunque los cromosomas tengan una longitud fija.

En caso de verdaderos cromosomas de longitud variable se necesitarán operadores especiales como: *añadir* y *eliminar*. Estos operadores se utilizan para añadir o eliminar un gen del cromosoma. La forma más habitual de añadir un locus es duplicar uno ya existente, el cual sufre mutación y se añade al lado del anterior.

Goldberg propuso el uso de un tipo especial de Algoritmo Genético de longitud variable el cual usa también poblaciones de tamaño variable. A este algoritmo se lo denominó desordenado –messy GA o mGA– en contraposición con el algoritmo estándar –u ordenado–, que tiene longitud y tamaño de población fijos.

La idea básica de los Algoritmos Genéticos desordenados es empezar con cromosomas cortos, identificar un conjunto de buenos bloques constructores y después incrementar la longitud del cromosoma para propagar estos buenos bloques constructores a lo largo del resto del string.

Poseen dos fases, una primordial, en la que se evolucionan bloques constructores cortos un número fijo de generaciones que provoca el incremento del tamaño de la población. Terminada esta fase primordial se pasa a la fase yuxtaposicional. A partir de este punto, el tamaño de la población permanecerá fijo, y se usan dos operadores especiales llamados "corte" y "unión". El operador de corte simplemente remueve una porción del cromosoma, mientras que el de la unión junta dos segmentos cromosómicos. Debido a la naturaleza del Algoritmo Genético desordenado, los strings producidos por los operadores de corte y unión siempre serán válidos. Si los bloques constructores producidos en la fase primordial acarrean suficiente información, entonces el Algoritmo desordenado será capaz de arribar al óptimo global aunque el problema sea de los AG-difíciles (Ver “Desenvolvimiento de los Algoritmos Genéticos” –pág. 75–).

Representación de árbol

Utilizada por John Koza para desarrollar la denominada “programación genética” –ver pág. 80–. En lugar de utilizar un arreglo unidimensional de datos –cromosoma clásico– utiliza una estructura de árbol. Deben redefinirse adecuadamente los operadores genéticos.

El crossover puede efectuarse numerando los nodos de los árboles correspondientes a los 2 padres elegidos y seleccionando –al azar– un punto en cada uno de ellos de manera que los subárboles por debajo de dicho punto se intercambien.

Generalmente los tamaños de los 2 árboles padres pueden ser diferentes. Si las raíces de ambos padres son seleccionadas como puntos de cruce, habitualmente no se efectúa la cruce, y los hijos pasan a ser idénticos a sus padres.

La mutación se efectúa mediante la selección –aleatoria– de un cierto punto de un árbol. El subárbol que se encuentre por debajo de dicho punto es reemplazado por otro árbol generado al azar .

La permutación es un operador asexual, una especie de mutación, que emula el efecto del operador de inversión en los cromosomas lineales. Este operador reordena las hojas de un

subárbol ubicado a partir de un punto elegido al azar, y su finalidad es fortalecer la unión de combinaciones de alelos con buen desempeño dentro de un cromosoma.

Codificación Estructurada

Es un compromiso entre el cromosoma lineal clásico de longitud fija y la codificación de árbol. Los algoritmos que utilizan esta codificación se llaman Algoritmos Genéticos Estructurados y sus características son las siguientes:

Los cromosoma se representan por un arreglo lineal de longitud fija, pero en él se evidencian jerarquías codificando una estructura de varios niveles –grafo dirigido o árbol–.

Existen genes que actúan como operadores de cambio encendiendo o apagando otros genes.

Los genes de alto nivel activan o desactivan conjuntos de genes de más bajo nivel, lo que significa que cualquier cambio pequeño a un alto nivel se magnifica en los niveles inferiores.

La estructura de datos que se requiere para implementar estos algoritmos es ligeramente más complicada que el arreglo unidimensional que requiere un Algoritmo Genético tradicional.

Las representaciones anteriores no son las únicas alternativas que existen en la literatura especializada. Algunas son más dependientes del problema, tales como la matricial, la multi-dimensional y la de permutaciones que han sido propuestas también por algunos investigadores.

Dominancia y Número Diploide

En el capítulo anterior se presentaron las bases teóricas de la herencia desde el punto de vista de la Genética y la Biología. Allí se puede encontrar una descripción del mecanismo de propagación de los genes individuales a la descendencia y de cómo cada rasgo fenotípico en los organismos superiores está regulado por un par de alelos en lugar de uno sólo. El apartado titulado “Las leyes de Mendel” –pág. 20– describe aceptablemente este mecanismo y se recomienda su lectura previa.

Pero ¿Cuál puede ser el beneficio de esta redundancia en la codificación genética? Según Goldberg la teoría que más sentido tiene dentro del contexto de los Algoritmos Genéticos es que un número diploide de cromosomas provee un mecanismo de memoria para los alelos que en algún momento fueron útiles en un ambiente que al cambiar los transformó en menos aptos que otros. Así los organismos diploides pueden responder rápidamente a un cambio brusco del entorno pues la redundancia en la codificación permite múltiples soluciones a un mismo problema, y aunque sólo una es expresada, la alternativa está presente con carácter recesivo. Los alelos no dominantes tienen chance de expresarse en el caso de formarse un organismo homocigota –el mismo alelo en los dos genes del par homólogo–. Pero además los biólogos han demostrado que la dominancia evoluciona, es decir que se encuentra bajo control genético y por ende un cambio adaptativo puede hacer que un alelo recesivo se transforme en dominante –o viceversa–.

En el ejemplo citado en el capítulo anterior de las mariposas del barrio industrial de Inglaterra que debieron cambiar rápidamente el color de sus alas de blanco a negro para mimetizarse con la corteza de los abedules oscurecidas por el hollín puede realizarse la siguiente interpretación: “las mariposas no afrontaron la tarea de *evolucionar* el color negro a partir del blanco, sino que una adaptación mucho más simple fue suficiente: el alelo para el color negro, antes recesivo, evolucionó a dominante”. Esto quiere decir que en algún momento de su evolución estos insectos poseían alas de color negro, rasgo que se transformó en recesivo.

Pero en el momento de necesitarlo, un cambio rápido de dominancia permitió la sobrevivencia de esta especie a un cambio brusco del entorno.

Algunos de los primeros ensayos experimentales de Algoritmos Genéticos aplicaron el concepto de genotipos diploides y dominancia. Tal es el caso de Bagley que en 1967 experimentó utilizando dominancia variable codificada dentro del cromosoma. Desafortunadamente encontró que los valores de dominancia tendían a quedar fijos bastante temprano en la evolución. Utilizó también un esquema de dominancia aleatoria pero no llegó a resultados concluyentes.

Los estudios de optimización de funciones realizados por Hollstien en 1971 incluyeron un mecanismo de evolución de dominancia. Utilizó genotipo diploide con codificación binaria, y para permitir que la dominancia pudiese cambiar con la evolución desarrolló un esquema algo complejo donde cada gen –en cada locus homólogo– se codificaba por dos genes, uno funcional y otro modificador. Pero el mismo efecto lo consiguió luego de manera más simple introduciendo un esquema trialélico con alfabeto de codificación {0,1,2}. En este esquema el 2 juega el rol de 1 dominante, mientras que el 1 representa al 1 recesivo.

Cromosoma 1	0 0 0 1 1 1 2 2 2
Cromosoma 2	0 1 2 0 1 2 0 1 2
Dominancia	0 0 1 0 1 1 1 1 1

Este ha sido el esquema más claro y simple propuesto para representar la dominancia en los Algoritmos Genéticos. Es austero en cuanto al espacio requerido, a penas medio bit más por locus y la dominancia puede cambiar fácilmente mutando un 2 por un 1 o viceversa.

Hollstien consiguió mayor grado de diversidad genética pero no una mejora significativa en el desempeño comparado con un esquema haploide. Según Goldberg, esto se debe a que los test fueron realizados sobre ambientes estacionarios, mientras que la ventaja de evolucionar genotipos diploides se manifiesta sobre ambientes que cambian con el tiempo.

Brindle en 1981 experimentó con varios esquemas de dominancia en problemas de optimización de funciones. Desafortunadamente los test y códigos utilizados han sido cuestionados y por lo tanto sus resultados.

Goldberg y Smith en 1987 experimentaron con el esquema trialélico de Hollstein sobre un problema de entorno no estacionario obteniendo mejoras significativas respecto a una solución haploide e incluso a otros esquemas de dominancia testeados.

Frecuencia de los operadores

En la ejecución de todo algoritmo genético es necesario especificar la frecuencia con que se aplicarán los operadores genéticos. Como se explicara en su momento –ver “Crossover y Mutación” pág. 49– el operador de mutación se aplica con baja frecuencia, en contraste con el crossover que se aplica frecuentemente. En general, aquellos operadores que generan diversidad de manera aleatoria se aplican con baja frecuencia para no llevar al algoritmo a una búsqueda aleatoria.

La frecuencia de aplicación de cada operador generalmente no cambia durante el algoritmo, sin embargo, se puede diseñar un algoritmo que varíe dinámicamente la probabilidad de aplicar cada operador en función de la performance de la evolución. Hay que tener en cuenta que en las primeras generaciones tanto el crossover como la mutación son operadores eficaces, pero conforme la población va convergiendo el crossover pierde utilidad al trabajar con individuos muy similares. En esta etapa la mutación puede llevar la exploración hacia otras áreas en el espacio de búsqueda, pero incrementar demasiado la frecuencia de mutación

tampoco es buena idea pues transformará el algoritmo en una búsqueda aleatoria. Otra opción es aplicar operadores especializados.

Extensiones y Variantes

Elitismo

El método más utilizado para mejorar la convergencia de los Algoritmos Genéticos es el elitismo. Se utiliza en problemas donde la evolución se comporta de manera errática con excesivo grado de diversidad. Es el opuesto al caso de crowding que se vio anteriormente (“Diversidad Genética y Convergencia” –pág. 64–).

Consiste básicamente en seleccionar una elite de individuos de entre los mejores de la población e incorporarlos directamente a la siguiente generación, sin pasar por el proceso de reproducción.

Comúnmente, el tamaño de la elite es bastante pequeño –de 1% a 5% de la población–, y el tipo de muestreo es directo –los mejores ranqueados– o bien por sorteo proporcional a su fitness. No obstante existen variantes, por ejemplo SANE –un método de neuroevolución que se verá más adelante y al cual ya se ha hecho referencia– utiliza marcadamente el elitismo pasando intactos a la próxima generación el 50% de los individuos de la población, claro está que dicha presión hacia la convergencia se ve compensada por la aplicación de un doble mecanismo de mutación con alta frecuencia de acción. –ver en pág. 210–.

El esquema básico de un algoritmo genérico elitista es el siguiente :

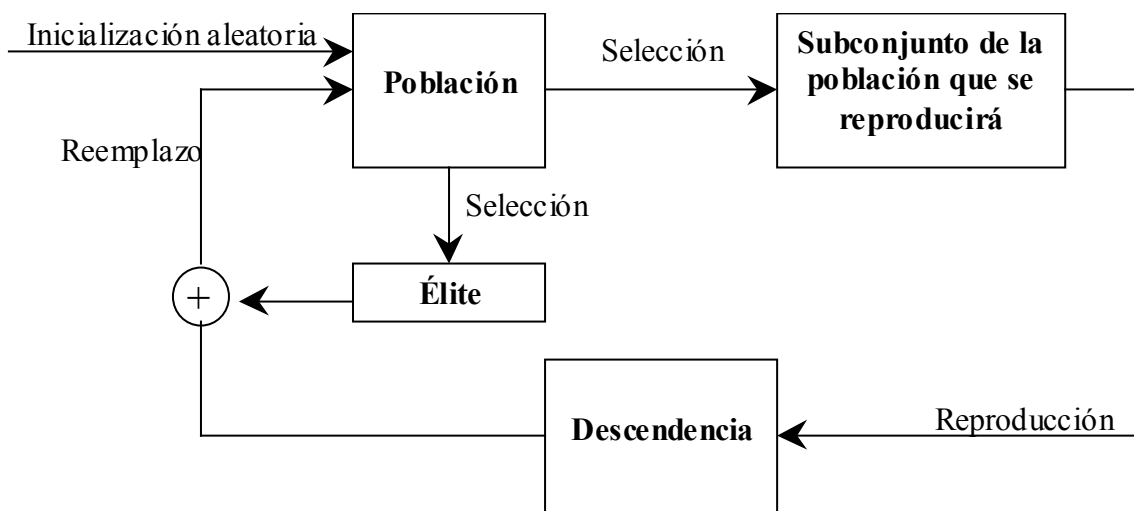


Ilustración 21 Esquema de Algoritmo Genético Elitista

El elitismo en la práctica, mejora la velocidad de convergencia de los Algoritmos Genéticos cuando la función de evaluación es unimodal –no hay subóptimos–, sin embargo la velocidad de convergencia empeora con funciones fuertemente multimodales.

En poblaciones de tamaño reducido puede utilizarse la reinicialización, obteniéndose resultados similares al elitismo. Consiste en esperar a que el algoritmo converja, mantener los mejores individuos evolucionados, reiniciar aleatoriamente todos los demás, y proseguir con la evolución. Esta técnica tiene efectos beneficiosos debido a que introduce diversidad genética, que como se vio anteriormente, resulta crítico en los Algoritmos Genéticos, sobre todo con poblaciones pequeñas.

Aislamiento geográfico

Es otra variante que procura mantener diversidad, esta vez, basada en el hecho que poblaciones aisladas geográficamente evolucionan independientemente de las demás favoreciendo la aparición de distintas características.

Para emular esta situación, en el contexto de los Algoritmos Genéticos, se han desarrollado varios métodos denominados modelos de islas, donde las poblaciones evolucionan independientemente permitiendo migraciones periódicas entre ellas.

La idea básica consiste en dividir la población total en varias subpoblaciones en cada una de las cuales se lleva a cabo un Algoritmo Genético. Cada cierto número de generaciones, se efectúa un intercambio de información entre las subpoblaciones, proceso que se denomina migración. La introducción de la migración hace que los modelos de islas sean capaces de explotar las diferencias entre las diversas subpoblaciones, obteniéndose de esta manera una fuente de diversidad genética. Cada subpoblación es una “isla”, definiéndose un procedimiento por medio del cual se mueve el material genético de una “isla” a otra. La determinación de la tasa de migración, es un asunto de capital importancia, ya que de ella puede depender la convergencia prematura de la búsqueda.

Se pueden distinguir diferentes modelos de islas en función de la comunicación entre las subpoblaciones. Algunas comunicaciones típicas son las siguientes:

Comunicación en estrella, en la cual existe una subpoblación que es seleccionada como maestra –aquella que tiene mejor media en el valor de la función objetivo–, siendo las demás consideradas como esclavas. Todas las subpoblaciones esclavas mandan sus mejores individuos a la subpoblación maestra la cual a su vez manda sus mejores individuos a cada una de las subpoblaciones esclavas.

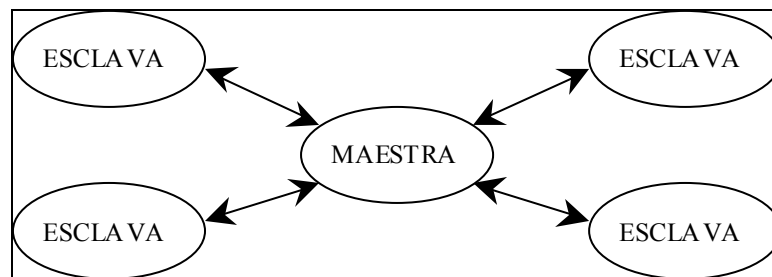


Ilustración 22 Comunicación en estrella

Comunicación en red, en la cual no existe una jerarquía entre las subpoblaciones, mandando todas y cada una de ellas sus mejores individuos al resto de las subpoblaciones.

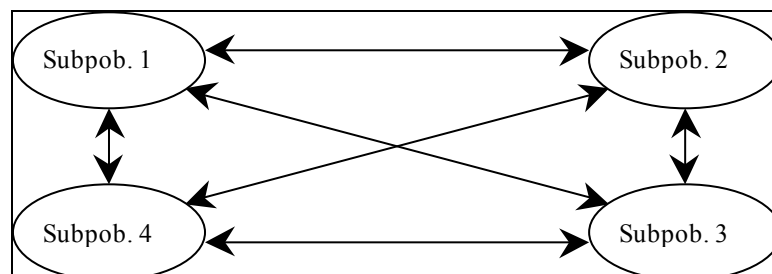


Ilustración 23 Comunicación en red

Comunicación en anillo, en la cual cada subpoblación envía sus mejores individuos, a una población vecina, efectuándose la migración en un único sentido de flujo.

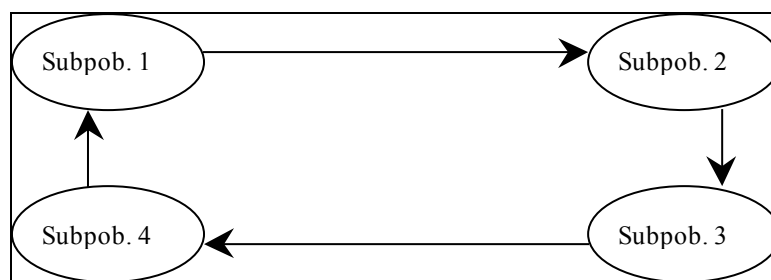


Ilustración 24 Comunicación en anillo

Evaluación del Comportamiento de los Algoritmos Genéticos

Explotación vs. Exploración

Cualquier algoritmo de optimización debe usar dos técnicas para encontrar el máximo global: exploración para investigar nuevas y desconocidas áreas en el espacio de búsqueda y explotación utilizando el conocimiento de los puntos visitados previamente para ayudar a encontrar mejores puntos en esa región del espacio de búsqueda. Estas dos técnicas son contradictorias, la primera está asociado a una búsqueda a lo ancho investigando nuevas zonas, y la segunda está asociada a una búsqueda en profundidad sobre la región que se está investigando. Un buen algoritmo de búsqueda debe encontrar un buen compromiso entre ambas técnicas.

Una búsqueda puramente aleatoria es buena para la exploración pero no para la explotación puesto que no aprovecha el conocimiento sobre los puntos ya visitados, mientras que una búsqueda del tipo hillclimbing es buena para la explotación pero realiza poca exploración. La combinación de ambas estrategias puede ser bastante más efectiva que cualquiera de ellas por separado, pero es difícil saber donde está el mejor balance, es decir cuánto de exploración se debe realizar antes de cambiar a una técnica explotatoria en la región más prometedor, o a la inversa, comenzar con una técnica de explotación y si no se arriba a una buena solución, buscar otra región con una exploración para volver a explotar una región distinta.

Holland demostró que los Algoritmos Genéticos combinan ambas técnicas al mismo tiempo de manera óptima, no obstante, tal demostración, por simplificación, está basada en una serie de suposiciones que difícilmente puedan hacerse verdaderas en la práctica. Esta suposiciones incluyen:

1. Que la población sea de tamaño infinito.
2. Que la función de fitness refleje certeramente la utilidad de la solución.
3. Que los genes en un cromosoma no interactúen significativamente –poca epistasis–.

El primer supuesto, obviamente es imposible de satisfacer en la práctica. Se presenta el problema de determinar el tamaño de la población. Una población demasiado grande consumirá innecesariamente recursos de cómputos así como tiempo de procesamiento. Una

población más chica de lo necesario puede ser afectada por la “deriva genética”¹ que establece que aún en ausencia de cualquier presión selectiva –por ejemplo con una función de fitness que asigne el mismo valor a todos los individuos– los miembros de la población pueden converger a algún punto en el espacio de soluciones. Esto ocurre por la acumulación de errores estocásticos que por hechos fortuitos comienzan a predominar en la población. La deriva genética atenta contra la convergencia hacia la solución correcta. Por ejemplo si el algoritmo utiliza el gradiente como información en el fitness, esta función debe ser lo suficientemente importante como para contrarrestar cualquier deriva genética. En poblaciones suficientemente grandes –o infinita– no existe este inconveniente.

Las suposiciones 2 y 3 pueden satisfacerse en ejemplos de laboratorio pero rara vez se cumplen en problemas del mundo real.

Además hoy día se ha puesto en duda la demostración que realizara Holland sobre cómo el algoritmo genético es capaz de aprovechar de manera óptima o cuasi-óptima la información que va obteniendo sobre la distribución de los valores de fitness argumentando que cualquier algoritmo ofrece los mismos resultados medios si se promedia sobre todos los problemas posibles. [CottaPorras98]

Desenvolvimiento de los Algoritmos Genéticos

Una de las características, quizá la más importante, atribuida a los Algoritmos Genéticos es la robustez, es decir, el hecho de que un mismo algoritmo sirva para una gran variedad de problemas sin perder eficiencia. No obstante debe recordarse que las opiniones están divididas al respecto y algunos investigadores no están del todo de acuerdo (ver “Críticas a la Base Teórica de los Algoritmos Genéticos” –pág. 63–).

Donde existan técnicas para resolver problemas específicos, estas generalmente sobrepasarán a los Algoritmos Genéticos en velocidad y precisión. Por lo tanto, una de las áreas de mayor interés son aquellos problemas para los que no se tiene suficiente información como para preparar un algoritmo más específico.

De todas maneras, es de suponer que no todos los problemas son igualmente aptos para ser resueltos por un Algoritmo genético. La hipótesis de los bloques constructivos –ver pág. 62– dice que los Algoritmos Genéticos son viables para resolver los problemas cuya solución pueda formarse con bloques constructivos.

En la práctica se ha observado que, en ocasiones los Algoritmos Genéticos resuelven satisfactoriamente un problema de optimización dado y otras se quedan muy alejados del óptimo. Los estudiosos del tema han tratado de caracterizar lo que han denominado problemas AG-fáciles –aquellos en los que los Algoritmos Genéticos proporcionan buenos resultados– y AG-difíciles con el objetivo de saber de antemano, al estudiar un nuevo problema, si los Algoritmos Genéticos son una buena elección para su resolución.

Se han tratado de caracterizar estas clases mediante el concepto de engaño considerando que si el algoritmo converge al mejor esquema –aquel con mejor promedio de fitness de sus strings– y en éste se encuentra el óptimo, entonces es fácil que se resuelva satisfactoriamente. En caso de que el óptimo esté en un esquema con bajo promedio se denomina engaño y se

¹ Desde el punto de vista biológico puede encontrarse información respecto de la deriva genética en el apartado “Teoría de la genética de poblaciones –neodarwinismo–” –pág. 23–

pensaba que en estos casos es cuando el problema es AG-difícil. Sin embargo se ha visto que esta caracterización mediante el engaño no es siempre cierta y no constituye un criterio fiable.

En el algoritmo clásico de Holland si un esquema con buena performance tiene bits muy alejados –en la representación binaria con la que trabaja el algoritmo– la probabilidad de que el crossover los elimine es alta –recordar concepto de ligamiento visto en el capítulo anterior en la pág. 37–. Si los bits estuviesen juntos la probabilidad sería baja. El algoritmo va a tener mejor performance para aquellos problemas en donde la distancia entre bits dependa de la correlación de sus valores. Si esto no se cumple la performance del Algoritmo Genético será pobre.

Este factor se debe al operador de cruza utilizado, no al método, y cambiándolo por uno independiente de la posición de los bits, como es la "cruza uniforme", se obtiene otro Algoritmo Genético sin pérdida de performance para esos problemas.

Una técnica interesante para avanzar en este sentido es la utilización de operadores adaptativos. En estos hay distintos parámetros que se van ajustando en base al progreso que van produciendo en la población. Algunos candidatos para ser estos parámetros son: las probabilidades de los operadores de crossover y mutación, los puntos de cruce en el crossover de n -puntos y el tipo de operador utilizado.

Sin llegar a ser adaptativos los parámetros pueden ser dinámicos, es decir, cambiantes pero sin basarse en los resultados parciales. Por ejemplo, se puede comprobar que el efecto del crossover va disminuyendo a medida que la población converge, por lo tanto incrementar la probabilidad de utilizarlo a medida que avanza la evolución podría llegar a ser de utilidad.

Ventajas de los Algoritmos Genéticos respecto de otros métodos

A continuación se detallan algunas ventajas de los Algoritmos Genéticos respecto de otros métodos de búsqueda tradicionales, la mayoría expuestas por Goldberg:

- Cuando se usan para problemas de optimización resultan menos afectados por los máximos locales –falsas soluciones– que las técnicas tradicionales.
- Son fácilmente implementables en computadoras con capacidades medias, proporcionando resultados aceptables, en cuanto a precisión y recursos empleados, para una gran cantidad de problemas difícilmente resolubles por otros métodos.
- Existe una gran cantidad de ensayos empíricos que proporcionan operadores, parámetros e implementaciones específicas para una amplia gama de problemas.
- Los Algoritmos Genéticos manejan variables de decisión o de control representadas como cadenas con el fin de explotar similitudes entre cadenas de altas prestaciones. Otros métodos tratan habitualmente con las funciones y sus variables de control directamente. Dado que los Algoritmos Genéticos operan en el nivel de códigos, son difíciles de engañar aun cuando la función sea difícil para los enfoques tradicionales.
- Los Algoritmos Genéticos trabajan con una población; muchos otros métodos trabajan con un único punto. De este modo, los Algoritmos Genéticos encuentran seguridad en la cantidad. Al mantener una población de puntos bien adaptados se reduce la probabilidad de alcanzar un falso óptimo. Dicho de otra forma, operan simultáneamente con varias soluciones a la vez, en lugar de trabajar en forma secuencial.
- Los Algoritmos Genéticos consiguen gran parte de su amplitud ignorando la información que no sea la del objetivo. Otros métodos se basan fuertemente en tal información, y en

problemas donde la información no está disponible o es difícil de conseguir, estos otros métodos fallan.

- Los Algoritmos Genéticos son generales porque explotan la información disponible en cualquier problema de búsqueda, procesando similitudes en el código subyacente junto con información proveniente de la ordenación de las estructuras de acuerdo con sus capacidades de supervivencia en el entorno actual. Al explotar una información tan fácilmente disponible, los Algoritmos Genéticos se pueden aplicar en prácticamente cualquier problema.
- Las reglas de transición de los Algoritmos Genéticos son estocásticas; otros muchos métodos tienen reglas de transición deterministas. Hay una diferencia, no obstante, entre los operadores estocásticos de los Algoritmos Genéticos y otros métodos que no son más que paseos aleatorios. Los Algoritmos Genéticos usan el azar para guiar una búsqueda fuertemente explotadora. Esto puede parecer inusual, usar el azar para conseguir resultados concretos –los mejores puntos–, pero hay gran cantidad de precedentes en la naturaleza.

Aplicación de los Algoritmos Genéticos

Los Algoritmos Genéticos se han aplicado con éxito en numerosas áreas. Algunas se mencionan a continuación:

- *Identificación de sistemas:* Se refiere a la obtención de la estructura y los parámetros de un modelo matemático generalmente dinámico. Por ejemplo la neuroevolución donde se consiguen tanto la estructura de las Redes Neuronales como los pesos de las conexiones –parámetros–.
- *Control:* En el diseño de los controladores normalmente se requiere optimizar algunos parámetros, la configuración o el orden para reducir la complejidad del sistema. Tales tareas pueden llevarse a cabo por medio de un Algoritmo Genético.
- *Robótica:* Se han utilizado en sistemas de navegación controlados por robot para optimizar la trayectoria desde el arranque hasta el destino deseado y evitar las colisiones.
- *Diseños de ingeniería:* Se han utilizado para optimizar formas y disposición de elementos en circuitos de alta escala de integración.
- *Planeamiento y ordenamiento:* Se refiere a optimizar acciones, movimientos y tareas que deben realizarse cumpliendo determinadas restricciones. Son ejemplos el problema del viajante de comercio, la distribución de bombas de agua, organización del trabajo y el mantenimiento, así como los sistemas de producción-inventario-distribución.
- *Sistemas clasificadores:* Los sistemas clasificadores, son arquitecturas que aprenden reglas sintácticamente simples que, en su conjunto, resuelven un problema determinado. Un sistema clasificador contiene, como una de sus partes, a un Algoritmo Genético que evoluciona un conjunto de reglas –que son los individuos de la población– que, trabajando conjuntamente, resuelven un problema determinado.

También los Algoritmos Genéticos se han aplicado a reconocimiento de patrones y del habla, estrategias de juegos, vida artificial y análisis de modelos políticos y económicos.

Otros Paradigmas de la Computación Evolutiva

Hoy día, términos como *Algoritmos Genéticos*, *Estrategias Evolutivas*, *Programación Genética*, *Programación Evolutiva* y otros más, se utilizan a veces de manera algo confusa. Hasta hace poco tiempo se usaba el término *Algoritmos Genéticos* para designarlos a todos ellos, pero poco a poco se han empezado a reconocer algunas divisiones aunque aún no del todo precisas.

Los Algoritmos Genéticos y su rol en la Computación Evolutiva

La Computación Evolutiva es un compendio de paradigmas donde los Algoritmos Genéticos ocupan un papel importante debido a que:

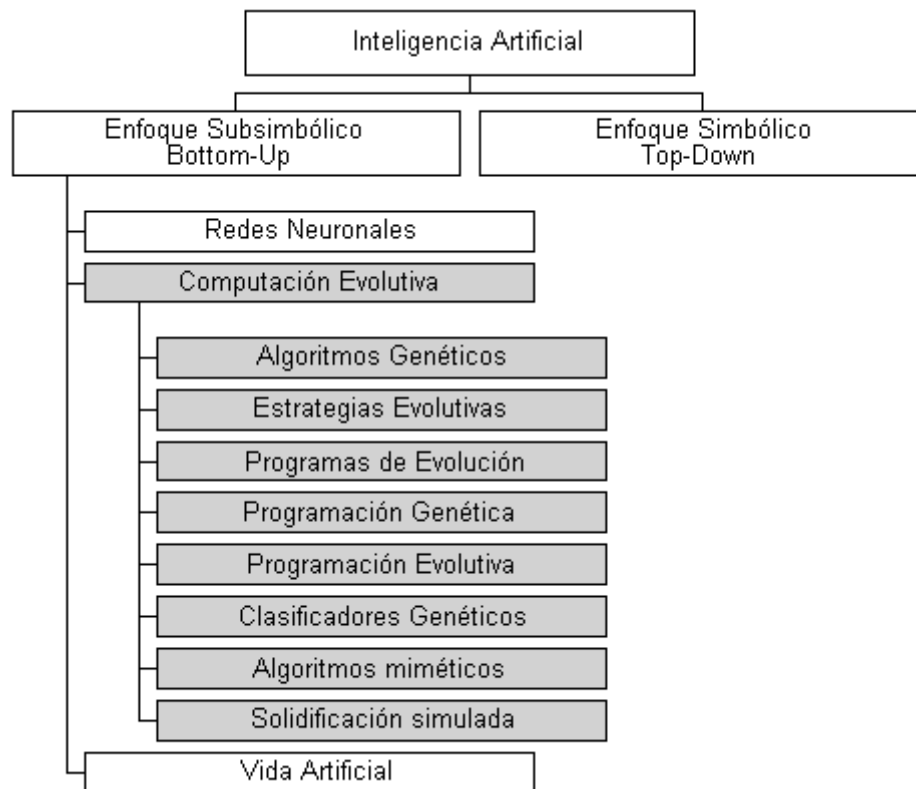
- Es una de las técnicas más usadas actualmente –si no la más– comparándola con cualquier otra familias de algoritmos de la Computación Evolutiva.
- Constituyen el paradigma más completo de la Computación Evolutiva, resumiendo de modo natural, todas las ideas fundamentales de esta rama de la Inteligencia Artificial Subsimbólica.
- Son generales y flexibles pudiendo adoptar con facilidad nuevas ideas, generales o específicas, que surjan dentro del campo de la Computación Evolutiva. Además se pueden hibridar fácilmente con otros paradigmas y enfoques, aunque no tengan ninguna relación con la computación evolutiva.
- Es el paradigma con mayor base teórica de entre los de la Computación Evolutiva. Además, dicha base teórica es sencilla en su desarrollo y con grandes posibilidades de ampliación.
- De entre todos los paradigmas de la Computación Evolutiva es el que menos conocimiento específico necesitan para su funcionamiento, y en consecuencia, los más versátiles. Pero es que además pueden incorporar conocimiento específico con poco esfuerzo adicional.

Clasificación

El enfoque subsimbólico de la Inteligencia Artificial se caracteriza por crear sistemas con capacidad de aprendizaje. Éste se puede obtener a nivel de individuo imitando el cerebro – Redes Neuronales–, o a nivel de especie, imitando la evolución –Computación Evolutiva–. En ambos casos se parte de representaciones de bajo nivel –neuronas y cromosomas– que se manipulan para conseguir el objetivo propuesto –de alto nivel de abstracción–, por eso se dice que, a diferencia de la Inteligencia Artificial Simbólica, el enfoque es de abajo hacia arriba (Bottom–Up).

Por razones históricas, para los métodos que incorporan a un Algoritmo Genético conocimientos sobre el dominio se usa frecuentemente la palabra "evolutivos" mientras que el vocablo "genéticos" se reserva para los relacionados con los Algoritmos genéticos robustos, generales, independientes del problema a resolver, salvo, claro está en lo relativo a codificación y función de fitness que deben ser específicos del problema.

Se presenta a continuación una clasificación obtenida de [Gascón]. Según el autor, las nuevas ciencias siempre pasan por una fase inicial de desconcierto e inconsistencia hasta llegar a unas convenciones aceptadas por todos. Vale decir, la siguiente es una de las clasificaciones posible pues aún existen límites borrosos.



El paradigma conocido como Vida artificial es difícil de clasificar. Utiliza tanto técnicas de la Computación Evolutiva y las Redes Neuronales para llevar a cabo sus objetivos. Precisamente éstos son los que lo diferencian de la Computación Evolutiva.

La Vida Artificial pretende respetar al máximo la metáfora con el mundo real. Según este planteamiento, es preferible programar estas simulaciones desde el punto de vista de las entidades, siendo la propia entidad la que "decide" en cada momento cómo actuar. Por el contrario la Computación Evolutiva simula tan sólo los aspectos de la naturaleza que sean útiles para resolver problemas, pero no se pretende mantener una congruencia con el mundo real. Es por ello que los algoritmos de Computación Evolutiva se enfocan, por comodidad, desde el punto de vista del proceso.

Como puede verse en el gráfico, el término Computación Evolutiva agrupa un conjunto de paradigmas muy relacionados cuyas competencias aún no están bien definidas. Partiendo del conocimiento del funcionamiento de los Algoritmos Genéticos –probablemente la técnica más conocida de todas ellas–, el resto de los algoritmos se pueden interpretar como variaciones o mejoras de éstos –excepto Simulated Annealing que se consideraría una simplificación–, añadiendo complejidad o modificando ciertas características del algoritmo.

Esta clasificación se basa sobre todo en detalles de desarrollo histórico más que en el hecho de un funcionamiento realmente diferente, de hecho las bases biológicas en las que se apoyan son esencialmente las mismas. Las diferencias entre ellos se centran principalmente en los operadores que se usan en cada caso y en general en la forma de implementar la selección, reproducción y sustitución de individuos en una población.

A continuación se presenta una brevísima explicación de cada una de estas clasificaciones.

Estrategias Evolutivas

Se centran en el estudio de problemas de optimización paramétricos, y trabajan sobre poblaciones de cromosomas compuestos por números reales. Su nombre viene del alemán “*Evolutionstrategien*”. Como decíamos anteriormente la clasificación se debe más que nada a causas históricas. Las estrategias Evolutivas son la versión alemana de los Algoritmos Genéticos. Aunque los orígenes de ambas disciplinas partieron de enfoques distintos, los sucesivos desarrollos de ambos los han ido acercando hasta el punto de que, actualmente, una estrategia evolutiva se puede considerar como un tipo de algoritmo genético y viceversa.

Un aspecto característico de las Estrategias Evolutivas es que las tasas de mutación de cada gen se codifican como parte del genoma, por lo que también se encuentran sujetas a evolución.

Programas de Evolución

Los *programas de evolución* son refinamiento de los Algoritmos Genéticos, propuesto por Michalewicz en 1994. En principio los Algoritmos Genéticos recomendaron para la codificación un alfabeto minimal –en lo posible binario–. Michalewicz propuso hacer una codificación más próxima al problema. Un alfabeto binario evitaba ciertos problemas que los Programas de Evolución deben resolver explícitamente, como por ejemplo, de qué manera efectuar una mutación o un crossover.

Se debe procurar que:

- La Mutación y crossover sean cerrados en el conjunto de representaciones válidas, es decir el resultado obtenido al aplicar cualquiera de estos operadores debe ser un elemento válido en la representación elegida.
- Todo elemento del espacio de soluciones válido, debe poder alcanzarse por medio de mutación o cruzamiento de otros elementos válidos.

A veces asegurar estas dos características puede ser complicado pero, al utilizar representaciones más cercanas al problema a resolver y de alto nivel, suelen ser mas eficientes y la hibridación con otro tipo de sistema de aprendizaje es mucho más simple.

Por lo general, los Programas de Evolución se han englobado dentro de los Algoritmos Genéticos –como una variante–, y al revés –los Algoritmos Genéticos como un caso particular–. En definitiva, actualmente se utilizan los términos de Algoritmo Genético y Programa de Evolución indistintamente para designar algoritmos evolutivos con cualquier tipo de alfabeto.

Cuando el alfabeto utilizado en un Programa de Evolución no es discreto sino continuo, es decir, cuando los genes son números reales con la máxima precisión –decimales– posible, los Programas de Evolución se confunden con las Estrategias Evolutivas.

Programación Genética

Podríamos definirla como un caso particular de programas de evolución, donde los individuos de la población son programas que se acercan más o menos a realizar una tarea que es la solución buscada. Es decir que aspira a la autoprogramación de las computadoras.

Codificar un conjunto de instrucciones de un lenguaje de programación en un arreglo unidimensional –cromosoma– y encontrar una forma que al recombinarlos tenga sentido no es una tarea fácil. Pero si se usa una estructura de árbol junto con ciertas reglas para evitar la generación de expresiones no válidas, se puede construir un evaluador de expresiones

primitivo que pueda producir programas simples. Este fue precisamente el enfoque tomado por John Koza para desarrollar la denominada "programación genética", en la cual se usó originalmente el lenguaje de programación LISP para aprovechar su evaluador de expresiones integrado al intérprete.

Pese a los avances de John R. Koza –discípulo de Holland– en 1992, creando árboles de análisis, aun se está trabajando en conseguir que la programación genética se pueda aplicar sobre lenguajes de programación tradicionales como C o C++, rompiendo así el mito de que sólo los lenguajes de las características propias de LISP –funcionales– son apropiados.

Programación Evolutiva

Este mecanismo fue creado por Fogel, y parte de la idea de que el genoma codifique la programación de un autómata finito –máquina de estados–. Es un planteamiento bastante elegante y fácil de implementar.

Cada gen puede consistir en una terna del tipo "estado origen + cadena reconocida + estado destino". En definitiva, cada gen es una regla que especifica las condiciones –cadena reconocida– que se deben cumplir para llegar a un estado –estado destino– suponiendo que la máquina se encuentra en un cierto estado –estado origen–. Dicho en otras palabras, en los cromosomas se codifica la función de transición del autómata.

Clasificadores genéticos

Los Clasificadores Genéticos son algoritmos evolutivos cuya finalidad es obtener un sistema clasificador. La idea fundamental consiste en evolucionar un conjunto de reglas –que son los individuos de la población– que, trabajando conjuntamente, resuelvan un problema determinado. Su principal diferencia con los Algoritmos Genéticos es que aquí la solución no se asocia a un único individuo, sino a la población completa. Todos los sistemas que siguen este modelo deben hallar un mecanismo razonable de asignación de la bondad –función fitness u objetivo– a cada individuo pues la bondad de la solución hace referencia a todos ellos.

Algoritmos miméticos

Son la combinación de un algoritmo genético, que se encarga de realizar la búsqueda aplicando el énfasis en la exploración, con otro algoritmo especializado en la explotación consiguiéndose una convergencia más rápida hacia una solución aceptable. No se intercambian los métodos entre sí según la velocidad de convergencia, sino que la búsqueda explotatoria mejora los individuos de la población.

Solidificación Simulada

Su nombre en inglés es *Simulated Annealing*. Su origen está en la Física estadística. Los procedimientos físicos de solidificación controlada consisten en calentar un sólido hasta que se funde, y seguidamente, ir enfriándolo de forma que cristalice en una estructura perfecta, sin malformaciones locales.

Se parte de una solución válida a la que se le provocan pequeñas mutaciones aleatorias. Se puede aceptar una solución peor que la anterior. A medida que avanza el algoritmo, éste cada vez será más exigente en cuanto a las soluciones aceptadas. En analogía con la solidificación física, se trata de tener un cuerpo –problema– a alta temperatura –admitiendo soluciones muy diversas a pesar de ofrecer malos resultados– e ir disminuyendo la temperatura –aumentando

la exigencia del algoritmo– de forma que termine por solidificarse según la forma deseada – ofreciendo el mejor resultado–.

Redes Neuronales. Fundamentos biológicos

Introducción

“Cualquiera que medite sobre el ser humano se enfrenta inmediatamente con la extraordinaria paradoja de que somos parte de la naturaleza y al mismo tiempo nos distinguimos profundamente de incluso nuestros más cercanos parientes, los otros primates superiores. Tenemos en común con el chimpancé casi todos nuestros genes y compartimos con todos los mamíferos cada una de las clases de componentes que integran nuestro cerebro. Sin embargo, ninguna otra especie terrestre comparte con nosotros el mundo de las narraciones, de los planes o inventarios, de la poesía o el software, de la filosofía o las matemáticas, del derecho o de la demagogia”.

Claudio Gutiérrez, *La coevolución del cerebro y el lenguaje dentro de una perspectiva humanista*

Las capacidades del cerebro, y sobre todo las del cerebro humano, han intrigado durante mucho tiempo y de hecho lo siguen haciendo, a investigadores de diversas ramas. Ya los antiguos filósofos griegos se preocupaban por el origen de nuestra capacidad de razonar. Sin embargo Aristóteles, impactado por la sangre caliente que brotaba de las heridas del corazón, creía que este último era el centro del "sensorio commune", el sitio donde se reunían todos los nervios, mientras que el cerebro era un simple refrigerador de la sangre. Fue Alcmeon de Crotona –Alcmalión el Pitagórico–, quien fue capaz de diseccionar el sistema nervioso y comprobar que los nervios de los sentidos terminaban en el cerebro. Hipócrates, el gran médico de la Antigüedad, concluía que el cerebro era el lugar del intelecto y el órgano que controla la conducta.

"No solamente nuestro placer, nuestra alegría y nuestra sonrisa, sino también nuestro dolor, nuestra aflicción, y nuestras lágrimas nacen en el cerebro, y tan solo en el cerebro. Con él pensamos y comprendemos, vemos y oímos y discriminamos entre lo repugnante y lo maravilloso, entre lo placentero y displacentero, entre lo bueno y lo malo"

Hipócrates ¹

Podríamos preguntarnos ¿Qué hace del cerebro un órgano tan especial? ¿Dónde radica la causa de su increíble potencial? Las células del sistema nervioso están compuestas por los mismos tipos de moléculas y funcionan utilizando los mismos mecanismos generales que las demás. Sin embargo poseen una característica especialísima, son células con una alta capacidad de comunicación entre ellas mismas y con otras células de distintos tipos.

Se puede decir que de la misma manera que las células del tejido muscular se especializaron como un elemento contráctil, las neuronas –células principales del sistema nervioso– lo hicieron como elementos de comunicación. Con una salvedad, su especialización ha sido tan grande que han perdido otros tipos de actividad celular como por ejemplo la de reproducción.

La capacidad funcional de las neuronas depende esencialmente de la producción y transmisión de estímulos inhibidores y excitadores, y de que una neurona se comunique con un número enorme de congéneres, dando lugar a intrincadas redes de extraordinaria complejidad.

¹ Citado por Emilio Bestoso en el artículo “El Cerebro: Enigma de la Evolución”. [Bestoso]

En los últimos años se ha avanzado mucho en este campo de la ciencia sobre todo en lo referente a la relación de circuitos cerebrales con capacidades motoras¹ pero no cabe duda que lo que falta por entender sobre estas redes neuronales excede considerablemente lo que hasta hoy se conoce.

El presente capítulo tiene por objetivo acercar al lector una visión desde la neurobiología de los conocimientos que hoy poseen los investigadores acerca del funcionamiento del cerebro relevantes en el marco de este trabajo. Por relevante en este contexto debe entenderse aquellos aspectos que de alguna manera han influido o pueden influir en el desarrollo de aplicaciones de Redes Neuronales Artificiales a las que se dedicará el capítulo próximo. Es así que se tocarán diversos aspectos del sistema nervioso del Hombre, sin hacer demasiado hincapié en la cuestión anatómica o estructural del mismo, por el contrario se prestará especial atención al aspecto funcional, evitando así apartarse del objetivo propuesto.

Se comenzará con una breve descripción de las células que componen el sistema nervioso en donde el papel protagónico pertenece sin lugar a dudas, a las neuronas, o mejor dicho, a cada uno de los distintos tipos de ellas. Los papeles secundarios son para las células gliales que interpretan actuaciones de soporte necesarias para que la función se lleve a cabo.

Luego se abordará al sistema nervioso como un sistema de comunicación en donde el órgano maestro –el cerebro– recibe información de la periferia, la procesa y la envía en forma de órdenes a los músculos y glándulas. Por periferia se entiende el medio externo, ajeno a nosotros mismos y accesible a través de los sentidos, y el medio interno conformado por todo aquello que perteneciendo a nuestro cuerpo no forma parte del sistema nervioso. Una breve discusión acerca de la comunicación molecular en los seres vivos ha sido necesaria para comprender la forma en que las neuronas se comunican utilizando mensajes químicos –los neurotransmisores–.

No podía faltar una sección dedicada exclusivamente a la arquitectura neuronal y a lo que se conoce respecto de los circuitos –o redes– neuronales. Se mostrarán algunos ejemplos de redes simples totalmente identificadas como así también la disposición en capas de complicados circuitos descubiertos en la retina –parte posterior del ojo– y en la corteza cerebral. También se presentará la organización funcional del cerebro y cómo ciertas zonas se asocian con distintas capacidades sensorio-motriz y cognitivas. Esta localización cerebral sugiere que distintas redes neuronales son activadas en función de los distintos estímulos recibidos, hecho que ha sido significativo para el nuevo modelo propuesto de neuroevolución GNE en el que se utiliza más de una red neuronal para la realización de una tarea –ver “Grupos Neuronales Evolutivos –GNE–” en la pág. 224–.

¹ Entender cómo el sistema nervioso es capaz de controlar todas las actividades motoras del organismo (las conscientes y las autónomas) ha resultado relativamente fácil sobre todo por la posibilidad de utilizar animales en la experimentación sabiendo que biológicamente no existen grandes diferencias con el hombre. Sin embargo otro tipo de capacidades, como las relacionada con la percepción, la conciencia, la creatividad, los sentimientos etc. –llamadas funciones superiores–, son más difíciles de escudriñar y la investigación con animales muchas veces se hace imposible pues se necesitaría que éstos sean capaces de comunicar lo que están sintiendo. Carl Sagan en [*Sagan_2*] reconoce que ciertos sentimientos como el amor parece haber sido inventado por los mamíferos, y que aparentemente las emociones intensas son básicamente privativas de éstos y en menor grado de las aves. Es conocida la tristeza que invade a las hembras de muchos mamíferos cuando se les arrebatan las crías, o la de un perro doméstico ante la ausencia de su amo, pero el autor se pregunta: “¿Acaso los caballos albergan a veces sentimientos de fervor patriótico? ¿Experimentan los perros hacia el nombre un cierto arrobamiento parecido al éxtasis religioso? ¿Qué otra clase de intensas y recónditas emociones albergan los animales que no comunican con nosotros?”.

Finalmente el capítulo concluye con un vistazo de las principales funciones cerebrales superiores –percepción, atención, sentido común, lenguaje, memoria y aprendizaje– sin omitir una pequeña discusión previa sobre la problemática mente y cerebro. Se pondrá particular interés en el aprendizaje y las ideas de Donal O. Hebb. Estas ideas han influido de una u otra forma en las grandes corrientes de pensamiento en neurobiología de la memoria. También son indiscutibles sus consecuencias en el campo de las ciencias de la computación donde sirvieron de inspiración para los primeros desarrollos en Redes Neuronales Artificiales que lograron hacer efectivo el denominado *aprendizaje de máquina*.

Las células del sistema nervioso

El investigador español Ramón y Cajal realizó el estudio más completo acerca del sistema nervioso que hasta la fecha se haya hecho individualmente, describiendo prácticamente todos los tipos de neuronas de las distintas regiones del cerebro, cerebelo y la retina de muchas especies animales. Demostró que el tejido nervioso estaba compuesto por células individuales, separadas una de otra, es decir, que no se continúan como se creía en esa época. Este postulado, que fue plenamente confirmado con el uso del microscopio electrónico a mediados del siglo XX, es un punto importantísimo para el entendimiento de cómo funcionan las células nerviosas y cómo se comunican entre sí, lo que veremos más adelante en este mismo capítulo.

La forma de las neuronas

Las células nerviosas, también conocidas como neuronas, constituyen la unidad neurológica del sistema nervioso. Son células muy especializadas y por ende no es raro que su estructura sea también muy particular.

Existen distintos tipos de neuronas y por lo tanto su tamaño y forma también es variable, pero en todas ellas se distingue una zona más voluminosa llamada *cuerpo* o *soma* –donde se encuentra el núcleo celular– con forma de pera, esférica o ligeramente alargada. Múltiples ramificaciones parten del soma en la que se distingue una particularmente larga llamada *axón* y muy delgada en comparación con el cuerpo de la neurona. La función del axón será conducir el impulso nervioso hacia la periferia de la célula.

Además del axón, que arranca desde un solo sitio del soma pero que puede después ramificarse profusamente –*telodrón*–, existen en las neuronas muchas otras prolongaciones que parten también del soma, generalmente del lado opuesto al del origen del axón. A diferencia de éste, son casi siempre muy numerosas desde su origen mismo, y se ramifican casi inmediatamente. Estas prolongaciones constituyen la parte receptora de información de la célula, por lo que puede decirse que son la contrapartida del axón, y reciben el nombre de *dendritas*. Dicho en términos muy coloquiales por las dendritas “entra” información, la neurona la procesa, y el resultado –si es que lo hay– “sale” por el axón y será comunicado a otras neuronas en las *sinapsis* –sitios precisos de comunicación entre las neuronas–. No obstante se aplazará la discusión sobre este maravilloso mecanismo de procesamiento y comunicación de la información para más adelante en este mismo capítulo.

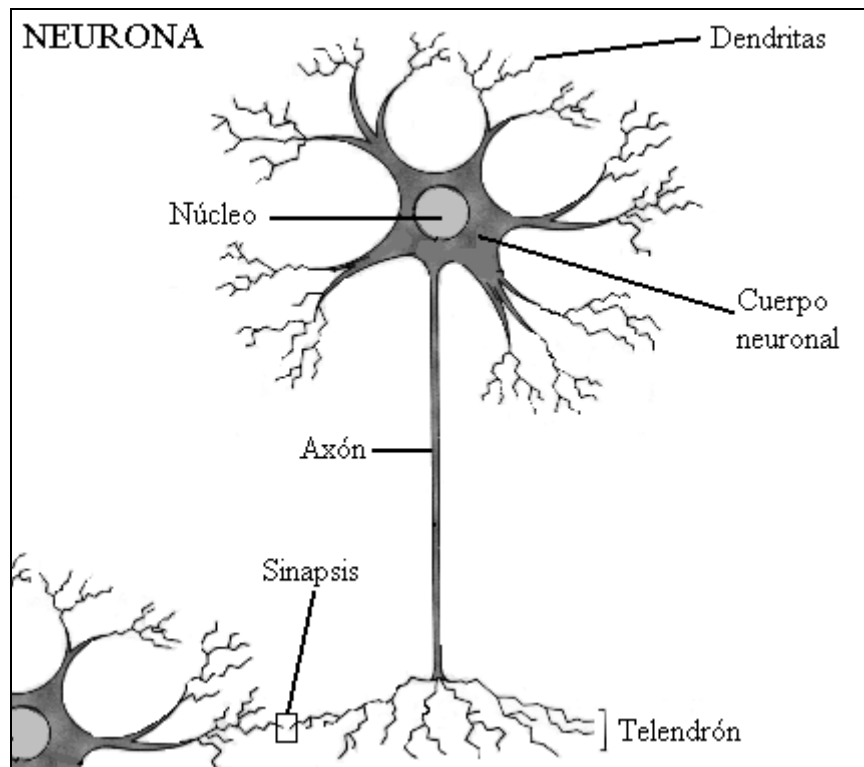


Ilustración 25 Esquema general de la forma y estructura de una neurona.

Según el tipo de neurona de que se trate su soma puede medir tanto como medio milímetro y ser visible a simple vista o tan poco como menos de 2 milésimas de milímetro. La coloración de las neuronas es oscura lo cual da el nombre de sustancia gris a las agrupaciones de estas células. Las variaciones en la forma de estas células son verdaderamente sorprendentes sobre todo si observamos sus prolongaciones. Algunas neuronas especializadas en detectar estímulos exteriores y que reciben el nombre de *neuronas receptoras* (ver más adelante –pág. 89–) pueden tener axones de hasta un metro de longitud¹ mientras que otras pueden ser mucho más cortos pues van a comunicarse con otras neuronas que se encuentran muy cerca, en el mismo interior del cerebro. El grosor de los axones, sí es muy pequeño, pues su diámetro raramente sobrepasa 3 o 4 milésimas de milímetro. La longitud de las dendritas, finalmente, es de sólo fracciones de milímetro en la gran mayoría de las neuronas.

Se puede resumir lo dicho hasta ahora sobre la forma y estructura de las neuronas de la siguiente manera: todas tienen una porción receptora de señales, formada por las dendritas, sumamente ramificadas, un soma o cuerpo –en donde se encuentra el núcleo y la maquinaria química para fabricar proteínas y otras moléculas, y además es la porción encargada de manejar e integrar la información–, y una porción emisora de la información recibida, constituida por el axón, el cual también se ramifica considerablemente. Sin embargo, estas tres bien diferenciadas porciones de cada neurona son sumamente variables en forma, en tamaño y modo de ramificarse, de tal manera que aparte de la definición general que se acaba de hacer es prácticamente imposible hablar de una morfología general de las neuronas. [Tapia96]

¹ En algunos animales grandes como por ejemplo las jirafas se han llegado a medir neuronas con un axón de más de 2 metros, correspondientes a neuronas receptoras de estímulos táctiles en sus extremidades.

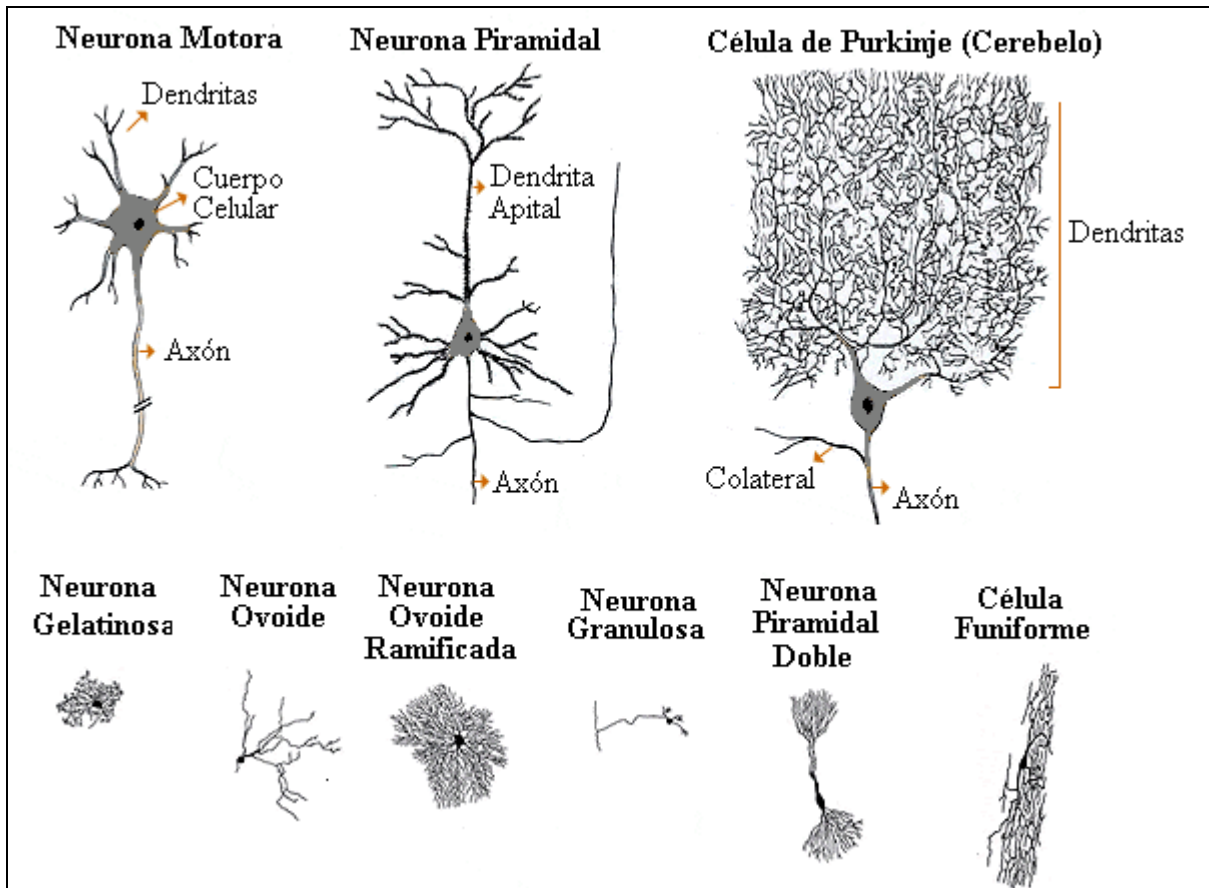


Ilustración 26 Ejemplo de algunos tipos de neuronas.

Las otras células del cerebro

Las neuronas no son las únicas células que existen en el cerebro, por el contrario, aún más numerosas que ellas se encuentran las *células gliales* o simplemente *neuroglía*. La función de estas células es de soporte a las neuronas y no pueden considerarse de naturaleza nerviosa puesto que no reciben, no procesan ni envían información a otras células.

Las células gliales más abundantes son las llamadas *astrocitos*, que proporcionan a las neuronas sustancias de varios tipos, incluyendo algunas de gran importancia para la función neuronal, pues tienen que ver con la fabricación de las moléculas que la neurona usa para comunicarse con sus vecinas, como veremos más adelante en este capítulo. Tienen un cuerpo de forma irregular, presentan numerosas ramificaciones. Se adosa a algunas de las estructuras neuronales o sobre capilares sanguíneos.

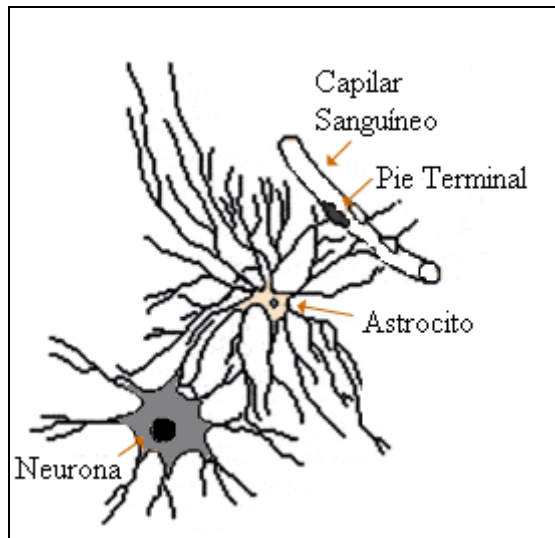


Ilustración 27 Astrocito.

Otras células gliales de suma importancia son los *oligodendrocitos* en el sistema nervioso central, cuyas prolongaciones se enrollan alrededor de los axones propiciándoles un recubrimiento de *mielina*, y las *células de Schwann* en el periferia, también encargada de la mielinización. La mielina está constituida por un material que, por su alto contenido en grasa, es un buen aislante eléctrico lo que permite aumentar la velocidad de conducción eléctrica por el axón. (para más información ver Comunicación entre neuronas –pág. 95–).

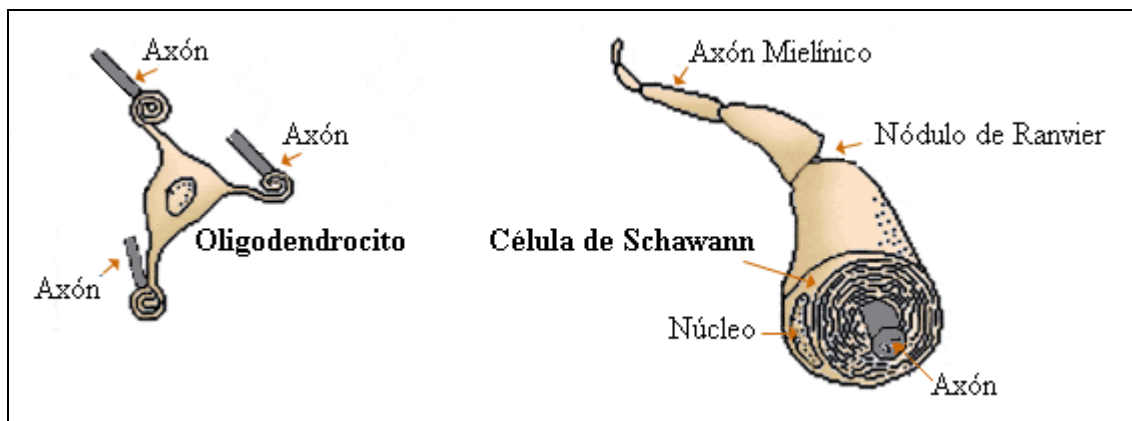


Ilustración 28 Oligodendrocito y célula de Schwann proveen el recubrimiento de mielina a los axones de las neuronas. Los nódulos de Ranvier interrumpen periódicamente la vaina de mielina a lo largo del axón.

Aunque ciertamente las células de la neuroglia no son células nerviosas propiamente dichas, en conjunto dan un soporte y un apoyo funcional esencial para la actividad neuronal. El cerebro queda así definido como un gran conjunto de neuronas, apoyadas por la neuroglia, organizado de manera precisa en redes, circuitos, capas y conexiones. Esta organización se verá más adelante en este capítulo.

El Sistema nervioso. Especialista en comunicación

Una de las funciones más evidentes del sistema nervioso son aquellas que nos permiten comunicarnos con el *medio exterior*, con el ambiente que nos rodea, en dos direcciones, de afuera hacia nosotros, a través de los sentidos, y de nosotros hacia fuera por medio de nuestras capacidades motoras.

Pero además existe otro medio que llamaremos *medio interior*, formado por nuestro propio organismo que también es controlado por el sistema nervioso generalmente de forma involuntaria, autónoma y que conforma un rico sistema de comunicación esencial para nuestras funciones vitales.

Se analizará aquí brevemente esta función del sistema nervioso y el papel que juegan en ella dos tipos especiales de células: las *neuronas receptoras* y las *neuronas motoras*. Cuando en el capítulo siguiente veamos las redes neuronales artificiales, podrá apreciarse la relación que existen entre estas células con los elementos de la capa de entrada y salida en las arquitecturas de redes neuronales artificiales.

Neuronas receptoras

Es conocido por casi cualquier persona que en el hombre son cinco los sentidos que utilizamos para informarnos del mundo que nos rodea y que existe un órgano diferente y especializado para cada uno, a estos órganos se los denomina órganos de los sentidos. Asociado a ellos existen células particulares capaces de captar una sensación específica. Así en el ojo son sensibles a la luz, en el oído a la vibración que el sonido produce en la membrana del tímpano, en la mucosa nasal a ciertas moléculas volátiles que llegan a ella y así en otras percepciones.

Estos elementos sensibles a distintos estímulos no son otra cosa que algún tipo de neurona. Estas células nerviosas de los órganos de los sentidos poseen una región muy especializada en uno de sus extremos –ver Ilustración 29– para captar los estímulos específicos según el órgano de que se trate. Pero es obvio que tales neuronas receptoras deben ser capaces de conducir la información que reciben hasta algún lugar en que sean procesadas convenientemente, es decir, hacia el cerebro, órgano maestro del sistema nervioso, o hacia la médula espinal, encargada del control de algunas funciones menores. Para ello, se valen, ni más ni menos que del axón, el cual puede ser muy largo en el caso de las neuronas receptoras del tacto, o relativamente corto, como en las neuronas que perciben la luz en la retina del ojo o las olfatorias que reconocen los olores desde la parte más alta del interior de la nariz.

Pero además existen neuronas receptoras especializadas en captar información proveniente del interior del organismo necesaria para ajustar los engranajes de esta delicada máquina de precisión. Aunque en este caso, la información no llega a nivel de conciencia como sucede con los sentidos y como veremos más adelante, muchas veces ni siquiera llega al cerebro sino que se resuelve a nivel de la médula espinal. Este mundo interior es también extraordinariamente rico en información y de su correcto funcionamiento depende, por supuesto, que todo marche bien. Como ejemplo pensemos lo que sucede cuando flexionamos o extendemos un brazo utilizando la articulación del codo. Existen músculos flexores y extensores de actividad opuesta cuando uno se contrae el otro debe relajarse pues de lo contrario el brazo quedaría inmovilizado. De alguna manera el músculo flexor debe recibir la orden de relajarse cuando el extensor se contrae y viceversa, esta coordinación necesita de la información de estado interno del músculo de actividad opuesta –músculo antagonista–. Y de manera similar a lo que sucede con los sentidos y la información del mundo exterior, existen neuronas receptoras a estos estímulos internos, en este caso particular, a la tensión de los músculos, es decir a qué tanto están contraídos o relajados.

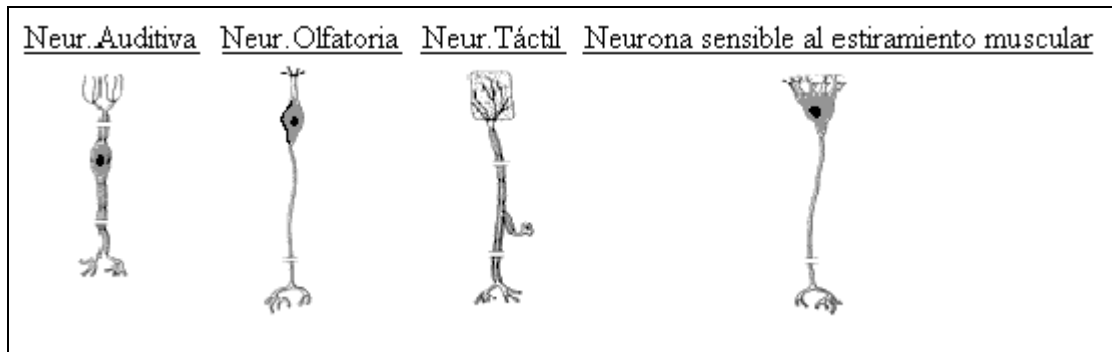


Ilustración 29 Algunas neuronas receptoras. Estas neuronas se han especializado en recibir un tipo específico de estímulo, mediante las estructuras que se observan en la porción más superior de cada una de ellas.

El ejemplo anterior sobre la relajación del músculo antagonista ilustra un tipo de *reflejo*. Hay una diferencia muy importante entre estas neuronas receptoras de estímulos internos y aquellas que reciben los estímulos externos. La información que estas últimas reciben debe llegar al cerebro para que podamos ver, oír, oler, etc. En cambio, la información de las primeras no se hace consciente, porque la respuesta apropiada al estímulo en cuestión se produce sin necesitar que la información llegue a las regiones del cerebro encargadas de hacer conscientes los estímulos. Así, en nuestro ejemplo de los músculos que se oponen, los flexores y los extensores, la información llega sólo hasta la médula espinal para que se establezca la regulación correcta entre la contracción de un músculo y la relajación de su antagonista. A este tipo de mecanismo de funcionamiento involuntario e inconsciente se lo denomina reflejo.

Un ejemplo conocido por todos es el reflejo rotuliano, donde un estímulo exterior es capaz de excitar al receptor de manera similar al estímulo interior. Como resultado, la pierna se levanta levemente sobre la articulación de la rodilla, y este movimiento es completamente involuntario, pues todo sucede en la médula espinal y la información no llega al cerebro¹

¹ Se puede hallar una analogía con un sistema de cómputo convencional. Así como el procesador de una computadora delega ciertas funciones en dispositivos secundarios dotados de una inteligencia limitada llamados controladores, es como si el cerebro (órgano maestro de nuestro sistema nervioso) hace lo propio con ciertas redes neuronales que controlan ciertas funciones desde la médula espinal.

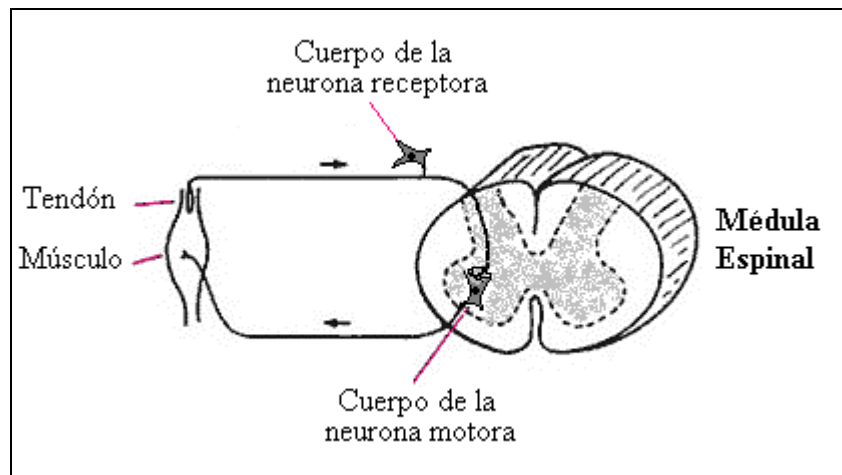


Ilustración 30 Reflejo rotuliano. Una neurona receptora detecta el estiramiento del tendón y conduce la información hasta la médula espinal. El soma de la neurona receptora está cerca de la médula espinal, y su prolongación penetra a la médula por su región posterior y transmite la información a una neurona motora, la cual a su vez envía la información al músculo para que éste se contraiga. La información no llega al cerebro, el músculo se contrae involuntaria e inconscientemente de manera refleja

Además del antagonismo de los músculos flexores y extensores, existen innumerables ejemplos que podríamos citar en los cuales se ejerce una acción reguladora automática a través de la recepción de información interna. Tal es el caso del mecanismo capaz de detectar cuánta azúcar tenemos en la sangre, cuál es la presión arterial, cuánta sangre está circulando, el grado de llenado de la vejiga, qué tan ácida está la sangre, qué tan distendido está el estómago y muchos otros más. En todos estos casos las neuronas receptoras captan el estímulo correspondiente y lo comunican a la zona de la médula espinal o a regiones que no tienen que ver con la conciencia del cerebro, en donde, sin que estemos conscientes de ello, se activan los mecanismos que originarán una respuesta adecuada al estímulo en cuestión.

Neuronas motoras

La forma en que nuestro organismo se manifiesta hacia el exterior es a través de los músculos. Y otra vez es el cerebro, órgano maestro del sistema nervioso, quien asistido por la médula espinal y las terminaciones nerviosas ejerce una acción sobre los músculos para responder a estímulos exteriores y para manifestar lo que en él se ha procesado.

Para ello se vale de un tipo de células nerviosas, las *neuronas motoras*, llamadas así por su influencia sobre los músculos que se traducen en movimiento. Estas neuronas llevan información desde el cerebro –o desde algún circuito de la médula espinal– hacia los músculos para que éstos se puedan contraer, y así podamos expresarnos. Constituyen la contrapartida de las neuronas receptoras de las que hablamos anteriormente, conduciendo información en sentido contrario.

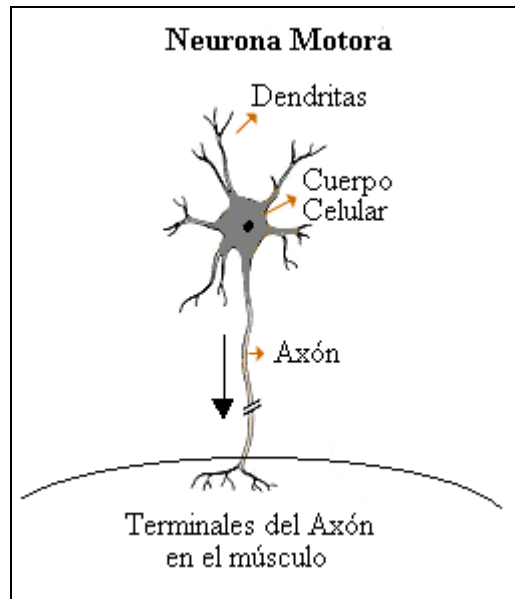


Ilustración 31 Esquema de una neurona motora. La información es recibida en las dendritas, procesada en el cuerpo y enviada hacia el músculo a través del axón, el cual puede ser muy largo, hasta de más de un metro, pues el cuerpo de la neurona se encuentra en la médula espinal y los músculos pueden estar muy alejados, como los que mueven los dedos de las manos o, aún más, los de los pies.

Las neuronas motoras no sólo sirven para que podamos expresarnos por medio de la contracción muscular, sino también intervienen en los movimientos involuntarios que ya mencionamos anteriormente. Estos movimientos de los que no tenemos conciencia son los relacionados con el llamado *sistema autónomo*. Las neuronas motoras del sistema autónomo son las que entre otras cosas provocan que se contraigan los músculos que a su vez contraen las arterias para aumentar la presión arterial, se acelere la frecuencia cardíaca, se muevan el estómago y el intestino para que se digieran los alimentos, o se secreten a la sangre las hormonas que regulan la química sanguínea. Todas estas acciones son automáticas y no dependen de la voluntad, lo cual nos beneficia enormemente liberándonos del control consciente de estas actividades. En relación a ello, Ricardo Tapia, investigador titular del Instituto de Fisiología Celular de la UNAM, un estudioso del funcionamiento del sistema nervioso, en su libro “Las Células de la Mente” nos dice:

“Basta con considerar, por ejemplo, qué difícil sería si cada vez que comemos tuviéramos que secretar conscientemente la saliva a la boca, el jugo gástrico de la pared del estómago hacia el interior del mismo, el jugo intestinal hacia la luz del intestino delgado; y que, además, así como debemos masticar voluntariamente la comida, debiéramos también hacer que el estómago iniciara sus movimientos característicos de la digestión en este órgano, y que después tuviéramos que lograr que el alimento parcialmente digerido pasara del estómago al intestino y que éste se contrajera con su peculiar peristaltismo que va empujando al bolo alimenticio por su interior, y simultáneamente debiéramos secretar las enzimas digestivas que componen el juego intestinal, y por si fuera poco hacer que se contraiga la vesícula biliar para que la bilis, que fue formada en el hígado y se almacenó en la vesícula, se vierta al intestino a ejercer su importante papel en la digestión de las grasas. Y además de todo esto, tendríamos también que preocuparnos por relajar los músculos de la pared de las arterias del tubo digestivo, pues después de comer debe aumentar el caudal de sangre que llega a él, pero sin que disminuya la sangre que llega al cerebro, que debe mantenerse a toda costa, sin olvidar además al corazón, de cuya frecuencia de contracción depende también que haya la suficiente presión arterial.”

Ricardo Tapia, Las Células de la Mente

Ciertamente, resulta afortunado contar con un sistema autónomo que regula la comunicación con nuestro medio interno a base de reflejos que por su misma naturaleza no requieren que la información sea consciente y que por lo tanto funcionan involuntariamente.

Queda claro que nuestro sistema nervioso puede dividirse en dos partes en cuanto a su función: la de relación con el medio externo, que también se llama somática, y la autónoma relacionada con el medio interno que también se llama vegetativa. Pero además, el sistema nervioso también se puede dividir en dos grandes partes en cuanto a su organización estructural: la que lleva información entre los medios interno y externo hacia el cerebro, la cual se llama *sistema nervioso periférico*, y la parte que procesa esa información y emite las señales hacia la periferia. Esta parte es el *sistema nervioso central*, y está constituida por la médula espinal, que se encuentra en el interior de la Columna vertebral, el tallo cerebral, porción superior de la médula que se "introduce" en el cerebro, y el cerebro mismo, encerrado y protegido por los huesos del cráneo y al que llega directamente –sin pasar por la médula espinal– la información de la vista, el oído, el olfato y el gusto, así como el tacto y la sensibilidad de la cara.¹ [Tapia96]

Ahora se abordará el tema de cómo se comunican las neuronas y que tipo de información intercambian. Para ello primero se dedicarán unas líneas a un interesante mecanismo de comunicación presente en los organismos vivos, la comunicación a través de las moléculas.

Acerca de la comunicación

Las neuronas, como cualquier célula viviente, están constituidas por átomos organizados en moléculas, y las moléculas tienen también una particular organización, muy precisa, de la cual depende que la célula pueda vivir llevando a cabo todas las funciones. No debe sorprender que los mecanismos de comunicación entre las neuronas sean moleculares, es decir, realizados mediante ciertas moléculas específicamente organizadas para permitir y facilitar dicha comunicación interneuronal.

Por lo tanto, antes de analizar cuál es la naturaleza de la información que “llevan y traen” las neuronas, en qué consiste esa información y cómo la comunican entre sí y a otras células, se deben realizar algunas consideraciones respecto al *lenguaje de bajo nivel* que utilizan las moléculas para transmitir información.

Información molecular y reacciones químicas en la célula

Toda célula viva es un gran laboratorio de pequeñas dimensiones donde se realizan multitud de reacciones químicas esenciales para el funcionamiento vital de la célula. Estas reacciones químicas se realizan a una enorme velocidad gracias a un tipo especial de grandes moléculas conocidas con el nombre de proteínas. Una sola célula contiene miles de proteínas diferentes, y por supuesto el número de moléculas de cada proteína es también muy grande.

Ya se han comentado algunos aspectos relacionados con las proteínas en el capítulo “Evolución y genética. Una mirada desde la biología” en el apartado “Genes y cromosoma” –pág. 32– donde se les reconocieron dos importantes funciones. O bien forman parte de la estructura celular –membranas y orgánulos– o bien catalizan –hacen posible o aceleran– las

¹ Salvando la distancia, esta segunda clasificación nos recuerda nuevamente a los sistemas de cómputo convencionales, donde existe un componente especializado en la comunicación con el exterior, denominado subsistema de Entrada/Salida, que correspondería a las funciones del Sistema Nervioso Periférico, y un componente central (procesador y memoria) que realiza las funciones análogas a las del Sistema Nervioso Central.

reacciones químicas en el funcionamiento del organismo. Aquí se repasará este segundo aspecto relacionado con su particular propiedad de poder reconocer otros tipos de moléculas y acoplarse a ellas. Este reconocimiento se lleva a cabo en función de su forma tridimensional en el espacio que debe “encajar” con la forma de la otra molécula que reconoce. Así se establece un mecanismo de comunicación nunca visto antes de su aparición sobre el planeta: la comunicación por medio del reconocimiento de la *forma específica* de cada molécula. De esta forma, una molécula proteica reconocerá específicamente sólo a otra molécula y no a ninguna otra, por más cerca que se encuentre.

Aprovechando esta capacidad para reconocer moléculas específicas, las proteínas pueden actuar sobre dichas moléculas para acelerar miles de veces la velocidad de sus reacciones de interconversión, mediante el proceso que los químicos conocen como catálisis. Cada proteína cataliza una reacción específica y ninguna otra, haciendo uso de la información que posee en su forma tridimensional. Veamos un esquema:

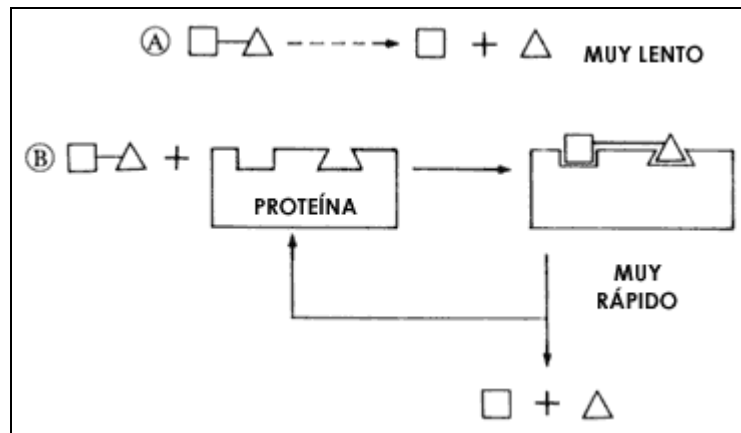


Ilustración 32 A) Rompimiento de la molécula en cada uno de sus componentes en ausencia de una proteína catalizadora –lento o imposible-. B) Igual pero en presencia de la proteína catalizadora, se facilita cientos de veces el proceso del rompimiento, por lo que la reacción se acelera esos mismos cientos de veces; además la proteína queda libre para repetir el proceso. –Ilustración obtenida de [Tapia96] –.

Información molecular y mensajes: transmisores

Queda claro que si la llegada de una molécula específica produce una determinada reacción y la llegada de otra una distinta, de alguna manera podemos decir que estas moléculas transmiten información, algo así como una “directiva de lo que se debe hacer”.

Algunas células secretan proteínas que actúan sobre otras moléculas o sobre otras células. Por ejemplo cada una de las células de las glándulas que participan en la digestión, desde las salivales que secretan saliva hasta las de la pared del estómago que hacen el jugo gástrico, secretan proteínas que van disueltas en su respectivo líquido. Estas proteínas son capaces de reconocer ciertas moléculas que son ingeridas en los alimentos y que deben ser convertidas en otras más simples, más pequeñas, que puedan ser absorbidas desde el intestino y aprovechadas por todas las células del organismo. Así, es posible afirmar que las células secretoras de estas proteínas digestivas transmiten la información de ellas hacia la boca o hacia el tubo digestivo. Este es un interesante ejemplo de especificidad de la información de una proteína trabajando en otro sitio diferente a la propia célula que la produce.[Tapia96]

Otro ejemplo interesante donde se aprecia el mecanismo de reconocimiento de las moléculas proteicas lo encontramos en nuestro sistema inmunológico. Los anticuerpos, que también son

proteínas, son capaces de reconocer a proteínas ajenas y dañinas, como las de las bacterias o los virus.

Muchas proteínas, se comportan como mensajeros entre células. El ejemplo más demostrativo es el de las hormonas de naturaleza proteica. Las hormonas se sintetizan en las células de las glándulas endocrinas, y se secretan hacia la sangre para que sea transportada a otras células, sobre las que ejercerá alguna acción. Según el tipo de la hormona de que se trate será la clase de efecto que produzca sobre la célula sobre la que actúa. La célula que recibe la acción de la hormona debe tener en su membrana externa, que es lo primero con lo que la hormona se va a encontrar, otra proteína con la afinidad suficiente para identificar a la hormona. Gracias a esta identificación la célula podrá interactuar específicamente con la hormona y como resultado de esta interacción se ejercerán sus efectos.

A las moléculas proteicas que se encuentran en la membrana de las células, que pueden reconocer a otras moléculas e interactuar con ellas para que se produzca una determinada respuesta celular, las llamamos *moléculas receptoras*, o simplemente *receptores*. Aquellas moléculas portadoras del mensaje químico se las llama *moléculas transmisoras* o simplemente *transmisores*.

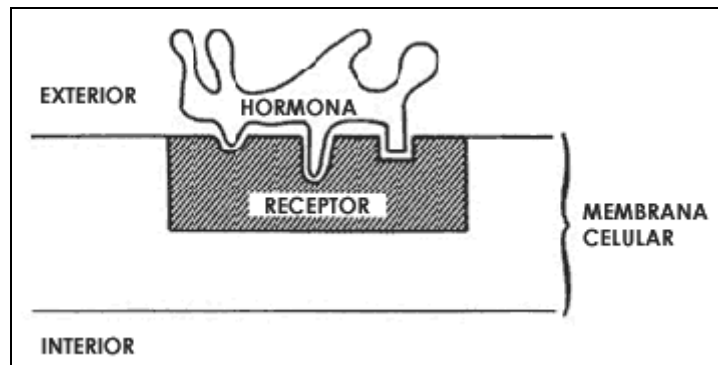


Ilustración 33 La hormona constituye un mensajero químico cuya información está contenida en su forma tridimensional.

Comunicación entre neuronas

No existe ninguna otra célula viviente que se haya especializado tanto en la comunicación como las neuronas abandonando prácticamente toda otra función celular, aún la de reproducirse.

La información que comunican estas células recibíendolas por las dendritas y emitiéndolas por el axón que permite a nuestro sistema nervioso interactuar con nuestro medio externo e interno, y que también permite que se lleven a cabo todo los procesos mentales en alguna parte del cerebro como razonar, memorizar, evocar, imaginar, sentir etc., esta información es de naturaleza eléctrica.

Cada axón, por consiguiente, genera en su sitio de origen una corriente eléctrica que será conducida –a mayor velocidad si éste se encuentra recubierto por mielina– a lo largo del mismo hasta el sitio donde termina y establece comunicación ya sea con un músculo, una glándula endocrina u otras neuronas.

Pero como postulara Ramón y Cajal, las neuronas no se continúan unas con otras, no comparten la membrana celular, sino por el contrario están perfectamente delimitadas y separadas de cualquier otra célula. Así las terminaciones del axón y las dendritas se encuentran separadas físicamente y se comunican por medio del espacio sináptico. La

pregunta ahora es ¿cómo puede llevarse a cabo esta comunicación con otra célula si ni siquiera están en contacto?

Las fotografías de microscopía electrónica indican que si bien existen algunas sinapsis de apenas 2 nanómetros de espesor –2 millonésimas de milímetro– en donde la conducción eléctrica es factible y de hecho así sucede, la gran mayoría de estos espacios de comunicación son de unos 20 nanómetros, lo suficientemente grandes como para que la corriente eléctrica no pueda “saltarlo”.

Esto hace necesario algún mecanismo para transformar la información eléctrica en algún otro tipo capaz de traspasar los espacios sinápticos¹. Ese mecanismo consiste en que, como efecto directo de la llegada del impulso eléctrico hasta la terminal del axón en la neurona presináptica, ésta libera una sustancia química a través de su membrana, la cual atraviesa el espacio de 20 millonésimas de milímetro que la separa de la siguiente neurona y así llega a ella. Al igual que las hormonas requieren de una molécula receptora en la membrana de la segunda neurona –neurona postsináptica– para ser reconocidas como mensajeros y la comunicación se lleve a cabo. Estas moléculas mensajeras reciben el nombre de *neurotransmisores* porque se forman en las neuronas.

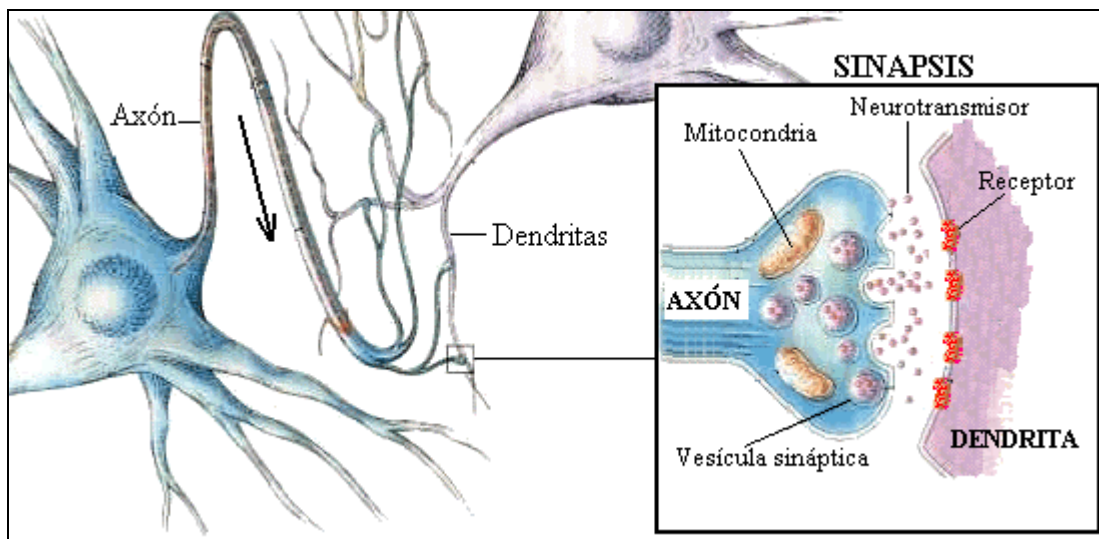


Ilustración 34 Sinapsis química. En las terminaciones del axón se encuentran mitocondrias –para el aporte de energía– y pequeñas vesículas que contienen moléculas de neurotransmisor

Una vez reconocido y aceptado el neurotransmisor mediante el receptor correspondiente en la membrana, la neurona receptora o postsináptica sólo puede hacer dos cosas en cuanto a excitación eléctrica se refiere: emitir la señal eléctrica o no emitirla. Si emite la señal, una vez que ésta se ha iniciado en el sitio de origen del axón, la corriente eléctrica será indefectiblemente conducida hasta todas las terminales de dicho axón, sin que la neurona pueda hacer algo para impedirlo.

Pero ¿cómo puede excitarse eléctricamente la neurona receptora del neurotransmisor? El mecanismo es muy interesante. Como en el medio externo a las neuronas hay átomos que tienen carga positiva, especialmente sodio, cuando el transmisor se combina con el receptor se

¹ Algo similar al proceso de modulación que realizan los modems para convertir una señal digital en otra capaz de ser transmitida por una línea telefónica.

abren unos pequeños canales en la membrana por donde penetran rápidamente estos átomos de sodio. Dado que el sodio tiene carga positiva, su entrada genera cambios en las características eléctricas de la membrana, y la consecuencia de estos cambios es que la neurona se excita al grado de generar en su axón los impulsos eléctricos¹.

También existen neurotransmisores que al combinarse con sus receptores en la siguiente neurona, causan una *inhibición* y no una excitación de ella, es decir, disminuyen la probabilidad de que esa neurona dispare señales eléctricas por su axón. Es más, las neuronas que tienen esta acción inhibitoria sobre otras neuronas son mucho más numerosas que aquellas de cuya acción resulta una excitación. Cuando la comunicación entre las neuronas es de naturaleza inhibitoria, se abren canales para átomos cargados negativamente, los cuales al penetrar al interior de la neurona la inhiben.

Esto se explica de la siguiente manera: la proteína que actúa como receptor en la membrana de las neuronas, al reconocer y aceptar a un determinado neurotransmisor sufre un cambio en su estructura tridimensional el cual produce una modificación en otras proteínas. Estas últimas son canales para sodio (Na^+) o para cloro (Cl^-), por lo que al modificarse por efecto de la combinación transmisor-receptor dejan pasar estos átomos. Debido a que, los átomos de sodio y de cloro tienen una carga positiva o negativa, respectivamente, al entrar, cambian las propiedades eléctricas de la membrana: cuando entra el sodio hay excitación de la neurona y cuando entra el cloro hay inhibición. El transmisor que produce excitación es distinto al que produce inhibición, y naturalmente también sus correspondientes receptores son diferentes, pues reconocen distintas moléculas.

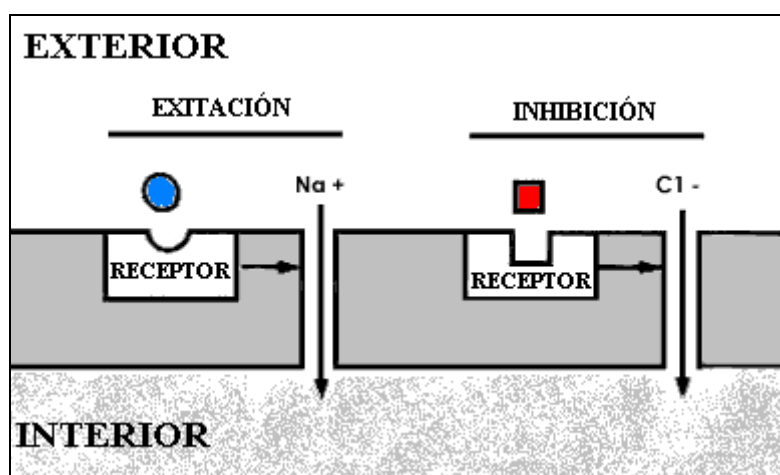


Ilustración 35 Excitación e inhibición de las neuronas.

Una vez generado el impulso eléctrico –llamado *potencial de acción*– se transmitirá a lo largo del axón como resultado de una serie de despolarizaciones sucesivas que tienen lugar en los nódulos de Ranvier. Estos nódulos interrumpen periódicamente la vaina de mielina que recubre el axón –ver Ilustración 28–. Cuando uno de los nódulos se despolariza se desencadena la despolarización del siguiente, así el potencial de acción viaja “a los saltos” de

¹ Siguiendo la comparación realizada en la nota al pie anterior, ahora estamos en presencia de la operación inversa a la modulación: la *demodulación*, obteniendo nuevamente una señal con las características de la original, en este caso un impulso eléctrico. No obstante esta comparación deja de ser pertinente al considerar que la neurona receptora no siempre se excitará eléctricamente, es decir la demodulación no siempre se llevará a cabo. Además una neurona recibe información no por uno sino por miles de canales (dendritas) y como veremos más adelante todos deben contemplarse para “decidir” la respuesta neuronal (excitación o no excitación).

un nódulo a otro de manera mucho más veloz que en el caso de un axón sin recubrimiento de mielina. Una vez que el impulso eléctrico ha pasado por un cierto punto, ese punto no puede volver a ser excitado durante un lapso aproximado de 1 milisegundo, que es el tiempo que tarda en volver a su potencial de reposo.¹

Pero cada neurona recibe miles de terminales nerviosas cada una proveniente de otra neurona. La pregunta ahora es cómo la neurona maneja toda esa información que le llega simultáneamente tomando en cuenta que esa información es de naturaleza distinta, pues muchas de esas señales serán inhibitoras mientras que otras serán excitadoras. Recordemos que lo único que puede hacer la neurona con toda esta información es enviar o no un impulso eléctrico por el axón.

Aquí asume un rol importante el cuerpo de la neurona que será quien procese la información de entrada. En efecto el soma debe integrar, es decir sumar algebraicamente toda la información inhibitora o excitadora que recibe a cada instante para obtener un resultado final que se traducirá en un impulso eléctrico si existe la suficiente carga excitatoria.

Este hecho pone de manifiesto algo muy interesante. De alguna manera podemos considerar a las neuronas como elementos binarios pues sólo admiten dos estados en cada momento, o bien se encuentran en reposo o bien activas generando un impulso eléctrico por su axón. Se puede deducir entonces que el inmenso potencial del cerebro humano descansa en la compleja estructuración de elementos sumamente sencillos.

Ventajas de la comunicación química

La primera gran ventaja de la comunicación química es la posibilidad de inhibición. Ya se dijo incluso que este tipo de estímulo es más común que el exitatorio. Si la comunicación fuese eléctrica, está claro que no podría haber inhibición tan sólo excitación o ausencia de estímulo. Solamente mediante el procedimiento de los neurotransmisores que al acoplarse con su receptor abren canales para átomos con carga negativa, es posible que las neuronas se inhiban.

Otra ventaja es la naturaleza unidireccional de la comunicación sólo de la terminal axónica a la neurona siguiente, pero no a la inversa. Recordemos que los impulsos eléctricos fluyen como en un cable, con una comunicación eléctrica esas corrientes podrían pasar de una neurona a otra y viceversa, es decir, la comunicación entre dos neuronas sería bidireccional, lo cual podría tener una serie de consecuencias indeseables en términos del control de la información a lo largo de determinados circuitos.

Además de estas ventajas, la existencia de la comunicación química interneuronal representa una ganancia adicional enorme: la capacidad de regular o controlar la transmisión de información entre las neuronas que no podría realizarse con una comunicación eléctrica². Por ejemplo, la neurona puede regular la velocidad de síntesis y de destrucción de los

¹ Como puede verse, nuestras pequeñas y simples unidades de procesamientos poseen un reloj interno de apenas 1 kilohertz. Evidentemente la potencialidad del cerebro no radica en la velocidad de procesamiento de cada neurona sino en el alto grado de paralelismo (10^{11} elementos de procesamiento) e interconexión.

² Esta facultad está directamente relacionada con la *plasticidad neuronal*, la memoria y el aprendizaje que se verán más adelante en este mismo capítulo –pág. 118–. Además en el próximo capítulo tendremos oportunidad de ver cómo esta capacidad es la base fundamental en la mayoría de los algoritmos de aprendizaje en las distintas arquitecturas de redes neuronales artificiales en donde las arquitecturas fijas “aprenden” cambiando la importancia o peso de cada conexión (sinapsis) regulando así la transmisión de información a través de la red.

neurotransmisores como también controlar la liberación del mismo desde la terminal axónica, con lo cual se puede gobernar la comunicación. Pero también la neurona receptora puede aportar lo suyo haciéndose más sensible o insensible a un neurotransmisor determinado. Por lo tanto, mediante cualquiera de los mecanismos señalados se puede modificar la comunicación interneuronal haciendo que la transmisión de información sea flexible, variable a lo largo del tiempo. Se supone que la capacidad de aprender del cerebro está íntimamente ligada a esta posibilidad de controlar la transmisión de información entre las neuronas que participan en intrincados circuitos de comunicación, a los que se dedicarán unos párrafos, más adelante en este capítulo.

Ahora quisiera cerrar este apartado citando nuevamente a Ricardo Tapia que refiriéndose a la comunicación interneuronal nos dice:

“...la existencia de los mecanismos químicos de la transmisión de información permite que dicha transmisión sea modificable, maleable, plástica. ¿Y no es precisamente ésta una de las propiedades más sorprendentes del cerebro, si pensamos en su capacidad de memorizar, de dar marcha atrás, de reconsiderar en función de nuevos datos, de aplicar los conocimientos o las experiencias previas a nuevos actos, de recordar, de asociar eventos, de recapitular, en una palabra de aprender? ...

Es muy poco lo que sabemos respecto al funcionamiento de grandes grupos de neuronas trabajando sincrónicamente para dar lugar a nuestros sentimientos, placeres, actos inteligentes, pensamientos, reflexiones, creaciones, conciencias, remordimientos, arrepentimientos, dudas, odios, iras, pasiones, tristezas y alegrías...”

Ricardo Tapia. Las Células de la Mente

El poco conocimiento que los neurobiólogos poseen respecto a cómo se organizan los grupos de neuronas para formar intrincados circuitos capaces de llevar a cabo las complejas funciones cerebrales, es más que suficiente para inspirar a otro tipo de científicos, los de las ciencias de la computación, al desarrollo y a la investigación en el campo de las redes neuronales artificiales, que en definitiva, es el punto que interesa en el presente trabajo.

En el próximo apartado se expondrán aquellos resultados más sobresalientes que los estudiosos del cerebro han sacado a la luz acerca de la arquitectura neuronal y su funcionamiento.

Arquitectura neuronal

Un cerebro humano, sin contar el cerebelo que se encuentra dentro de la parte posterior del cráneo, posee aproximadamente 10^{11} neuronas, es decir, nada menos que 100 mil millones de estas especializadas células. Para tener una idea más acabada de esta enorme cantidad pensemos que si quisiéramos contarlas a razón de una por segundo, tardaríamos alrededor de 3.171 años en hacerlo.

Sin embargo ese número tan elevado de células en el cerebro no es tan sorprendente al compararlo con el que existe en otros órganos del cuerpo, por ejemplo es posible que el hígado, cuyo tamaño es considerablemente mayor que el del cerebro, tenga más células que este último.

Lo que distingue al cerebro de otros órganos desde el punto de vista anatómico, sin lugar a dudas, es la peculiar organización de las células que lo conforman, organizadas en intrincadas redes de conexión. Las neuronas no se encuentran distribuidas al azar, una junto a la otra, sino que están organizadas de una manera extraordinariamente precisa, de tal modo que mediante sus dendritas y sus axones forman complicadas vías o circuitos, a través de los cuales cada una de ellas se puede comunicar sólo con aquellas que forman parte de la vía o circuito.

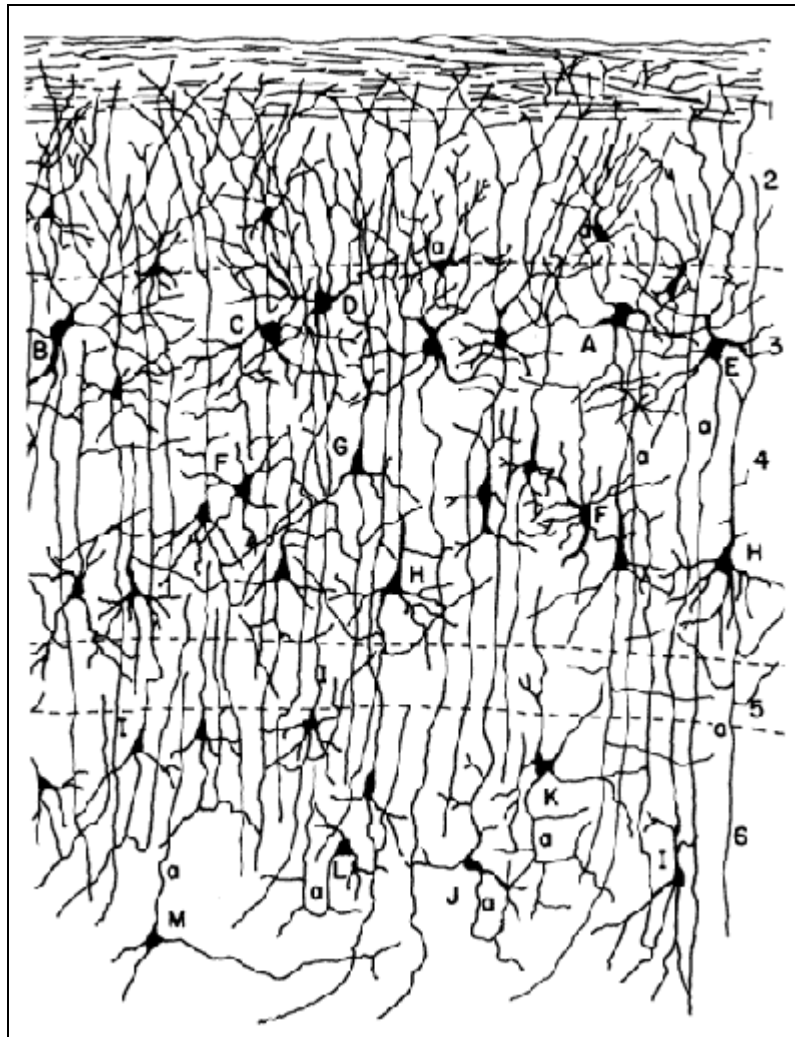


Ilustración 36 Copia de un dibujo realizado por Ramón y Cajal de la corteza cerebral de un niño. –Ilustración obtenida de [Castrillon95] –.

Ahora podemos hacernos algunas preguntas: ¿Cuál es la importancia de esta arquitectura neuronal? ¿Qué posibilita la existencia de estas intrincadas redes de comunicación celular que no puede llevarse a cabo con una organización más sencilla como por ejemplo al estilo de las células del hígado o las musculares? Parte de la respuesta nos la brinda el astrofísico Carl Sagan en [Sagan_2] donde realiza una serie de reflexiones interesantísimas sobre las capacidades de los organismos vivos de almacenar información genética y extragenética.

Casi todos los organismos actúan en buena medida conforme al legado genético de que son portadores y que ha sido previamente transmitido al sistema nervioso del individuo, siendo la información extragenética recogida en el curso de su vida un factor secundario. Sin embargo en el caso del Hombre y los demás mamíferos sucede justamente lo contrario. Nuestros cerebros ofrecen muchísimas más oportunidades de establecer nuevos modelos de conducta y nuevas pautas culturales en cortos períodos de tiempo que en cualquier otro ser vivo. El difícil proceso de maduración del niño, marcado por una larga infancia, viene compensado por su capacidad de aprendizaje, lo que incrementa en gran manera las posibilidades de supervivencia de la especie humana.

Ya se ha visto a lo largo del capítulo “Evolución y genética. Una mirada desde la biología” cómo es que la información genética de todo ser vivo se encuentra codificada en el ADN – ácido desoxirribonucleico– de cada una de sus células. También se ha mencionado que esta

supermolécula, contiene toda la información necesaria para construir el organismo que la porta, información que se codifica por medio de una secuencia determinada de cuatro tipos de bases nucleótidas: Adenina, Timina, Citosina y Guanina, –puede consultarse el apartado “Genes y cromosoma” en la página 32– para obtener información más precisa al respecto.

Conociendo cuantos nucleótidos posee la molécula de ADN en cada especie de organismo vivo podemos calcular la cantidad de información en bits –alfabeto binario– que cada especie puede almacenar –cada nucleótido equivale a 2 bits de información–. Así, los virus almacenan unos 3000 bits de información genética, lo que equivale a decir que son suficientes 375 bytes –menos de ½ Kb. – para reconstruir completamente uno de estos especímenes¹. Una bacteria, organismo unicelular muchísimo más complejo que los virus almacena unos 122 Kb., un reptil más 119 Mb. y el ser humano unos 1192 Mb.²

Retomando la pregunta que nos hacíamos párrafos arriba respecto al potencial de la peculiar organización neuronal, y suponiendo que cada sinapsis puede estar en cada instante en uno de dos estados –activa o en reposo– la arquitectura neuronal del cerebro humano ofrece una enorme capacidad de almacenamiento aún sin considerar la existencia de circuitos específicos. Parece que muchas neuronas de la médula espinal poseen alrededor de 10.000 sinapsis, y es posible que las llamadas células de Purkinje, en el cerebelo, tengan todavía más. En cuanto al número de conexiones por neurona en la corteza cerebral, es probable que sean menos de 10.000. Sabiendo que hay aproximadamente 10^{11} neuronas un cálculo sencillo nos lleva a concluir que existen entre 10^{14} y 10^{15} sinapsis³. Tomando el número más chico (10^{14}), el número de estados mentales alcanzables es de $2^{100000000000000}$ –pues supusimos que cada sinapsis es binaria– o sea, 2 multiplicado por sí mismo 100 billones de veces. Según Sagan esta cifra irrepresentable, es mucho mayor, por ejemplo, que el número de partículas elementales –protones y electrones– que existen en todo el universo, número bastante inferior a 2^{1000} .

“Debido a esta ingente cantidad de configuraciones cerebrales funcionalmente distintas no puede haber dos hombres iguales, ni si quiera dos gemelos monovitelinos que se hayan criado juntos. Esta cifra colosal puede explicar también, hasta cierto punto, el porqué de la imposibilidad de predecir la conducta humana y el hecho que en un momento dado lleguemos a sorprendernos de nuestros propios actos. Y, ciertamente, a la vista de tales magnitudes, es realmente asombroso que existan pautas regulares de conducta en el hombre. La única respuesta válida es la de que en modo alguno se han agotado la totalidad de estados cerebrales y que, por consiguiente, existe un ingente número de configuraciones mentales no experimentadas, ni siquiera atisbadas por el ser humano a lo largo de la historia de la humanidad.”

Carl Sagan, Los Dragones del Edén

Estos cálculos sobre la cantidad de configuraciones mentales, como bien señala el autor, no tienen en cuenta la existencia de las redes neuronales, sólo la cantidad de sinapsis como si se tratara de un arreglo unidimensional de ellas. Claramente la existencia de estos circuitos

¹ Recordemos que los virus y viroides son los organismos más simples que se conocen, incluso puede decirse que se encuentran en un nivel de organización entre la materia inerte y la viva. Son entidades orgánicas compuestas tan sólo de material genético, rodeado por una envoltura proteica protectora, es decir, ni siquiera alcanzan el status de organismo unicelular. Carecen de vida independiente pero se pueden replicar en el interior de las células vivas, perjudicando en muchos casos a su huésped en este proceso.

² Estos datos han sido calculados en base a la información tomada de [Sagan_2]

³ Este número debe considerarse una estimación. Existen diferencias en la bibliografía consultada que en general señalan un número mayor de 10^{14} sinapsis.

neuronales hacen que el número de estados mentales sea superior al calculado anteriormente. Sólo pensemos cuántas topologías o arquitecturas de redes distintas pueden formarse con 100 mil millones de nodos interconectados por unos 10.000 links que parten de cada uno de ellos.

Puede apreciarse que la información extragenética es extraordinariamente superior a la genética en el caso de los seres humanos, mientras que en los organismos inferiores ni siquiera existe. Sólo a título de curiosidad comentaré que la evolución biológica logró producir organismos capaces de almacenar más información en sus cerebros y sistema nervioso que en sus genes sólo a partir de la aparición de los reptiles. En los anfibios y animales inferiores la relación es inversa.

Probablemente, según Sagan, sólo un mecanismo de aprendizaje extragenético puede afrontar el rapidísimo proceso de transformación que soporta la especie humana. En este sentido, la rápida evolución del intelecto humano que hoy se observa es, por un lado, la causa y por otro, la única solución concebible a los muchos y graves problemas que el Hombre debe enfrentar en estos tiempos.

Pero ¿Donde está la diferencia que nos ubica en el ranking intelectual, tan arriba de cualquier otro ser viviente en este planeta? Según Claudio Gutiérrez en [Gutierrez95] parece ser una cuestión de cantidad y no de calidad pues la total continuidad neurológica en el orden de los mamíferos es un hecho comprobado. Ni un solo tipo de neurona, ni siquiera un sólo tipo de neurotransmisor son exclusivas del género humano. Lo que parece de gran importancia es la masa crítica y la conectividad resultante.

El Hombre posee cuatro veces más corteza cerebral que el chimpancé –su primo más cercano y con quien comparte el 99 por ciento del genoma– dando origen a las áreas "vacantes" o desocupadas en el sentido que no están comprometidas con las funciones sensoriales o motoras. Estas neuronas vacantes que comprenden el 75% de la corteza cerebral, y sus respectivas conexiones nos han servido a los seres humanos para practicar algo que, dentro del reino animal, nos es exclusivo: "el pensamiento abstracto". Debe notarse que poseer este alto porcentaje de neuronas desocupadas no quiere decir, como algunas personas mal informadas suelen decir, que haya partes de nuestro cerebro ociosas o que nunca hayan sido utilizadas. Lo que sí quiere decir es que nuestra especie puede dedicar la mayor parte de su corteza para funciones de carácter simbólico que el cerebro del mono no cubre, como lenguaje, análisis de lo que percibimos, planeamiento y otras funciones cerebrales superiores.

Circuitos neuronales

La compleja organización de las neuronas permite que distintos grupos de estas células en regiones específicas del cerebro –llamados núcleos neuronales– se comuniquen directamente sólo con ciertos núcleos neuronales de otras regiones, siguiendo una determinada y precisa distribución. Así se forman los circuitos –o redes– neuronales.

Existen diferentes tipos de circuitos neuronales: sensoriales, motores, cognitivos, de regulación de modulación. Cada tipo de circuito presenta características propias, particulares, que dependen de las propiedades de las neuronas que los forman y de las sinapsis que ellas forman.

Un ejemplo simple de circuito o red neuronal es el que ya presentamos al comentar el reflejo rotuliano –ver Ilustración 30 pág. 91– donde el papel de cada neurona interviniente es claramente apreciable.

Veamos ahora un ejemplo apenas más complejo. El circuito neuronal que interviene en el movimiento de un miembro que pivotea sobre una articulación puede observarse en el siguiente esquema:

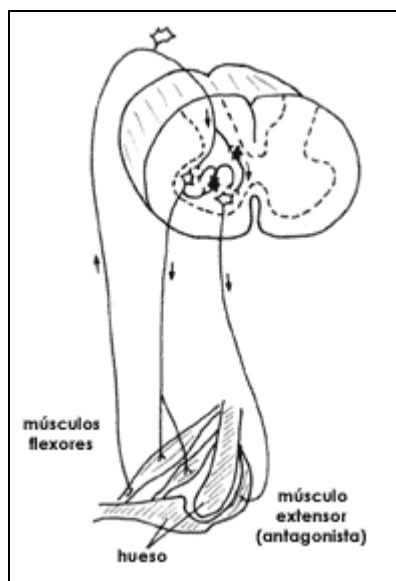


Ilustración 37 Mecanismos de regulación de músculos agonistas y antagonistas

Cuando el músculo flexor es excitado, como en la Ilustración 37, la neurona sensitiva excita a la motoneurona que hará contraer a ese músculo, pero al mismo tiempo, mediante una colateral, excita a una neurona inhibidora –en negro– de la médula espinal, la cual inhibirá la motoneurona del músculo antagonista extensor. Además el axón de la motoneurona del músculo flexor emite una colateral que excita a otra neurona inhibidora –también de negro– que a su vez inhibirá a ambas motoneurona –la del flexor y la de su antagonista–, proporcionando así un mecanismo regulador que permite una contracción más controlada de todos los músculos que participan en la movilidad de la articulación.

Es claro que la existencia de neuronas inhibitoras dentro de los circuitos neuronales conlleva la posibilidad de regular, controlar o modular el funcionamiento de ellos y, lo que es quizá más importante, permite la posibilidad también de modular la actividad de otros circuitos neuronales opuestos, al tiempo que se activa el requerido.

Más intrigante y mucho más complejo resulta la disposición y organización de la red neuronal que se encuentra en la retina, en la parte posterior del ojo. Mediante ella percibimos la luz en toda sus variantes e intensidades. Esta red inicia el complejo procesamiento de la percepción visual y está organizada en varios estratos o capas –ver Ilustración 38–. A partir de las neuronas receptoras de luz del fondo de la retina existen dos capas de neuronas de axón muy corto y una última cuyos axones se reúnen, constituyendo el nervio óptico que llega al cerebro en donde termina de procesarse la información. También existen otras neuronas de axón muy corto, que se conectan con sus pares de la misma capa permitiendo una ordenada interacción lateral así como regulación o control.

Al igual que en la retina, en la corteza cerebral vuelve a repetirse este esquema de organización en capas donde predominan los somas de las células, mientras que sus dendritas y sus axones se dirigen hacia los planos superior o inferior estableciendo contactos con neuronas vecinas de otras capas. También algunas de axones muy cortos no abandonan la zona, sino que se comunican con otras neuronas de la misma capa. La importancia de este tipo de arreglos es grande pues por tener estas células su axón corto y comunicarse con otras de la

misma región, funcionan como células de asociación y, como tales, regulan la información que toda esa zona procesa. Más precisamente se han descrito 6 capas celulares en la corteza cerebral aunque luego se han identificado subcapas en cada una de ellas.

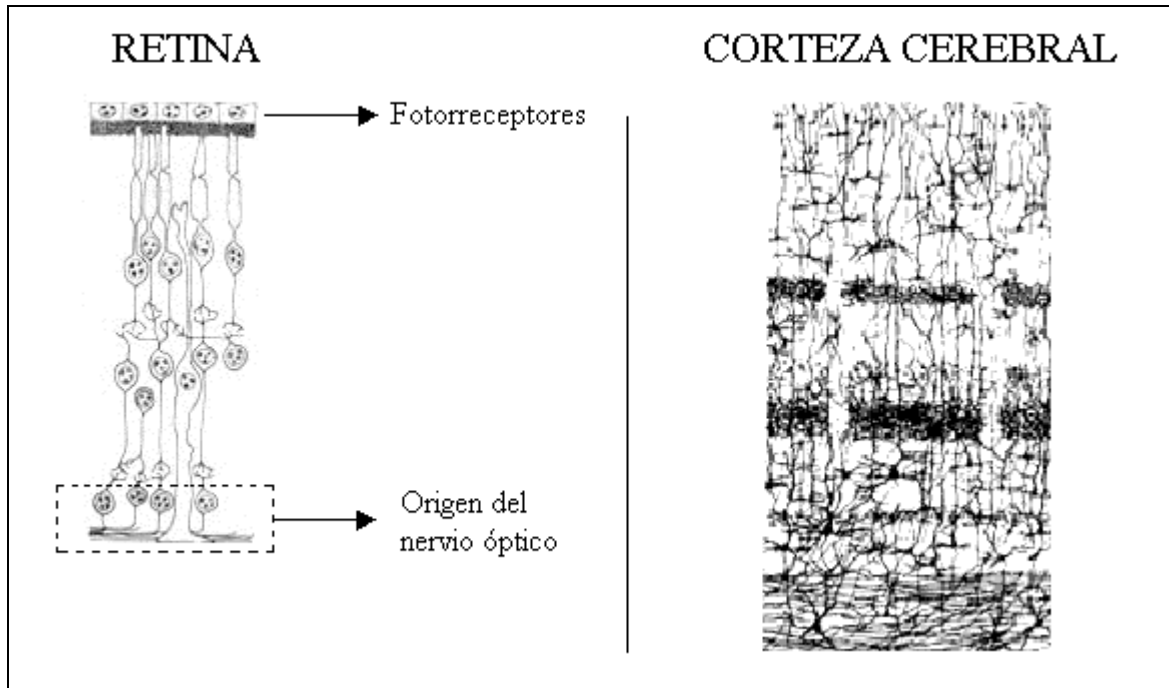


Ilustración 38 Esquema de la retina y corteza cerebral. Puede observarse claramente la organización de las neuronas en capas y circuitos localizados en láminas celulares muy delgadas.

También se observa cierta organización macroscópica considerando diferentes partes del cerebro. Los axones que emergen de una región se dirigen con suma precisión hasta otra zona del cerebro. Además es frecuente que los axones regresen a la misma región de la que se originaron, de manera que se establecen circuitos cerrados en los que la información que parte de una región puede repercutir sobre esa misma región –redes recursivas o recurrentes–. Este tipo de organización permite la posibilidad de mecanismos de regulación precisos, mediante la retroalimentación de información.

Podemos decir que las neuronas, mediante las redes que establecen constituyen la parte fundamental del cerebro al que se puede definir como un enorme conjunto de neuronas extraordinariamente organizadas en sus arreglos tridimensionales y en sus conexiones.

“Es con las neuronas que pensamos, sentimos, nos movemos, captamos todo lo que está a nuestro alrededor en forma de luz, imágenes, sonidos, ruidos, frío, calor, dolor, suavidad o dureza, dolor o placer, olores agradables o desagradables, sabores, consistencias, texturas y configuraciones, colores y formas. Es con las neuronas que creamos, imaginamos, sufrimos, amamos y gozamos.

Las neuronas son, por todo lo anterior, las células de la mente”

Ricardo Tapia, Las Células de la Mente.

Zonas del cerebro. Distintas funciones

Centro neuronales en la corteza cerebral

Experimentos en humanos llevados a cabo alrededor de 1950 por Jackson y Penfield revelaron la existencia de una notable localización funcional en la corteza cerebral.

Determinadas partes de las capas superficiales del cerebro envían o reciben señales de ciertas partes del cuerpo. En la Ilustración 39 puede apreciarse el croquis de la corteza motora y sensorial según Penfield. En ellos se refleja con claridad la importancia proporcional de las distintas partes de nuestro cuerpo. La gran extensión de área cerebral que corresponde a los dedos –sobre todo al pulgar–, así como a la boca y a los órganos del habla, corresponde exactamente a lo que en el plano fisiológico, y por vía del comportamiento, nos diferencia de la inmensa mayoría de las restantes especies animales. En cierto modo, el esquema de la corteza motora es una fiel representación de nuestra condición humana. [Sagan_2]

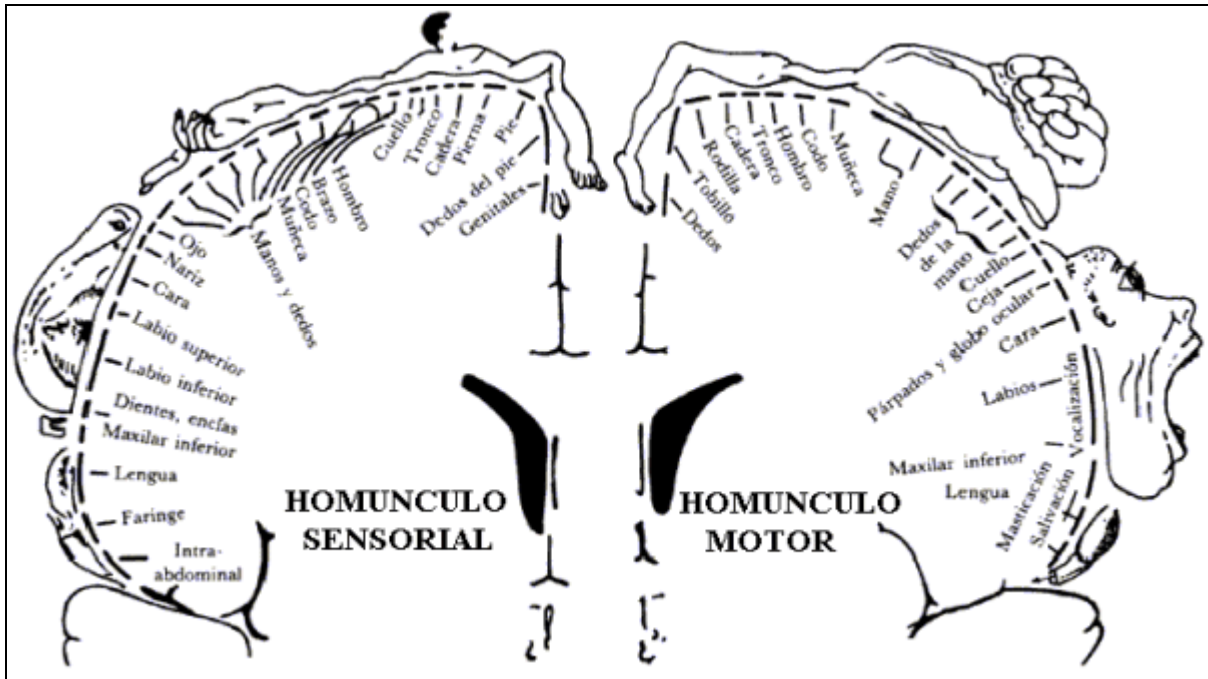


Ilustración 39 Homúnculo sensorial y motor, según Penfield. Las figuras humanas distorsionadas muestran la atención que la corteza concede a los distintos órganos del cuerpo humano. Cuando mayor es la parte del cuerpo expuesta, más importancia reviste.

Pero en la actualidad las pruebas que poseen los científicos acerca de la localización de las funciones son aún más sólidas. David Hubel, de la Facultad de Medicina de Harvard, a través de una serie de experimentos, descubrió la existencia de formaciones reticulares de células cerebrales que responden selectivamente a la percepción ocular de una serie de líneas orientadas según diversas direcciones. Así, existen células que detectan líneas horizontales, otras las verticales, unas terceras las diagonales, y cualquiera de ellas sólo resulta estimulada cuando percibe líneas con la orientación apropiada. Por tanto, ya se han registrado ciertos indicios indudables de pensamiento abstracto en las células cerebrales.

Centros neuronales en el interior del cerebro

En el interior del cerebro también existen grupos de neuronas que se han relacionado con ciertos aspectos específicos, por ejemplo con el control fino de movimientos musculares, o inclusive con otras funciones superiores relacionadas con los procesos mentales.

Hoy se sabe que funciones complejas como la actividad intelectual se localizan sólo parcialmente en la corteza cerebral y se concentran más bien en otras estructuras del cerebro situadas debajo de ella. En particular, el conjunto de estructuras que se conoce como *sistema límbico* –punteado en Ilustración 40– tiene gran importancia en el origen y el control de las emociones. Dentro de este gran circuito, una pequeña región, el *hipotálamo*, está asociada a muchas conductas emocionales y a funciones como el hambre y la sed.

Experimentos con ratas han demostrado que estimulando con un electrodo ciertas zonas del sistema límbico, dependiendo del punto estimulado se consiguen reacciones que hacen presumir que el animal siente ira, tranquilidad, placer y desagrado.

Hay todavía interesantes ejemplos de centros cerebrales aparentemente responsables de otras funciones específicas. Una lesión o la destrucción del hipotálamo en la base del cerebro, hace que una rata empiece a comer descontrolada y vorazmente, sin saciarse aparentemente y volviéndose, en consecuencia, extraordinariamente obesa, por lo que se ha concluido que dicha zona corresponde al centro de la saciedad. De manera similar, se han descubierto otros centros como el de la sexualidad.

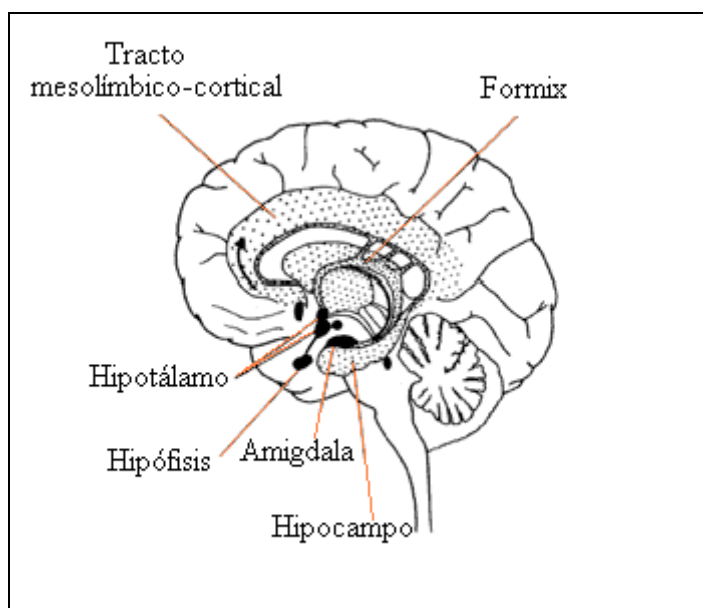


Ilustración 40 localización del hipotálamo en el cerebro. El hipotálamo forma parte de un conjunto de estructuras cerebrales conocidas como sistema límbico –punteado– que participan de manera importante en la modulación de las emociones.

No parece ilógico extrapolar al ser humano estas observaciones hechas con animales, ya que las diferencias con el Hombre, en lo que se refiere a su comportamiento biológico, no es tan grande. Los científicos saben que la enorme diferencia que existe entre una rata y un ser humano común, por no hablar de las mentes privilegiadas como Kant o Einstein, radica no en una diferencia en los principios generales con los que opera el sistema nervioso, que son exactamente los mismos, sino en la extrema complejidad de las conexiones interneuronales y tal vez en otros elementos que aún desconocemos. Resulta interesante que el problema mente-cerebro, es decir, el de la localización celular de las funciones mentales superiores, no se ha resuelto aún, y es uno de los grandes retos de la neurobiología moderna. [Pasantes97]

Es posible imaginar, a la luz de estos sencillos experimentos, que la diferencia entre un individuo colérico y otro apacible puede ser que en el primero estos centros de la agresividad en el hipotálamo estén menos controlados por acciones inhibitoras de otras neuronas, o más activados por una preeminencia de neuronas excitadoras. El mismo razonamiento podría aplicarse a los centros hipotalámicos del hambre y la saciedad e imaginar que esa afición por la comida, tenga una explicación, en parte, en el tipo de control que la corteza u otras estructuras ejercen sobre los núcleos del hipotálamo.

La extrapolación podría parecer bastante simplista, pero no deja de tener su contraparte experimental cuando se sabe que la administración de ciertas drogas, como las anfetaminas, que precisamente actúan aumentando la eficiencia de algunas conexiones neuronales del tipo

de las que se encuentran en el hipotálamo, da como resultado una pérdida casi total del apetito, además de modificar espectacularmente muchos rasgos del carácter del individuo. [Pasantes97]

Las funciones superiores

La problemática mente y cerebro

Además de las funciones sensoriales y motoras del sistema nervioso hay otras capacidades funcionales de este sistema como el aprendizaje, la memoria, las motivaciones, los matices de percepción, el lenguaje, la capacidad de planificar, el pensamiento abstracto, etc. que se enmarcan en las llamadas funciones superiores. Estas funciones han estado ligadas desde hace siglos al concepto de la *mente*. Podríamos ahora preguntarnos: ¿La mente es una entidad separada del cerebro que se vale de él para manifestarse –*dualismo*– o es sólo el producto de la actividad neuronal –*monismo*–?

La discusión sobre este tema no es fácil, y ha sido objeto de un debate tan largo como la filosofía misma, desde los griegos hasta los filósofos y neurocientíficos contemporáneos.

Para Platón la mente –*psyche*– era un órgano que se ocupaba de las ideas puras y la considerada completamente separada del cuerpo. Este estado de cosas se mantuvo hasta la época de Descartes –nacido en 1596– el cual trata de usar los métodos científicos, al alcance en la época, para comprobar los hechos de la mente. Es decir, considera a la mente un hecho observable. No obstante René Descarte, influenciado por el *platonismo* abrazó la idea del dualismo, y propone que mente y materia son dos tipos diferentes de substancia, de naturaleza opuesta, capaces de existir de manera independiente. Sin embargo les concede cierta capacidad de interacción.

Por otro lado, desde la antigüedad se conoce que la memoria humana opera utilizando principios asociativos. Estos principios han sido resumidos en leyes de contigüidad, que relacionan en tiempo y lugar unos objetos con otros, y leyes de refuerzo e interferencia, que nos indican que la presencia de estímulos positivos refuerza las conductas y la de estímulos negativos produce su extinción u olvido. De hecho, algunas de estas ideas son expuestas por Aristóteles, de manera cualitativa, en sus tratados acerca de la memoria y la remembranza. [Castrillon95]

La medición de procesos mentales se abordó posteriormente con G.T. Fechner –alrededor de 1850–, quien se dedicó a desarrollar métodos para medir procesos mentales. Este enfoque fue exaltado más aún por el médico alemán W. Wundt, fundador del primer laboratorio de métodos psicológicos, alrededor de 1880. Este investigador centró los estudios sobre la mente en el estudio de las sensaciones. Pero también se empieza a desarrollar en esa época el concepto de vida mental, que empieza a ser concebida más como un grupo de actos que de sensaciones. Nace así la Psicología y se incorporan a los estudios relacionados con la mente investigadores notables como William James que impulsó un gran desarrollo de la Psicología Experimental.

La corriente filosófica actual denominada *reduccionista*, supone que las percepciones corresponden a ciertos estados neurofisiológicos, por lo cual, al reproducirse un cierto estado neurofisiológico, se evoca la percepción correspondiente. De manera que sentir dolor es para el cerebro estar en un estado neurofisiológico determinado. Esta corriente es llamada en ocasiones teoría del paralelismo psicofísico de Spinoza o William James. Las imágenes ópticas virtuales se explican por el mismo principio: para crear una ilusión es suficiente

reconstruir los planos de onda ópticos, los cuales son réplicas aproximadas de las imágenes originales.

La Nación Line (<http://www.lanacion.com.ar>) publicó el 16 de noviembre de 2000 un artículo titulado “Identifican neuronas individuales que intervienen en la imaginación” donde se da cuenta del descubrimiento por parte de un investigador argentino de cierta representación neuronal para los procesos de visión y de imaginación y de cómo fue posible predecir qué estaban viendo o imaginando los pacientes en un experimento.

“...Un trabajo que se publica hoy en Nature, y cuyo autor es un joven científico argentino, Gabriel Kreiman, logró por primera vez cartografiar la actividad de neuronas individuales mientras sus dueños visualizaban mentalmente diferentes imágenes. Dicho de otro modo, lo que logra el estudio es comenzar a trazar una precisa radiografía de la imaginación.

... Kreiman les mostró a sus sujetos de investigación fotos de personas famosas y también figuras de animales, pinturas abstractas y otras imágenes. Las figuras aparecían en un monitor durante un segundo, mientras observaba qué neuronas se activaban. Luego se les pedía que cerraran los ojos e imaginaran las mismas figuras. ‘Encontramos que había neuronas que se encendían (mostraban actividad) solamente si los pacientes estaban viendo determinado tipo de imágenes –explica Kreiman–. Pero, más sorprendente aún, eran las mismas que se activaban cuando imaginaban las mismas figuras.’...”

Nora Bär, Redacción de La Nación

Durante el siglo XIX, se encuentran también algunas teorías importantes que daban alguna explicación acerca de las capacidades mentales del hombre. En particular, la *frenología* fue una hipótesis fisiológica que se desarrolló de manera empírica y que tuvo un gran auge, aun hasta principios del siglo XX, y se opone diametralmente al dualismo. Fundamentada en que el cerebro es el órgano de la mente y todos los procesos mentales son una combinación de un número definido de facultades independientes localizadas en un área específica de la superficie cerebral intenta hacer predicciones sobre la conducta, intelectualidad y moralidad de las personas simplemente analizando visualmente la forma de su cráneo. Esta corriente sostiene que el tamaño de cada región del cerebro asociada a una determinada facultad es directamente proporcional a la importancia que ésta tiene en la persona estudiada. Este hecho combinado con la supuesta capacidad del cráneo para adaptarse a la corteza cerebral hace posible que a través de una simple inspección ocular pueda determinarse ciertas características de la personalidad del individuo.

La hipótesis frenológica fue desechada científicamente, ya que las descripciones de individuos hechas a partir de la observación de sus protuberancias craneales no coincidían con la conducta que éstos manifestaban.

Hoy día, la postura más aceptada es la del monismo. Las nuevas técnicas de imagen, los últimos avances en el estudio de la estructura cerebral y de las sustancias que participan en la transmisión de mensajes químicos y eléctricos a través del organismo han demostrado que la causa de los estados de ánimo, de las enfermedades mentales y de otras patologías importantes como la epilepsia tienen casi siempre una causa física. Y, como estados físicos que son, pueden ser alterados con la farmacología.

En 1895, un neuropsiquiatra llamado Sigmund Freud escribió un trabajo que pasó totalmente inadvertido llamado Proyecto para una Psicología Científica. En él se proponía por primera vez que los mecanismos de la mente –normales o anormales– podían ser explicados con

sistemas cerebrales. Freud estaba en lo cierto, pero, sin embargo, desvió su camino, se olvidó de su proyecto, y se dedicó a otros menesteres durante el resto de su vida¹. Ahora, cuando ha pasado un siglo, la neurociencia ha retomado el proyecto visionario de Freud. [LopezBlanco97]

Entender el pensamiento es entender la mente. El pensar tiene que ver con la conciencia, la memoria, el aprendizaje y su estudio y el de esos parámetros acercará a la comprensión de la mente y de la actividad mental.

Veamos ahora brevemente alguna de las llamadas funciones cerebrales superiores que, por cierto, son las más interesantes desde la óptica de la Inteligencia Artificial.

La percepción

En el ser humano la percepción sensorial está siempre asociada a un proceso cognoscitivo. Existe una diferencia sutil, y muy importante, entre las funciones de "sentir" y "percibir". Mientras que la primera es una operación simple a nivel de los sentidos –cinco en los seres humanos–, la segunda es una función compleja a nivel cerebral.

El “sentir” es innato, pero el “percibir” debe ser aprendido y depende de las experiencias individuales, del entorno y de las características personales. Se han hecho estudios sobre personas ciegas de nacimiento que recuperaron la visión en la edad adulta que ponen de manifiesto la necesidad de aprender a percibir imágenes visuales. En un principio, estas personas, aunque se percatan de estar viendo una figura geométrica, no pueden reconocerla. Luego de unos meses de haber recobrado la vista son capaces de discriminar por ejemplo un triángulo de un rectángulo pero sólo contando sus esquinas. Es decir, si bien poseen el concepto de las diferentes figuras geométricas, pues terminan por reconocerlas, aún no han aprendido a relacionarlas con las imágenes percibidas. Aparentemente es más sencillo aprender a relacionar el concepto de vértice o esquina con la imagen correspondiente que hacerlo con figuras que obviamente son más complejas.

También se ha encontrado que no existen conexiones innatas entre la identificación auditiva del sonido producido por un objeto y su identificación visual. Todas estas asociaciones de ideas deben establecerse a través de la experiencia y forman una parte importante del aprendizaje que se lleva a cabo en niños pequeños. Más adelante expondré el condicionamiento de Pavlov relacionado con este concepto.

El tipo de procesamiento que realiza el cerebro humano para transformar las sensaciones en percepciones puede ponerse de manifiesto a través de las denominadas ilusiones ópticas. Debo aclarar que haré hincapié en las percepciones visuales por ser una de las más representativas pero debe considerarse que los mismos mecanismos se aplican a cualquier información que procese el cerebro proveniente de alguno de los cinco sentidos del Hombre.

¹ Debo destacar que la figura de Freud en relación al psicoanálisis ha sido cuestionada por la epistemología moderna. Según el conocido filósofo de la ciencia argentino Mario Bunge, el psicoanálisis no tiene carácter científico y se enmarca en las denominadas pseudociencias.

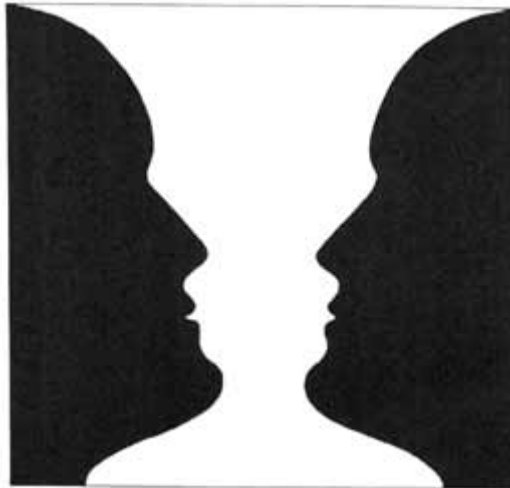


Ilustración 41 Jarrón o dos rostros enfrentados. Cuando una imagen observada es ambigua, la imagen que se hace conciente, llamada Gestalt, será una de las alternativas disponibles. Aunque se puede cambiar de Gestalt, únicamente una de las alternativas puede ser observada a un mismo tiempo.

La misma situación, aunque mucho más grato de contemplar la encontramos en gran parte de la obra de Maurits Cornelius Escher, la reproducción siguiente es un ejemplo de ello.

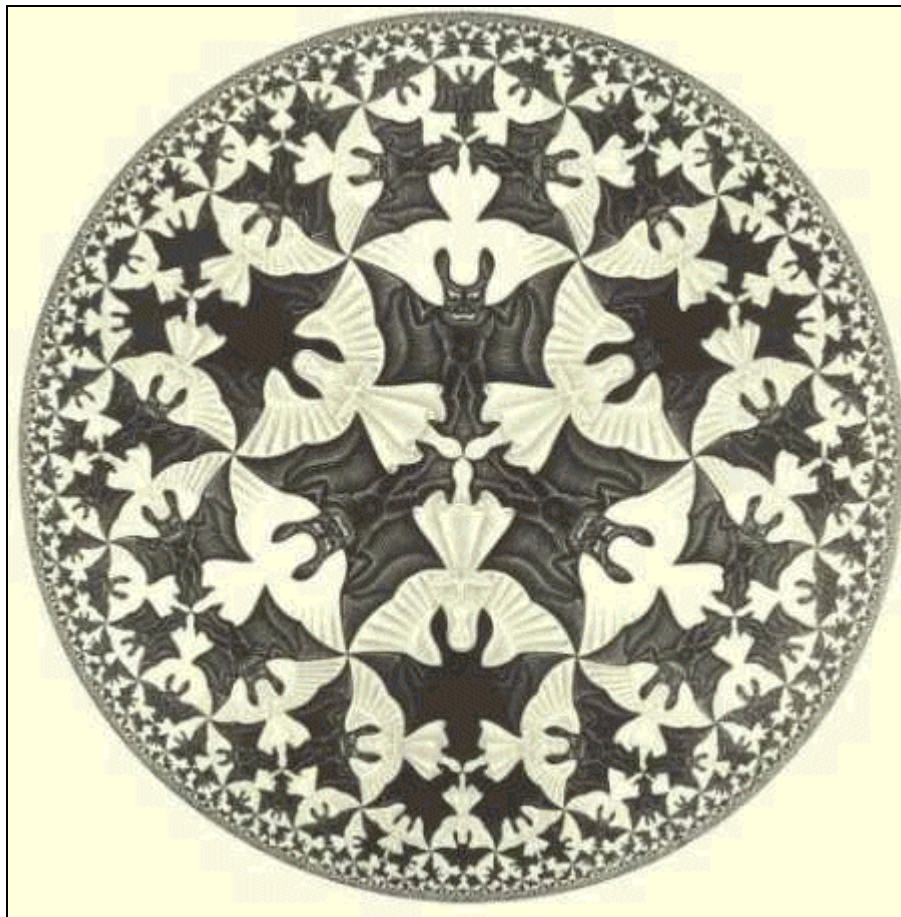


Ilustración 42 Cielo e infierno, 1960, grabado en madera. El autor escribió: “La frontera entre dos figuras adyacentes tiene una función doble y es complicada. A cada cual de sus límites, aparecen simultáneamente seres pero el ojo y la mente no se pueden ocupar de ambas formas a la vez y, por lo tanto, han de pasar rápida y continuamente de una a otra.” M. C. Escher.

Las imágenes mentales con que trabaja nuestro cerebro corresponden a modelos simplificados y promediados de la realidad. Como todo modelo, son representaciones incompletas del objeto observado, no obstante la información almacenada es significativa. Así, por ejemplo, somos capaces de distinguir un rostro conocido entre muchos otros de manera muy sencilla, pero nos enfrentamos con enormes dificultades para realizar una descripción detallada o un dibujo con base en la imagen memorizada. La percepción es un trabajo de sintetización y procesamiento de información que implica una combinación de elementos simples en un "todo". Percibimos casas, autos, árboles, etc. en vez de puntos, líneas texturas, colores etc. por separado.

En base a la experiencia acumulada el cerebro comienza a realizar suposiciones en la interpretación de la información que recibe a través de los sentidos. Esto se demuestra fácilmente por medio de las percepciones falsas o ilusiones. Ellas se producen al tomar ciertos parámetros de referencia para describir otras características. Por ejemplo, las dos rectas horizontales en la Ilustración 43 (A) son de igual magnitud pero se perciben distintas porque la característica longitud, esta definida por la forma de la figura. En la percepción visual juega un papel muy importante el proceso de rellenar, que es un típico proceso creativo. Como se observa en la Ilustración 43 (B), se ven contornos de triángulos que, en realidad, son imaginarios ya que son creados por el procesamiento cerebral.

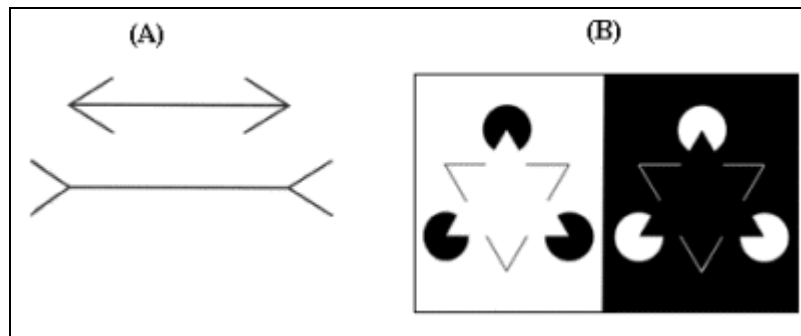


Ilustración 43 Ilusiones ópticas.

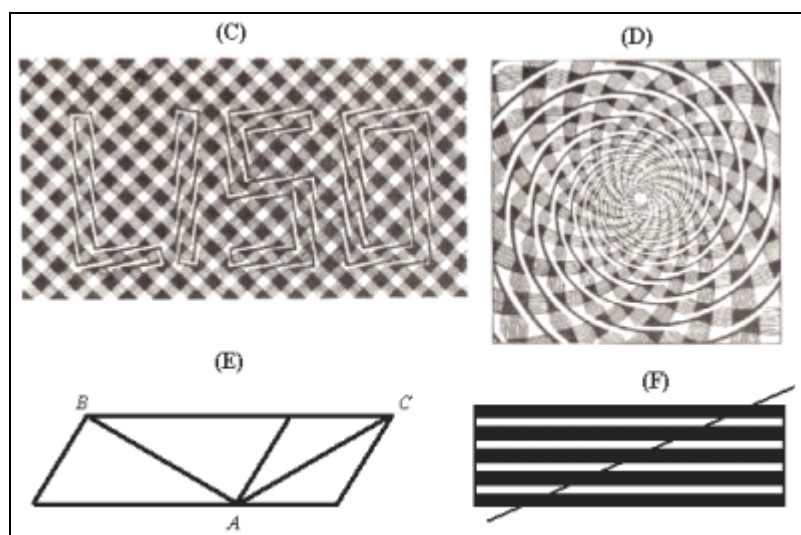


Ilustración 44 Más ilusiones ópticas. (C) Aunque no lo parezca las letras están derechas. (D) Si bien se aprecia un espiral, en realidad se trata de círculos concéntricos. (E) La distancia AB es igual a la distancia AC , aunque la primera parece mayor. (F) La recta oblicua que corta las franjas horizontales parece quebrada.

También el genio de Escher ha logrado engañar a nuestro cerebro con gráficos aparentemente correctos a primera vista. Una cuidadosa observación revela inmediatamente construcciones o situaciones imposibles de ser ciertas como se observan en la Ilustración 45

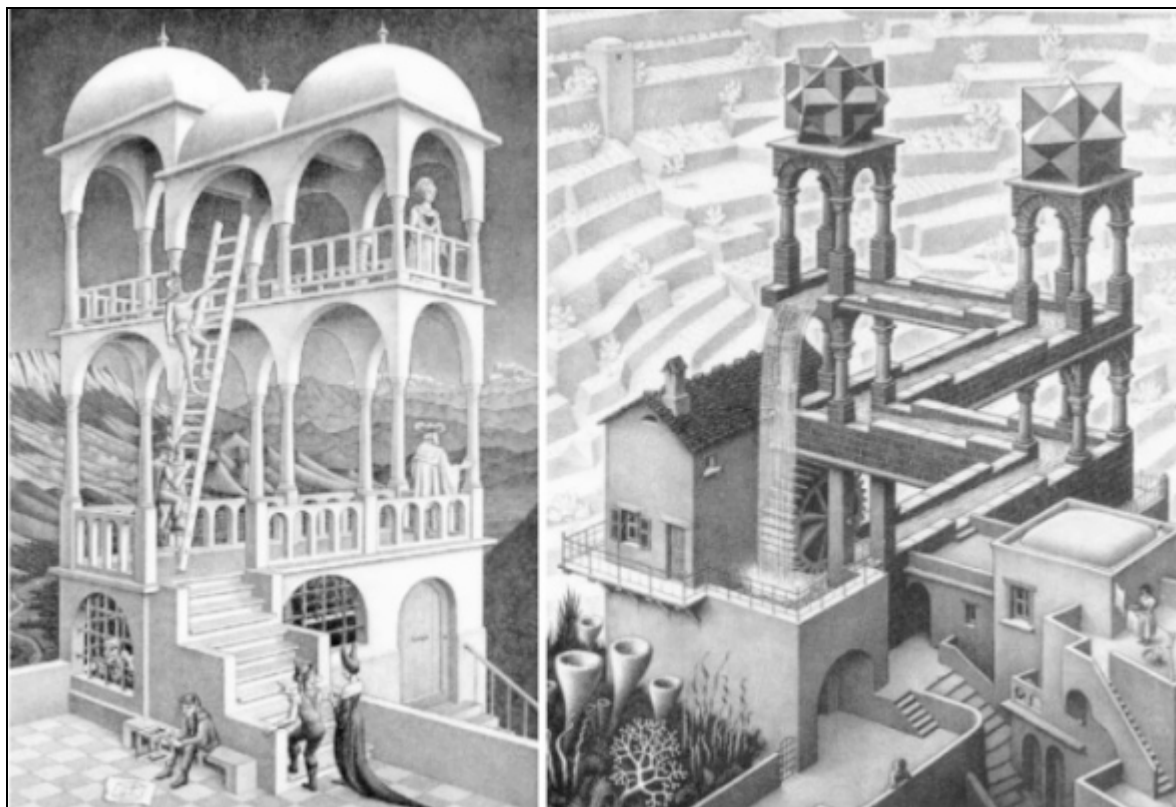


Ilustración 45 Gráfico de M. C. Escher. Observe el detalle de las columnas en las dos imágenes y el recorrido del agua en la imagen derecha.

Es indiscutible que el órgano de la percepción es el cerebro, quien se encarga de interpretar la información sensorial que recibe. Otro interesante ejemplo lo constituye la percepción visual de la profundidad. ¿Cómo sabemos, por ejemplo al mirar un paisaje, qué objetos están más cerca de nosotros y cuáles se encuentran más alejados? Nuevamente el cerebro se vale del contexto para crear la percepción. Detalles como la superposición de imágenes –el objeto que se encuentra delante oculta parcialmente uno por detrás– y el tamaño relativo entre los elementos del paisaje suele ser tomado en cuenta en el proceso de interpretación cerebral. Estos son detalles que todo pintor tiene presente cuando se dispone a realizar su trabajo y se relaciona con el concepto de perspectiva. No obstante existe otro importante detalle que completa la percepción de profundidad, a saber, la *estereoscopia* o visión doble.

El hecho de tener ambos ojos al frente con superposición de campos visuales posibilita que cuando observamos un objeto no muy distante nuestro cerebro procese en realidad dos imágenes diferentes –tomadas desde distintos ángulos por cada ojo– de manera simultánea. Así se perciben volúmenes y espacialidad. Para medir la distancia a un objeto, el cerebro toma las imágenes separadas de cada ojo y mide los pequeños desplazamientos en los objetos respecto a un punto de referencia. Esta referencia es el punto donde se cruzan las líneas de visión de ambos ojos.

La forma en que el cerebro interpreta una única imagen tridimensional a partir de dos ligeramente desfasadas ha sido aprovechada por las artes gráficas para producir efectos 3D desde el siglo XIX. Charles Wheatstone fue el primero, en 1832 comenzó tomando fotos con pares de vistas desfasadas, que se montaban en rudimentarios “estereoscopios”. Más tarde la

técnica se aplicó en la aerofotogrametría para la confección de mapas geográficos y Disney lo incorporó en juguetes infantiles –los *View Master*–.

Una técnica relativamente nueva y que cuenta con la ventaja de no necesitar ningún accesorio para percibir la tridimensión son las imágenes de puntos creada por el matemático húngro Bela Julesz quien publicó sus resultados en el *Bell Systems Technical Journal* No. 39, donde aparece un artículo acerca de percepción de profundidad binocular en patrones. A estos primeros estereogramas formados de puntos se les llama *Random Dots Stereograms* o *RDS* por sus siglas en inglés. Actualmente, la estereografía es una técnica por computadora cuyo resultado son imágenes en dos dimensiones que llevan codificada en su interior una figura con profundidad.

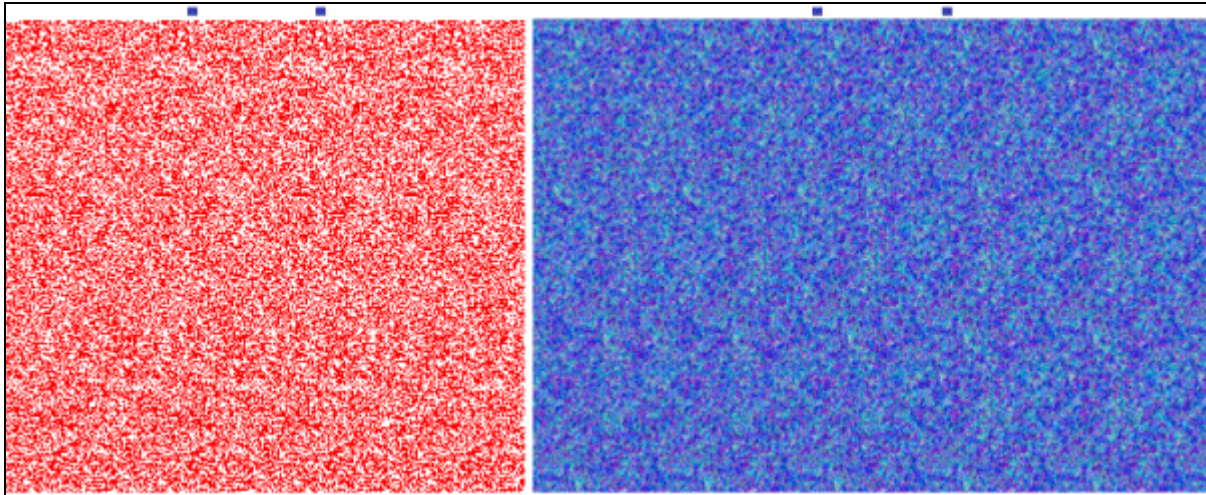


Ilustración 46 Estereogramas RDS –Random Dots Stereograms– Una rosquilla a la izquierda y un ave a la derecha.

Las imágenes estereoscópicas pueden mirarse sin necesidad de ningún aparato. Para esto no hay más que acostumbrarse a dirigir los ojos de una forma determinada. El resultado que se obtiene es el mismo que cuando se emplea el estereoscopio, aunque la imagen, en este caso, no se amplía. El inventor del estereoscopio, Wheatstone, utilizó primeramente este procedimiento natural. Se puede usar el método cruzado –ponerse bizco– o paralelo –hacer foco más allá del plano de la imagen–. Creo que el segundo es más fácil de dominar.

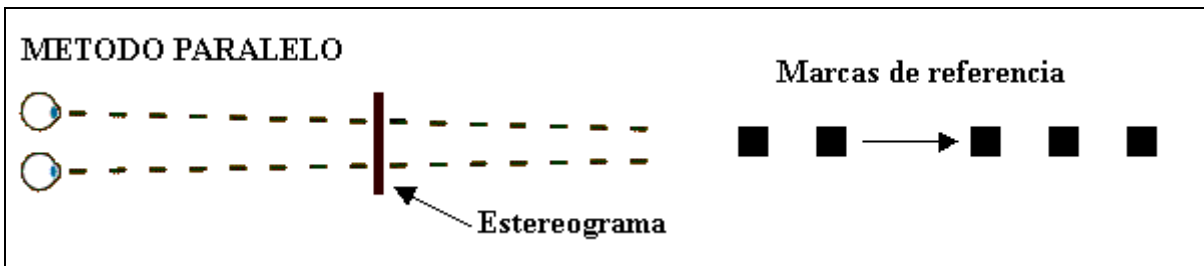


Ilustración 47 Método paralelo para visualizar estereogramas. Las marcas de referencia que suelen acompañar la imagen nos ayudan a encontrar el punto en el espacio que debemos enfocar. La idea es hacer foco a distintas profundidades más allá del plano del gráfico hasta lograr que las dos marcas se visualicen como tres. En ese momento comenzará a formarse la imagen tridimensional. Cuanto más la observemos, más detalles se harán evidentes, debido a que nuestros ojos terminan haciendo el ajuste fino del foco de manera automática.

Yakov Perelman, en su libro “Física recreativa”¹ nos advierte que no todas las personas consiguen ver estereoscópicamente, incluso con el estereoscopio; algunos –por ejemplo, los bisojos y los acostumbrados a trabajar con un solo ojo– son totalmente incapaces para ello; otros lo consiguen después de ejercitarse mucho, y finalmente, hay un tercer grupo, formado principalmente por jóvenes, que aprenden muy pronto, en sólo unos minutos. No obstante en el mismo artículo el autor expone un método para que las personas que hallan perdido un ojo puedan mirar fotografías estereoscópicas y recibir de ellas la impresión de relieve que directamente no puede percibir. El método consiste en proyectar sobre una pantalla, cambiándolas rápidamente, las fotografías destinadas al ojo derecho y al izquierdo. Así, las impresiones visuales, al cambiar rápidamente, se confunden en una sola imagen, de igual manera que ocurre cuando se ven a un mismo tiempo.

Los estereogramas RDS actuales han sustituido los puntos por patrones gráficos repetitivos que agregan gran colorido a la imagen final, sin estorbar la visión de la figura tridimensional. Es curioso que el cerebro pueda manejar la forma y profundidad de manera independiente al color y tono, lo que demuestra que son funciones cerebrales bien diferenciadas.

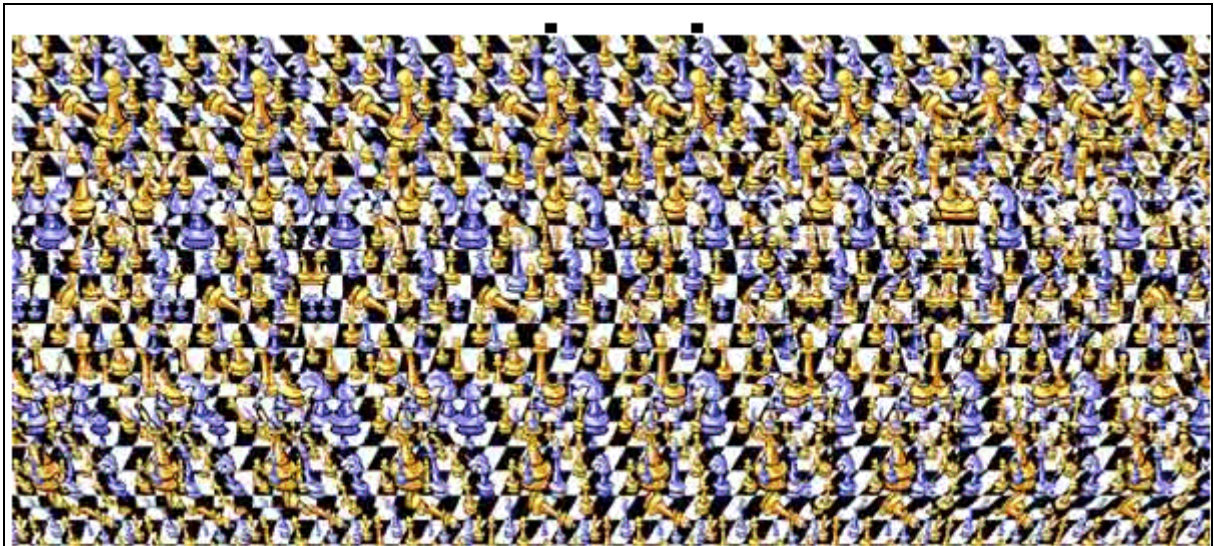
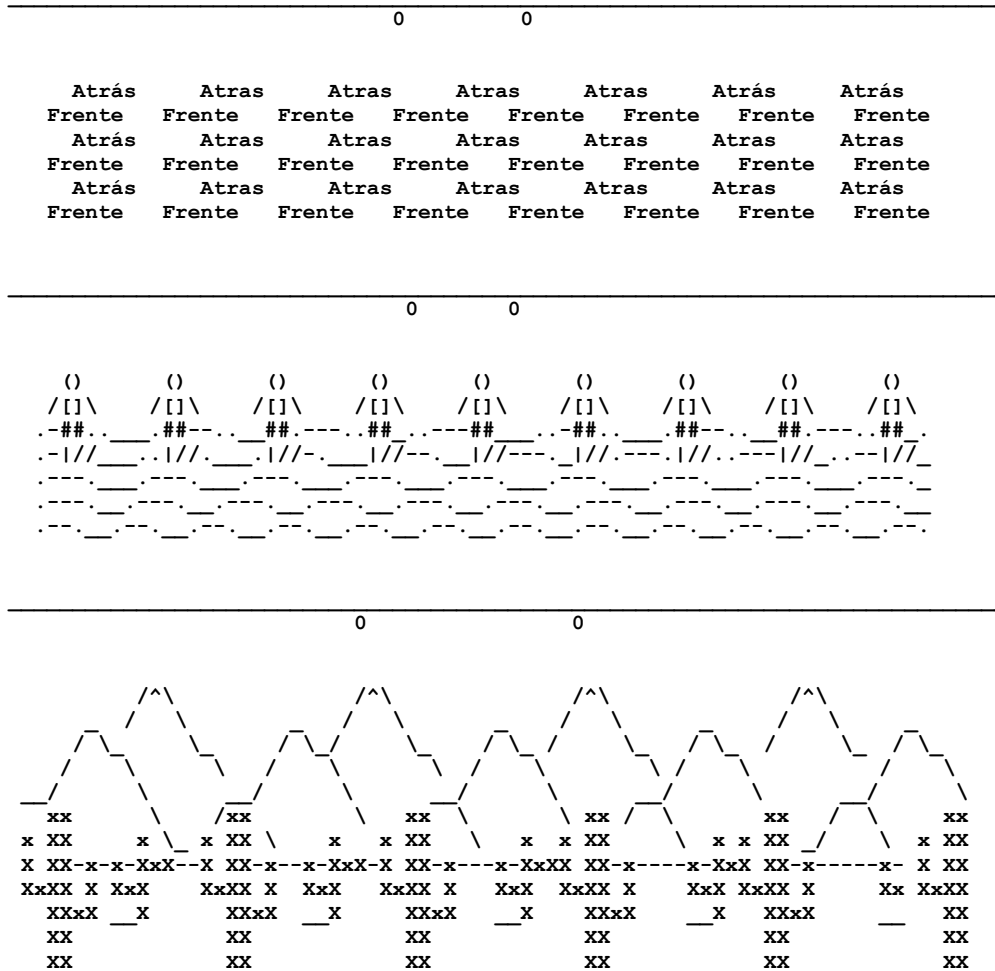


Ilustración 48 Estereograma editado a partir de imagen obtenidas de *Magic Eye How to See 3D* Copyright © 1995 by N.E. Thing Enterprises. A la que se le han agregado las marcas de referencia.

La utilización de esta técnica está prácticamente circunscrita al ámbito artístico debido sobre todo a la incapacidad de manejar otros dispositivos mientras se percibe la imagen tridimensional. Una relativamente nueva manifestación artística conocida como “Arte ASCII” –donde se crean imágenes a partir de la combinación adecuada de caracteres ASCII– también utiliza estereogramas para obtener percepciones tridimensionales. He aquí algunos ejemplos:

¹ Reproducción en línea en [Barros]



Otro sistema estereoscópico muy conocido, sobre todo por haber sido explotado por la industria del cine son las imágenes dobles con distintos tintes de color, denominados “*anaglifos*”. Las vistas derecha e izquierda se presentan superpuestas cargadas con distintos tonos. Al observarse con lentes que presentan ambos tonos –uno para cada ojo– se anula la imagen derecha en el ojo izquierdo y la imagen izquierda en el derecho. Así nuevamente nuestro cerebro es engañado produciéndose la percepción tridimensional a partir de una imagen plana.

Pero por ahora, el sistema más completo para percibir mundos virtuales en profundidad son los cascos o pantallas montadas en la cabeza que suelen referenciarse por sus siglas HMD: head-mounted displays. Dos pequeños monitores transmiten directamente la imagen izquierda y derecha a cada ojo y el usuario puede moverse libremente manteniendo las pantallas frente a sus ojos, esta tecnología posibilita el manejo de otros dispositivos –reales o virtuales– y es apta para aplicaciones tanto industriales como científicas.

La atención

Otra facultad del Hombre es su capacidad para "poner atención" a cierta parte de la información que se recibe por medio de sus sentidos, estando conscientes sólo de una pequeña parte de ésta. Poseemos un filtro que solamente permite el paso de cierta información relacionada con cambios abruptos del medio externo –un ruido súbito, un cambio de temperatura, etc.– pero que podemos desactivar en cualquier momento. El lector puede quizá en este punto concentrarse en los ruidos del ambiente y podrá oír seguramente algunos

sonidos de los que no estaba conciente cuando comenzó a leer este párrafo. Así será posible que con sólo pensarlo vuelva a percibir el peso del reloj que lleva en su muñeca, la presión de la silla en su cuerpo, o tomar conciencia de un incipiente dolor en la nuca producto de una mala postura imperceptible hasta este momento.

La concentración es una actividad mental y representa uno de los mecanismos fundamentales de la conciencia.

El sentido común

El Hombre, en su vida diaria, se enfrenta continuamente a la necesidad de tomar un sinnúmero de decisiones con base en una gran cantidad de información y opciones alternativas.

La mayor parte de las elecciones que el hombre enfrenta a lo largo de su vida las resuelve con base en su sentido común, el cual está formado por un número inmenso de experiencias acumuladas durante su vida. El sentido común es de naturaleza cualitativa, en el sentido de que raramente involucra relaciones numéricas entre conceptos; en cambio, ofrece perspectivas múltiples de un problema, a veces de manera simultánea. Conseguir máquinas que sean capaces de poseerlo, representa uno de los más grandes y distantes retos de la Inteligencia Artificial.

El lenguaje

El lenguaje, colección de símbolos, no es una variación de la vocalización animal. No descansa en conexión entre estímulo y respuesta como esta. Es una forma enteramente nueva de pensamiento individual y social iniciada seguramente por el *Homo habilis*, primera especie plenamente humana que comenzó a moverse hace unos 2 millones de años sobre las sabanas del África.

Pensemos en lo que significa manejar un lenguaje. Significa que desde la infancia, a pesar de que escuchamos un número limitado de frases, podemos producir y entender un número infinito de frases nunca antes habladas o escuchadas. Manejar un lenguaje significa poder identificar una palabra hablada, de entre un acervo de varios miles que tiene un adulto culto, en menos de 300 milisegundos. Significa poder armar frases en el mismo tiempo que se requiere para pronunciar las palabras. Todo ello supone el contar con un cerebro especializado en el manejo del lenguaje. [Diaz97]

No cabe duda de que el cerebro humano está especialmente dotado para la adquisición y manejo del lenguaje. Es así que los simios, nuestros parientes más cercanos sobre la Tierra, aunque pueden aprender palabras y expresarlas por signos del lenguaje manual de los sordomudos, no alcanzan, aun con el más dedicado entrenamiento, a manejar más lenguaje que el de un niño de dos años, lo cual no deja de ser sorprendente y significativo¹.

¹ Esto significa que un primate distinto de nosotros posee en principio la capacidad para manejar un sistema simple de símbolos. Este hallazgo tienen implicaciones muy importantes sobre la posibilidad de que el uso de símbolos por parte de un antecesor del género *Homo* haya precedido en el linaje humano al desarrollo del cerebro y contribuido a producirlo. Ya se ha mencionado en este capítulo que el pensamiento abstracto es exclusivo del Hombre. No obstante Carl Sagan dedica todo un capítulo en su libro "Los dragones del Edén" titulado "Las abstracciones de los brutos" en donde expone con lujo de detalle ciertas experiencias vividas con un grupo de monos a los que se les enseñó el lenguaje de señas de los sordomudos y que ponen en evidencia que estos animales son capaces de cierto grado de abstracción. Hoy día existen organizaciones que se plantean donde está la frontera que separa a los humanos de las especies de primates más próximas como los chimpancés, los gorilas o los orangutanes. La "Declaración de los Grandes Simios" pretende ampliar la comunidad moral de la igualdad más allá de la humanidad, de forma que no sólo abarque a nuestra especie, sino también a nuestro parientes más próximos y que

Contrariamente un niño puede adquirir cualquier lenguaje al que se le exponga sin ningún esfuerzo y sin enseñárselo a propósito.

¿Cómo es posible que los niños en tan poco tiempo –tres o cuatro años– puedan aprender tan bien el lenguaje materno, a pesar de las complejidades inherentes a los sistemas simbólicos y a la gran variedad de lenguas en las distintas poblaciones? Claudio Gutiérrez, filósofo e informático costarricense, se hace esta pregunta en un pequeño pero muy interesante artículo llamado “La coevolución del cerebro y el lenguaje dentro de una perspectiva humanista” [Gutierrez00_1]. En este artículo se alude a que la hipótesis de Noam Chomsky acerca de que todas las gramáticas son variaciones de una sola y genérica “gramática universal” con la que los cerebros humanos vendrían equipados por nacimiento, concebida hasta hace muy poco como una explicación válida al interrogante planteado, hoy es puesta en duda por muchos investigadores. Una explicación alternativa se enuncia en el título de este artículo, las lenguas se han ido adaptando durante siglos a las predisposiciones mentales infantiles y lo que hoy se tienen son “lenguas amigables para el niño”. Las lenguas humanas están todo el tiempo bajo la presión selectiva de tener que coincidir con las conjeturas más probables de los niños, pues si los niños no las aprenden las lenguas no existirán en la siguiente generación. Pero juntamente a la evolución del lenguaje debe haber evolucionado el cerebro humano, así toda mutación del cerebro que facilitara el uso de la palabra habrá ido siendo automáticamente seleccionada por la evolución de la especie humana misma.

Aparentemente la habilidad lingüística tiene estructuras y funciones que le son particulares y la diferencian de otras habilidades cognitivas. Se ha comprobado que la actividad neuronal que se manifiesta cuando un sujeto oye el lenguaje hablado es distinta a la que se produce cuando oye otro tipo de sonido.

La localización de las facultades del lenguaje en el cerebro no es un misterio, desde hace un siglo se sabe que el hemisferio cerebral dominante para la habilidad motriz –el izquierdo en los sujetos diestros– es también dominante para el lenguaje. Lo que no se sabe es cómo se ejecutan las habilidades lingüísticas –o de hecho ninguna de las facultades mentales superiores– en su sustrato nervioso.

Incluso se han observado zonas cerebrales que parecen estar especializadas en funciones particulares del lenguaje. Las evidencias más recientes indican que el procesamiento de los sustantivos y de los verbos ocurre en dos zonas distintas del cerebro.

Se debe mencionar también que el lenguaje cotidiano abarca no sólo habilidades lingüísticas sino que comprende también una importante porción llamada pragmática, que incluye las intenciones, actitudes y emociones que se expresan con entonación o gestos. A través de ellas es posible comprender el sarcasmo, el humor, el sentido figurado, la inferencia o la metáfora. Aparentemente el carácter pragmático del lenguaje es una habilidad del hemisferio cerebral no dominante.

Si bien existe una localización de funciones cognitivas fundamentales, las propiedades cognitivas superiores son producto de la interacción de esos sistemas.

más se parecen a nosotros en sus facultades y habilidades mentales, inteligencia y autoconsciencia así como en la complejidad de sus sistemas sociales o de comunicación. Puede encontrarse información al respecto en el sitio web de “Fundación Utopía Verde” <http://www.utopiaverde.org/sos/simios/>

La memoria y el aprendizaje

Aprendizaje puede ser definido como el proceso de adquisición de conocimientos acerca del mundo, en tanto que memoria sería mejor definida como los mecanismos que aseguran la retención, el almacenamiento y el recobro de tales conocimientos. [Cárdenas]

Concebido como órgano mental, el cerebro percibe, memoriza, decide y actúa por medio de la conducta. Podríamos preguntarnos ¿cómo se codifican y en donde están los recuerdos? ¿de qué manera se organiza la conducta en el cerebro? Debe existir una huella, alguna forma en la que la experiencia deje su marca en el tejido nervioso. A esa huella o templete se le ha llamado *engrama*, pero nadie sabe exactamente en qué consiste. [Diaz97]

La imagen del cerebro que tenemos a partir de muchos experimentos recientes es muy diferente de la de antaño. Se trata de un órgano con movilidad anatómica y celular. Este dinamismo cerebral está estrechamente relacionado con nuestras capacidades cognitivas y se ha acordado en llamar *plasticidad neuronal*.

Plasticidad neuronal

Se han establecido algunas correlaciones entre cambios anatómicos, eléctricos y químicos del cerebro, específicos en lo que se refiere a tiempo y espacio, con los cambios cognitivos y conductuales.

La llamada plasticidad cerebral se muestra coherente con la naturaleza cambiante de la mente y el comportamiento. El término plasticidad sugiere apropiadamente movimiento, procesos activos y reactivos de un material físico, en este caso, del órgano más complejo y evolucionado que conocemos. De acuerdo con su naturaleza biológica, el cerebro se comporta como la materia viva que es: cambia su estructura y sus funciones según la edad, el aprendizaje, la patología, el uso. La plasticidad se refiere no sólo a los cambios celulares del órgano sino a la producción, modificación o recuperación de la conducta o la cognición perdidas.

La plasticidad neuronal es la propiedad que tienen las células nerviosas en respuesta a estímulos internos o externos de reorganizar sus conexiones sinápticas y de modificar los mecanismos bioquímicos y fisiológicos implicados en su comunicación con otras células.

La plasticidad neuronal es sin duda el acontecimiento más importante que ocurre dentro del sistema nervioso. Si bien la capacidad de reorganizarse estructuralmente no es exclusiva del sistema nervioso –la angiogénesis en el sistema circulatorio es un hito similar–, en éste toma un cariz especial. Sobre este fenómeno residen todos los fenómenos psicológicos que acompañan a la formación de nuevas conductas, así como la extinción de otras viejas. El aprendizaje, la memoria, el cambio de actitudes, así como la permanencia de los conocimientos y habilidades, dependen en última medida de esta peculiar característica del sistema nervioso, de la formación de nuevas sinapsis en reacción a los estímulos.

Durante años los científicos han creído que el desarrollo cerebral finalizaba después de un "periodo crítico". Las investigaciones más recientes señalan que el cerebro está continuamente reorganizándose de un modo dinámico, también en los adultos.

En condiciones normales una vez formadas las células del tejido nervioso, estas no suelen regenerarse. Pero cuando una neurona pierde sus ramificaciones es capaz de poner en marcha una serie de mecanismos compensatorios mediante el cual mantiene su densidad sináptica. Por otro lado, la regeneración axónica dentro del Sistema nervioso periférico es un hecho ya totalmente demostrado.

Como ejemplo se puede citar lo que sucede cuando se sufre una amputación traumática. Al cambiar el input neuronal, el cerebro se reorganiza en cuanto a la nueva situación estimular. En los 80 se hicieron experimentos con monos, en este sentido. Cuando los impulsos nerviosos de uno de los dedos de un mono eran bloqueados la zona encefálica encargada de procesar la información correspondiente a ese dedo, tras unos meses, empieza a responder a las señales de estimulación de otros dedos de al lado –que al principio no le correspondían–. Este fenómeno explicaría "los miembros fantasma", en el que pacientes sufren de dolores o sensaciones en aquellas zonas que han sido objeto de amputación. Todas las áreas cerebrales están continuamente adaptándose a las situaciones estimulares. El remodelado sináptico está cambiando continuamente en el cerebro.

Aprendizaje

Ya he mencionado en los primeros párrafos del apartado “Arquitectura neuronal” –pág. 99– que casi todos los organismos actúan en buena medida conforme al legado genético de que son portadores, siendo la información extragenética recogida en el curso de su vida un factor secundario. Una abeja que, atraída por el aroma, se aproxima a una flor o un salmón luchando contra la corriente en una catarata están haciendo uso de los llamados comportamientos específicos de especie que codificados en sus genes han sido previamente transmitidos al sistema nervioso del individuo.

Ya se ha comentado también sobre cómo a medida que aumenta la complejidad de los organismos, el bagaje de información extragenética empieza a incrementarse hasta que en el caso del Hombre sobrepasa en gran medida al caudal informativo impreso en sus genes. De hecho, puede afirmarse que a medida que aumenta la complejidad de los sistemas nerviosos – y con ella las posibilidades perceptuales–, procesos de autodireccionamiento van alcanzando mayor relevancia en la producción y regulación del comportamiento, desvaneciendo el papel jugado por los comportamientos específicos de especie.

Se han identificado dos tipos fundamentales de aprendizaje. El primero de ellos, denominado aprendizaje no asociativo, se da luego de la exposición repetida a un estímulo. Es este tipo de aprendizaje el que permite que nos acostumbremos a, por ejemplo, el sonido de los autos que pasan por una avenida próxima a nuestra casa, o al tic-tac nocturno de un reloj, gracias a un proceso llamado *habituación*. También son ejemplos de este tipo de aprendizaje la *deshabituación* –recuperación de la respuesta habitual producida por el efecto de otro estímulo– y la *sensibilización*. –aumento de la respuesta frente a un estímulo repetitivo–.

Otro tipo de aprendizaje más complejo es el asociativo, donde se aprende sobre las relaciones entre dos o más estímulos, o entre ellos y nuestro propio comportamiento y es el responsable de eventos tales como la salivación ante el solo pensamiento de morder un limón.

El *condicionamiento clásico* o *Pavloviano* es una forma de aprendizaje asociativo, es decir, una forma de conducta en que se asocian estímulos y respuestas. El clásico ejemplo es el experimento de Pavlov en que un perro aumenta su secreción de saliva –respuesta– ante la visión del alimento. Si se repite el experimento, pero asociando el sonido de un timbre o el efecto de una luz a la presencia de ese estímulo, se encontrará que al cabo de un cierto número de repeticiones de estas asociaciones, bastará la presencia del timbre o de la luz para provocar el aumento de la secreción salival en el perro. Volveré sobre este tema unos párrafos más adelante.

Otro esquema de clasificación de aprendizaje se basa en el tipo de conocimiento adquirido. Pensemos en la diferencia existente entre, por un lado, memorizar eventos como fechas o definiciones del diccionario y, por otro, acciones como jugar al tenis o conducir una bicicleta.

En el primer caso, se está haciendo uso de la llamada *memoria declarativa*, a través de la cual recordamos hechos o eventos relevantes que pueden ser comunicados a otros. En el segundo caso se está utilizando otra forma de memoria denominada *memoria procedimental*, mediante la cual aprendemos procedimientos o habilidades motrices. La memoria declarativa, quizá por la mediación lingüística, se forma relativamente con mayor rapidez, en tanto que la memoria procedimental requiere de repetición y práctica para su formación. De la misma forma, la memoria declarativa parece ser más susceptible del olvido mientras que la procedimental se muestra más resistente.¹

Además de aquellas dos formas de clasificar los tipos de memoria, ha sido propuesta una tercera, de acuerdo con el curso temporal del recuerdo. El tipo de memoria que almacena informaciones por periodos cortos de tiempo es llamada memoria a corto plazo y aquel que permite el almacenamiento por periodos mayores de tiempo se denomina memoria a largo plazo. La memoria a largo plazo posee una mayor capacidad que la memoria a corto plazo.

Al comienzo de este apartado –“La memoria y el aprendizaje”– he comentado sobre la inquietud de los investigadores sobre los mecanismos cerebrales subyacentes a la memoria y la búsqueda del engrama de un recuerdo o aprendizaje. Esta inquietud llevó a algunos de ellos a proponer hipótesis de muy difícil confirmación experimental –como la memoria holográfica–. También condujo a otros a descomponer tan detalladamente el proceso, que resulta difícil encontrar lógica para su reconstrucción. De una forma o de otra, las grandes corrientes de pensamiento en neurobiología de la memoria han y siguen estando inspiradas en las ideas de Donald Hebb. En pocas palabras, Hebb propuso que el aprendizaje sólo sería posible si existiese una simultaneidad temporal del funcionamiento de dos neuronas conectadas entre sí, una presináptica y otra postsináptica. Ese funcionamiento simultáneo, permite la conformación de circuitos neuronales y sería el factor *sine qua non* para la formación del recuerdo. Desafortunadamente Hebb carecía de las herramientas tecnológicas que le permitiesen poner a prueba sus ideas.

Pero antes de la teoría de Hebb –la cuál procede de un libro escrito en 1949 llamado “Organization of Behaviour”– se debe mencionar como el primer intento significativo para explicar las capacidades del cerebro al trabajo de McCulloch y Pitts en 1943. La teoría de estos autores se basa en que la actividad de una neurona es un proceso “todo o nada”, es decir que las neuronas son binarias o bien están activadas o bien están desactivadas. Así es posible utilizar la lógica de proposiciones para describir la acción de ciertas redes neuronales. Es más, cualquier red que no tenga conexiones de realimentación se puede describir en término de combinaciones de unas cuatro expresiones sencillas. Aunque la teoría de McCulloch-Pitts ha resultado no ser un modelo preciso de la actividad cerebral, ha servido de inspiración para los investigadores de las ciencias de la computación en la actualidad. [Freeman93]

Pero volviendo al aprendizaje de Hebb, veamos su idea principal que se expresaba en forma de suposición:

Cuando un axón de la célula A está suficientemente próximo para excitar a una célula B o toma parte en su disparo de forma persistente, tiene lugar algún proceso de crecimiento o algún cambio metabólico en una de las células, o en las dos, de tal modo que la eficiencia de A, como una de las células que desencadena el disparo de B se ve incrementada.

¹ Esto da sentido a la expresión coloquial de “es como andar en bicicleta, una vez aprendido no se olvida nunca”.

Volvamos al experimento de Pavlov para ilustrar el concepto.

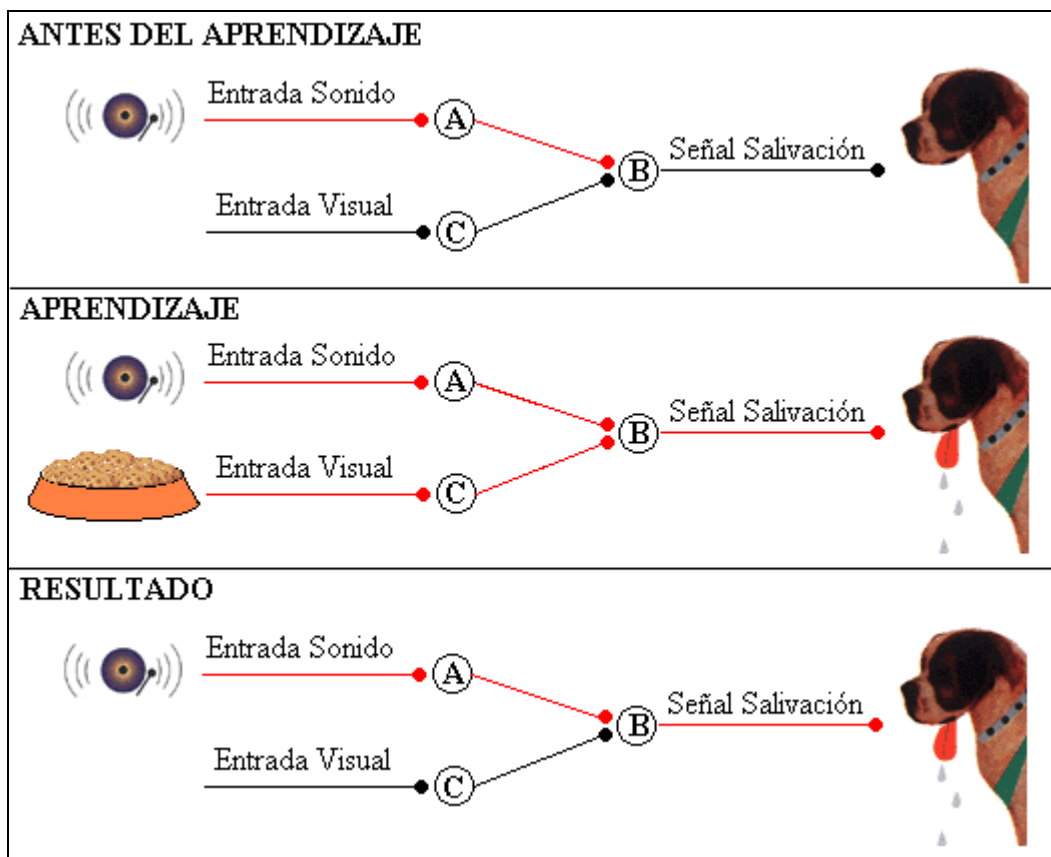


Ilustración 49 Aprendizaje de Hebb. El disparo de las neuronas ha sido graficado en color rojo. El color negro indica inactividad.

Como se observa en la ilustración precedente, previamente al aprendizaje, la excitación de la neurona A ante el estímulo sonoro no es suficiente para excitar a la neurona B que provoca la salivación, es decir que A es poco eficiente para desencadenar el disparo de B. Esto no ocurre con la neurona C, cuya excitación producto del estímulo visual, alcanza para excitar B y provocar la salivación.

Pero si en la etapa de aprendizaje estimulamos A junto a C mostrando comida al sujeto y al mismo tiempo haciendo sonar una campanilla estaremos logrando la excitación conjunta de A y de B –desencadenada por C–. Por lo tanto durante el aprendizaje que se repite un número adecuado de veces, A participa de la estimulación de B aunque por sí sola no sería suficiente para dar lugar al disparo de B. En esta situación la suposición de Hebb determina que se produce algún cambio entre A y B, de tal modo que la influencia de A para con B se ve incrementada hasta tal punto que A será capaz de lograr que se dispare B, incluso en ausencia de la estimulación visual procedente de C. Entonces será posible obtener la salivación del sujeto con sólo hacer sonar la campanilla.

Dado que la conexión entre neuronas se hace a través de las sinapsis es razonable suponer que tales cambios deben ocurrir en las sinapsis. Hebb suponía que aumentaba el área de la unión

¹ Citado por James Freeman y David Skapura en su libro “Redes neuronales Algoritmos, aplicaciones y técnicas de programación”. [Freeman93]

sináptica. Hoy se sabe que con el aprendizaje aumenta en el cerebro la síntesis de proteínas; se activan y se favorecen nuevas rutas de comunicación entre ciertas neuronas; se hacen circuitos de retroalimentación.

Fue a partir del descubrimiento del proceso de potenciación a largo plazo por Bliss y Lomo – 1973– que vinieron a ser confirmadas empíricamente las hipótesis de Hebb. Brevemente, la potenciación a largo término consiste en el fenómeno mediante el cual la excitación de una neurona presináptica –que normalmente ocasiona una respuesta de una intensidad determinada en una neurona postsináptica–, es capaz de ocasionar una respuesta de intensidad mayor como consecuencia de la previa excitación repetitiva. Actualmente se conocen muy detalladamente los mecanismos neuroquímicos subyacentes a este fenómeno. La potenciación a largo plazo descansa sobre el trabajo de un tipo de receptores presentes en la superficie de las membranas externas de las neuronas postsinápticas, denominados N-Metil-D-Aspartato o simplemente NMDA, a tal grado, que existen algunos autores que hablan de la potenciación dependiente de NMDA. [Cárdenas]

Son también de suma importancia los trabajos de Tang y colaboradores, en el ámbito de la neurobiología de la memoria. Sus experimentos consistieron en la manipulación genética de neuroblastos para incrementar la expresión de una parte del receptor de NMDA, responsable, como ya mencionamos, del proceso de potenciación a largo plazo. Los ratones manipulados genéticamente, aumentaron la cantidad de receptores de NMDA en sus cerebros. Comportamentalmente, estos ratones manifestaron un marcado incremento de la memoria. Las implicaciones de estos hallazgos sobre la prevención y el tratamiento de las disfunciones de la memoria, son enormes, así como son enormes también las posibilidades de modificación deliberada de embriones, razón por la cual ya están conformándose comisiones de bioética, encargadas de estudiar la futura dirección de los descubrimientos.

Hasta aquí el enfoque desde la neurobiología. En el capítulo siguiente entraremos directamente al tratamiento de las Redes Neuronales Artificiales, un intento de lograr “inteligencia artificial” emulando lo que se conoce acerca de la forma que tiene el cerebro para conseguir esta inquietante facultad que tanto interés a suscitado en todas las épocas: la inteligencia humana.

Redes Neuronales Artificiales y Neuroevolución

Introducción

Resulta irónico pensar que las computadoras actuales sean capaces de resolver complejos cálculos matemáticos a velocidades increíbles, y al mismo tiempo muestren tantas dificultades en la realización de tareas sencillas para los seres humanos, como identificar un rostro conocido entre muchos otros; lo que resulta relativamente fácil de hacer hasta para un niño de sólo dos años de edad. Las tramas complejas, formadas por numerosos elementos visuales que individualmente dicen poco acerca de la trama global y, sin embargo, colectivamente, representan objetos fácilmente reconocibles para un ser humano, son los tipos de tramas que resultan ser más difícilmente reconocibles para una computadora.

Evidentemente, los sistemas de cómputo que trabajan bajo la filosofía de los sistemas secuenciales, desarrollados por Von Neuman, muestran gran incapacidad para interpretar el mundo de la manera en que lo hacemos nosotros. Nuestras capacidades cerebrales superiores como la percepción –recordemos que “percibir” es distinto a “sentir” pues involucra procesamiento cerebral de interpretación de la información externa, ver “La percepción” pág. 109– son las que todavía nos distinguen como triunfadores absolutos en la competencia hombre-máquina.

Estas dificultades y limitaciones han hecho que un gran número de investigadores centre su atención en el desarrollo de nuevos sistemas de tratamiento de la información, que permitan solucionar problemas cotidianos, tal como lo hace el cerebro humano. Las características de este órgano biológico deseables para cualquier sistema de procesamiento digital, entre otras, son:

- Es robusto y tolerante a fallas, diariamente mueren neuronas sin afectar su desempeño.
- Es flexible, se ajusta a nuevos ambientes por aprendizaje, no hay que programarlo.
- Puede manejar información difusa, con ruido o inconsistente.
- Es altamente paralelo.

El cerebro humano constituye una computadora muy notable, es capaz de interpretar información imprecisa suministrada por los sentidos a un ritmo increíblemente veloz. Lo más impresionante de todo, es que el cerebro aprende sin instrucciones explícitas de ninguna clase, a crear las representaciones internas que hacen posibles estas habilidades.

Basados en la eficiencia de los procesos llevados a cabo por el cerebro, e inspirados en su funcionamiento, varios investigadores han desarrollado desde hace más de 40 años la teoría de las Redes Neuronales Artificiales –RNA–, las cuales emulan las redes neuronales biológicas, y se han utilizado para aprender estrategias de solución basadas en ejemplos de comportamiento típico de patrones; estos sistemas no requieren que la tarea a ejecutar se programe, ellos generalizan y aprenden de la experiencia.

Cuando la investigación sobre redes neuronales parecía prometedora, en 1969 Minsky y Papert publicaron un trabajo que significó el abandono masivo de las prácticas investigativas en el área, sobre todo en los Estados Unidos, ya que en Europa su influencia fue menor. La crítica del padre de la Inteligencia Artificial estaba centrada en la limitación del *Perceptrón* – primera red neuronal con capacidad de aprendizaje– para aprender a resolver una operación lógica tan básica como el Or-exclusivo o XOR. Durante un período de 15 años, las

instituciones no se atrevieron a financiar ningún trabajo que girara alrededor de las RNA, ya que parecía un camino sin salida.

Sin embargo, a principios de los 80, esta disciplina resurgió de sus cenizas. Apareció el exitoso algoritmo de *Backpropagation* –en realidad fue redescubierto–, y con él un conjunto de nuevas arquitecturas y algoritmos de entrenamiento basados en diversas ideas. En este nuevo período de esplendor de la neurocomputación, se ha encontrado en otra disciplina de la Inteligencia Artificial Subsimbólica, una de las mejores formas de conseguir redes neuronales de cualquier grado de complejidad: el uso de algoritmos evolucionarios. Esta combinación de paradigmas suele denominarse *Neuroevolución*.

El presente capítulo tiene por objetivo presentar los conceptos básicos de la Neuroevolución, paradigma en que se encuentra enmarcado este trabajo. Para ello se requiere una introducción general a las redes neuronales artificiales que reforzaré con algunos ejemplos puntuales para clarificar ciertos conceptos. Ya se ha mencionado que hoy día existen muchos desarrollos en el área que han producido una amplia gama de arquitecturas de redes neuronales distintas, cada una con sus propias características, algoritmos de aprendizajes y capacidades específicas para solucionar cierta clase de problemas. Se han seleccionado tres arquitecturas para exponer con algún grado de detalle en el presente trabajo: El *Perceptrón*, el *Adaline* y la *Backpropagation*. Las razones responden a la importancia histórica, al éxito de la *Backpropagation* para resolver problemas práctico, y a que los tres modelos constituyen una buena muestra que representa la filosofía y desenvolvimiento general de las RNA.

El capítulo está organizado de la siguiente manera. A continuación se presentará la descripción de algunas de las características principales de este paradigma enfatizando el contraste existente con los sistemas de cómputo tradicionales. Esta caracterización se completará una vez abordados con algún detalle los temas concernientes a la *neurona artificial* –primer modelo de procesamiento neuronal, pero carente de aprendizaje–, el *Perceptrón* –primera red neuronal con capacidad de aprendizaje–, el *Adaline* –que introduce una importante regla de aprendizaje más potente que la del Perceptrón– y la *Backpropagation* –que generaliza la regla de aprendizaje del Adaline para utilizarla en redes multicapas que no sólo resuelven el problema del XOR, sino muchos otros de gran complejidad–. Se verá la derivación de cada una de las reglas de aprendizaje de estas arquitecturas, y también las posibilidades y debilidades de las mismas.

Retomaré la perspectiva general dedicando algunas líneas a las arquitecturas neuronales clasificándolas según distintos criterios. Una descripción de los distintos tipos de aprendizajes existentes antecede la enumeración de los modelos de RNA más relevantes junto a sus características principales. Finalmente se citarán algunas aplicaciones de las RNA para pasar a tratar el tema de la Neuroevolución, que surge como respuesta a algunas debilidades del paradigma de las RNA. Además de las motivaciones que llevaron a un grupo de investigadores a desarrollar esta combinación de paradigmas –RNA y Computación Evolutiva– se presentarán distintas estrategias de evolución: de pesos de conexión, de arquitectura, de función de transferencia, de arquitectura y pesos simultáneamente y de reglas de aprendizaje.

Es necesario aclarar que existen muchos artículos presentados por varios investigadores que proponen distintas estrategias neuroevolutivas, cada una con sus capacidades especiales que superan el promedio de otras en algún aspecto determinado. Existen algunas importantes en el contexto del presente trabajo pero no serán expuestas en este capítulo sino en “Estrategias de Neuroevolución implementadas” –pág. 194– donde no sólo se comentan genéricamente sino que también se exponen los detalles particulares de la implementación aquí realizada.

Sin más preámbulo comenzaré con el tratamiento del paradigma conexionista de la Inteligencia Artificial en el próximo apartado.

Redes Neuronales Artificiales –RNA–

A diferencia de los sistemas de cómputo tradicionales, las RNA no ejecutan instrucciones sino que responden en paralelo a las entradas que se les presenta. El resultado no se almacena en una posición de memoria, este es el estado de la red para el cual se logra equilibrio. El conocimiento de una red neuronal no se almacena en instrucciones, sino que el poder de la red está en su topología y en los valores de las conexiones –pesos– entre neuronas.

A las RNA no se le indican las reglas para dar una solución a través de un programa, sino que extrae sus propias reglas a partir de los ejemplos de aprendizaje, modifican su comportamiento en función de la experiencia. Esas reglas quedan almacenadas en las conexiones y no representadas explícitamente como en los sistemas basados en conocimiento –simbólico-deductivos–.

Una vez entrenada, se le pueden presentar a la red datos distintos a los usados durante el aprendizaje. La respuesta obtenida dependerá del parecido de los datos con los ejemplos de entrenamiento. Esto representa la importantísima capacidad de *generalización*.

Las redes neuronales artificiales son capaces de extraer o abstraer las características esenciales de las entradas aprendidas, de esta manera pueden procesar correctamente datos incompletos o distorsionados. Esto se conoce con el nombre de *tolerancia a ruidos*.

Las neuronas reales trabajan en paralelo; en el caso de las redes artificiales es obvio que si usamos un solo procesador no podrá haber proceso paralelo real; sin embargo hay un paralelismo inherente, lo esencial es que la estructura y modo de operación de las redes neuronales las hace especialmente adecuadas para el procesamiento paralelo real mediante multiprocesadores.

El conocimiento acumulado por la red se halla distribuido en numerosas conexiones, esto tiene como consecuencia la *tolerancia a fallos*: una red neuronal es capaz de seguir funcionando adecuadamente a pesar de sufrir lesiones con destrucción de neuronas o sus conexiones, ya que la información se halla distribuida por toda la red, sin embargo en un programa tradicional un pequeño fallo en cualquier punto puede invalidarlo todo y dar un resultado absurdo o no dar ningún resultado.

Esta austera presentación de las RNA y sus características será completada luego. Considero necesario ahora adentrarnos en algunos temas puntuales, para después volver a retomar la perspectiva más general. Por lo tanto, se abordará inmediatamente el estudio de la *neurona artificial* –bloque constructivo principal de las RNA– y se presentarán alguna de las redes más relevantes, por sus implicaciones históricas –Perceptrón y Adaline– o por el éxito en aplicaciones concretas –Backpropagation–.

Neurona artificial

En el capítulo anterior vimos que el cuerpo de la neurona –soma– asume un rol importante al procesar la información que llega a la célula a través de las sinapsis dendríticas. Recordemos que el trabajo del soma es integrar esta información inhibidora o excitadora a cada instante para obtener un resultado final que se traducirá en un impulso eléctrico que se propagará por el axón si existe la suficiente carga excitatoria. También se ha observado el hecho que no todas las conexiones poseen la misma importancia, siendo algunas sinapsis mucho más eficientes que otras en el proceso de la comunicación.

A partir de estas observaciones biológicas, Warren McCulloch y Walter Pitts en 1943 presentaron el primer modelo de una *neurona artificial*. Según Freeman [Freeman93], estos elementos individuales de cálculo que forman la mayoría de los modelos de RNA no suelen denominarse neuronas artificiales, lo más frecuente es darles el nombre de nodos, unidades o elemento de procesamiento –PE por sus siglas en inglés–. En el presente trabajo se utilizarán estos términos de manera intercambiable.

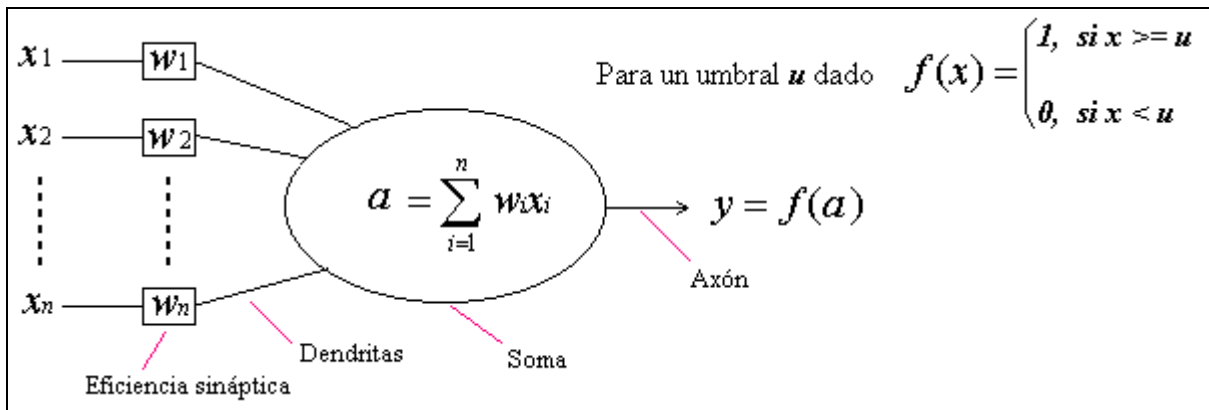


Ilustración 50 Modelo de neurona artificial de McCulloch y Pitts

La neurona artificial posee un número finito de conexiones de entrada – n en este caso– que se corresponden con las dendritas en el modelo biológico, por las cuales ingresa cada una de las n componentes de un vector de entrada \bar{x} . Toda conexión de entrada tiene asociada una magnitud llamada peso o intensidad –eficiencia sináptica– definida por las componentes del vector de pesos \bar{w} . Estos pesos w_i pueden ser positivos –excitatorios– o negativos –inhibitorios–.

La neurona acumula todas las señales de entradas multiplicadas por sus pesos de conexión obteniendo el valor de activación – a en la ilustración– al que se aplica una función llamada de *transferencia*¹, en este caso la función umbral para obtener la salida del PE.

Al igual que en una neurona biológica, tenemos muchas entradas pero una única salida, la cual podrá no obstante, conectarse con muchos otros PEs. En este modelo, una neurona es un elemento binario cuyo estado se manifiesta por la salida que pertenece al conjunto $\{0,1\}$. Así, los autores han utilizado la lógica proposicional para describir la acción de ciertas redes formadas por la conexión de estas neuronas.

El modelo McCulloch-Pitts intenta emular el procesamiento neuronal, pero la teoría presentada por estos investigadores no dice nada de como se organiza el dispositivo y/o como se ajustan los pesos y umbrales de los PEs. En otras palabras no se incluye proceso de

¹ Debo destacar que desafortunadamente no existe una terminología unificada respecto al nombre de esta función. Suele denominarse también función de activación. Freeman en [Freeman93] hace la siguiente distinción: a la sumatoria de la entrada ponderada por los pesos sinápticos la denomina *entrada neta*, una vez calculada se transforma en el valor de activación aplicando alguna función $a(t)=F(a(t-1),neta(t))$ para denotar que la activación es una función explícita de la entrada neta y puede también depender del valor anterior de activación. El autor acota que en la mayoría de los casos la activación y la entrada neta son idénticas y los términos suelen emplearse de manera intercambiable. Una vez calculada la activación se obtiene el valor de salida aplicándole la *función de transferencia*. Siguiendo a este autor, en el presente trabajo siempre se utilizará el término *función de transferencia* o *función de salida* para denotar a aquella función con la que se obtiene la salida del PE. Téngase en cuenta, sin embargo, que la literatura no siempre sigue de manera estricta esta convención.

aprendizaje alguno, no obstante su investigación ha sido significativa pues ha inspirado a un conjunto de investigadores del área que lo sucedieron.

Perceptrón

El primero al que se le ocurrió una aplicación práctica para la neurona de McCulloch y Pitts, y aplicando las ideas de aprendizaje de Hebb, fue el psicólogo Frank Rosenblatt que diseñó la red tipo Perceptrón en el año 1957. Rosenblatt creía que la conectividad existente en las redes biológicas tiene un elevado porcentaje de aleatoriedad, por lo que se oponía al análisis de McCulloch-Pitts en el cual se empleaba la lógica simbólica para analizar estructuras bastante idealizadas. Este psicólogo suponía que la herramienta de análisis más apropiada era la teoría de probabilidades.

El primer modelo de Perceptrón fue desarrollado imitando el funcionamiento del ojo humano, el fotoperceptrón como se le llamó era un dispositivo que respondía a señales ópticas, y era capaz de clasificar patrones correctamente, en lo que Rosenblatt denominaba un entorno diferenciado, en el cual cada clase estaba formada por patrones similares.

El Perceptrón era inicialmente un dispositivo de aprendizaje, y esta ha sido su mayor contribución, –recordemos que en el trabajo de McCulloch y Pitts no había aprendizaje–. El Perceptrón debía entrenarse hasta adquirir la capacidad de distinguir patrones. Este dispositivo trabaja con patrones de entrada binarios, y su funcionamiento se realiza en dos fases: una primera en la que se presentan las entradas y las salidas deseadas; en esta fase la RNA aprende la salida que debe dar para cada entrada, en la fase siguiente, de operación, la RNA "es capaz" de responder adecuadamente cuando se le vuelven a presentar los patrones de entrada. En esencia, el entrenamiento implicaba un proceso de refuerzo mediante el cual los pesos de las conexiones se incrementaba o se decrementaba dependiendo de si las mismas contribuían o no a las respuestas correctas del Perceptrón para una entrada dada. Este tipo de entrenamiento, en el que se debe conocer la respuesta correcta, se denomina supervisado.

El Perceptrón es una red neuronal que consiste en uno o más elementos de procesamiento – que son también referidos individualmente como perceptrones–. Por simplicidad ahora nos concentraremos en la operación de un Perceptrón simple, es decir un elemento de procesamiento de esta red, luego veremos como se utilizan varios en una red unicapa. El Perceptrón es aún hoy una red de gran importancia, pues con base en su estructura se han desarrollado otros modelos de red neuronal como la red Adaline y las redes multicapa.

Un Perceptrón simple tiene capacidad para distinguir sólo 2 clases de patrones de entre un conjunto de entrada. Para ver como trabaja en principio supongamos que es exactamente igual a la neurona artificial introducida por McCulloch y Pitts. Luego veremos cómo es que se agrega una nueva conexión de entrada llamada “tendencia” y cuál es su utilidad.

Veamos un ejemplo puntual de un Perceptrón que realiza una clasificación de vectores de dos dimensiones.

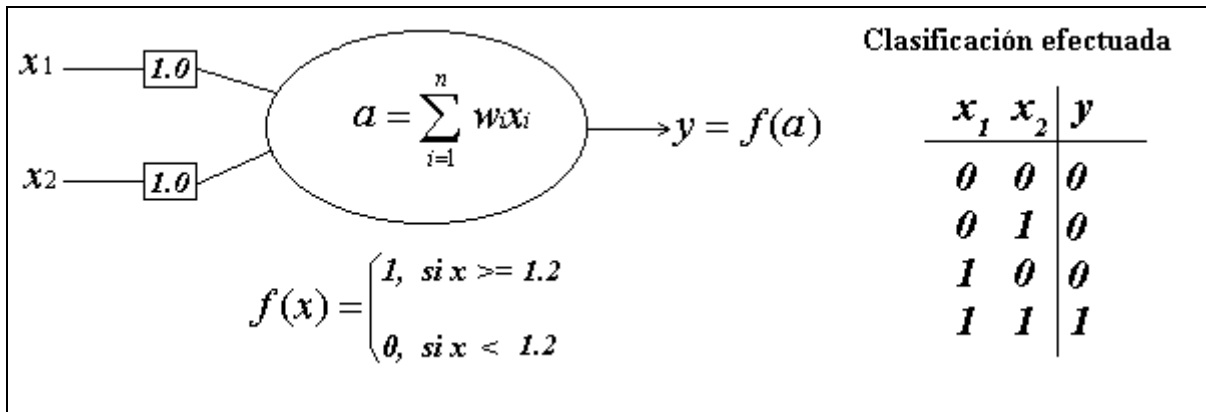


Ilustración 51 Ejemplo de clasificación efectuada por un Perceptrón simple. A la derecha se observa una tabla con el valor de salida del Perceptrón para cada entrada binaria.

En este ejemplo se puede decir que el Perceptrón reconoce el patrón (1,1) o bien que realiza una agrupación en dos clases $A=\{(0,0),(0,1),(1,0)\}$ y $B=\{(1,1)\}$ o simplemente que el Perceptrón calcula el AND lógico.

Veamos la siguiente interpretación geométrica que es útil para comprender las posibilidades y limitaciones de este elemento de procesamiento. Podemos reescribir $f(a)$ como:

$$f(a) = \begin{cases} 1 & \text{si } w_1 \cdot x_1 + w_2 \cdot x_2 \geq u \\ 0 & \text{si } w_1 \cdot x_1 + w_2 \cdot x_2 < u \end{cases}$$

Ahora examinemos la ecuación

$$w_1 x_1 + w_2 x_2 = u$$

En el plano x_1/x_2 representa la ecuación de una recta. En el ejemplo presentado, para $u=1.2$, $w_1=w_2=1$ la recta queda definida por la ecuación

$$x_1 + x_2 = 1.2$$

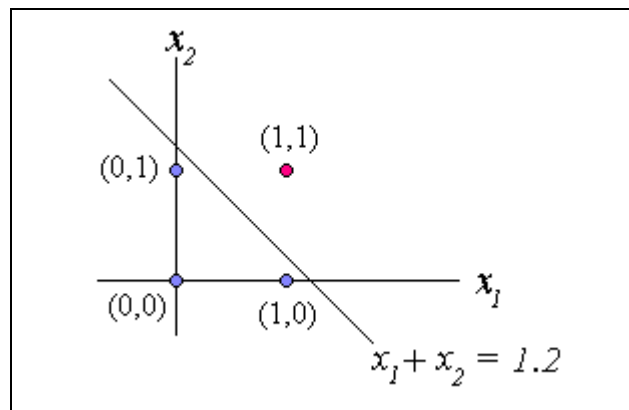


Ilustración 52 Interpretación geométrica de la función de clasificación que realiza el Perceptrón definido en la Ilustración 51. Claramente la salida de la neurona será 1 si como entrada recibe cualquier punto “por encima” o perteneciente a la recta, y 0 en caso contrario.

La recta divide al plano en dos regiones distintas. Entonces se pueden clasificar los puntos de una región como pertenecientes a la clase que posee una salida de 1 y los de la otra región como pertenecientes a la clase que posee una salida nula. Queda claro que la salida del Perceptrón es binaria pero su entrada no está restringida a valores binarios. El Perceptrón de la Ilustración 51 es capaz de clasificar cualquier punto perteneciente a \mathbb{R}^2 .

La interpretación geométrica también pone en evidencia una gran limitación que posee el Perceptrón simple y es la imposibilidad de clasificar grupos de puntos que no sean linealmente separables, tal es el caso de la función lógica XOR.

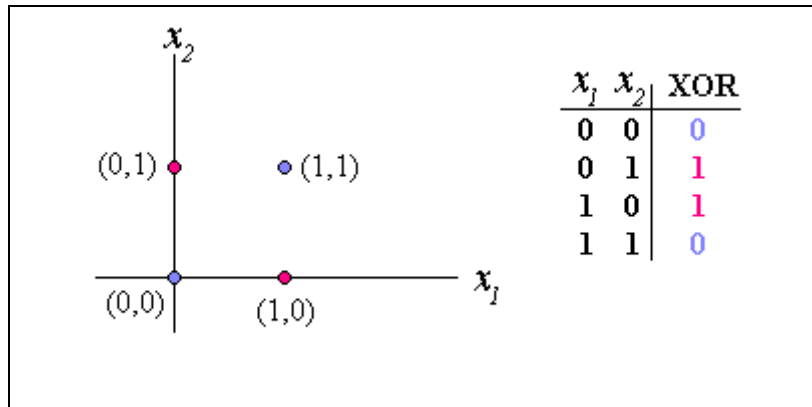


Ilustración 53 XOR Lógico. Esta función no puede implementarse por un Perceptrón ya que los puntos en el plano no son linealmente separables. Observe que no hay forma de trazar una recta que agrupe los puntos rojos de un lado y los azules del otro.

Un problema es linealmente separable si en el espacio n -dimensional de patrones de entrada se puede encontrar un hiperplano (de dimensionalidad $n-1$) que separe a los patrones de una clase de la otra.

En 1969 apareció el libro “*Perceptrons: An Introduction to Computational Geometry*” escrito por Marvin Minsky y Seymour Papert, ambos del MIT, tratando en detalle la limitación de los perceptrones consistente en que sólo pueden clasificar patrones linealmente separables, y como existen muchos problemas de clasificación que no cumplen con esta condición, la aplicabilidad del Perceptrón se hace bastante restrictiva. Ya comentamos que la aparición de este libro significó que el estudio e investigación en redes neuronales se estancara durante más de 15 años, y por lo tanto la solución a estos tipos de problemas de clasificación también se hizo esperar. Fue a partir de la aparición en escena uno de los algoritmos de aprendizaje más exitosos empleado para entrenar redes de tipo Perceptrón multicapa denominado Backpropagation que finalmente el problema se superó.

Volviendo a nuestro ejemplo obsérvese la importancia del umbral u . Si en lugar de $u=1.2$ hubiésemos tenido $u=0$, la recta pasaría por el origen y cualquiera sea su pendiente haría imposible la clasificación AND requerida.

Una variación interesante respecto a la neurona de McCulloch y Pitts es el tratamiento distintivo del umbral u . En lugar de preocuparnos por fijar un valor adecuado, se lo incluye como el peso de una nueva conexión llamada tendencia. Como todo peso sináptico, su valor será determinado adecuadamente por el entrenamiento.

La ecuación

$$\sum_{i=1}^n w_i x_i = u$$

Puede reescribirse como

$$\sum_{i=1}^n w_i x_i - u = 0$$

Y haciendo $w_0 = -u$, y $x_0=1$ nos queda

$$\sum_{i=0}^n w_i x_i = 0$$

De esta manera se puede reescribir la activación del Perceptrón ahora con $(n+1)$ entradas, por el agregado del término de tendencia x_0 cuyo valor es siempre 1. También se reescribe la función de transferencia $f(x)$ ya que el umbral ahora es cero

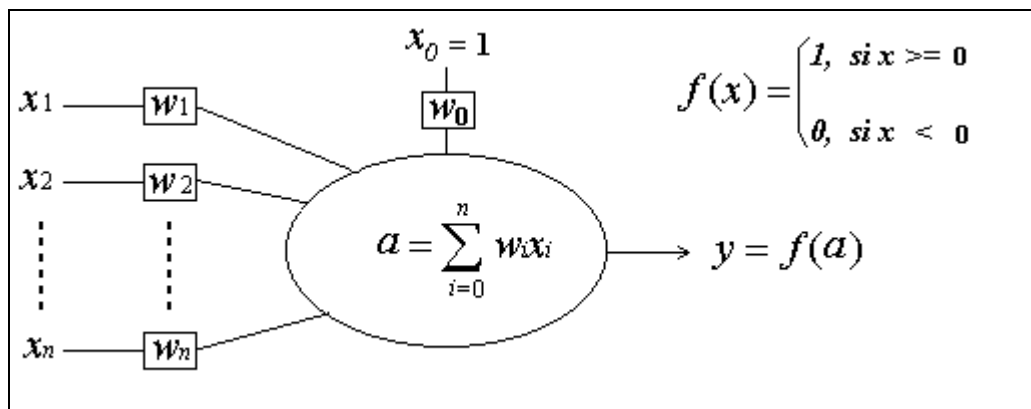


Ilustración 54 Perceptrón simple. Observe que se diferencia de la neurona artificial de McCulloch-Pitts por la inclusión de la nueva entrada x_0 llamada término de tendencia.

Recordemos que la ventaja de la inclusión del término de tendencia está dada por la determinación automática del umbral –indirectamente por w_0 – a través del algoritmo de aprendizaje que veremos inmediatamente.

Aprendizaje del Perceptrón

Hasta aquí no se ha realizado otra cosa que analizar cómo se comporta un Perceptrón en la fase de operación, suponiendo que es útil para realizar una clasificación determinada. Es hora de hablar sobre el entrenamiento a partir del cual el Perceptrón es capaz de aprender. Ya se dijo que la mayor contribución a las RNA de Rosenblatt ha sido la introducción del aprendizaje, es decir en lugar de preocuparse por encontrar los valores correspondientes a los pesos adecuados para el funcionamiento correcto, se somete al dispositivo a una serie de muestras para que éste de manera automática vaya modificándolos hasta que su comportamiento sea el deseado.

Los trabajos de Rosenblatt dieron lugar a que se demostrase un importante resultado conocido con el nombre de teorema de convergencia del Perceptrón. Este teorema afirma que un Perceptrón simple que está aprendiendo a diferenciar tramas de dos clases diferentes, si es que la clasificación puede ser aprendida –recordemos que deben ser clases linealmente separables– entonces una simple regla de corrección de los pesos en base a las desviaciones de las respuestas obtenidas garantiza la obtención de los pesos w_i que configuran la respuesta correcta del PE en un número finito de pasos. Este procedimiento determina la regla de aprendizaje de la red, conocida como regla de Hebb.

La tarea consiste en determinar los pesos sinápticos para que el dispositivo represente la relación entrada/salida lo más fidedignamente posible.

Regla de aprendizaje y algoritmo de entrenamiento

El aprendizaje del Perceptrón es de tipo supervisado. Esto implica que se debe conocer la salida esperada por cada patrón de entrada a clasificar. Será necesario elegir una secuencia de ejemplos $(\vec{x}_1, d_1), (\vec{x}_2, d_2), \dots, (\vec{x}_L, d_L)$ de pares de entrada/salida correctos. El Perceptrón a

través del proceso de entrenamiento debe aprender que a cada vector de entrada \bar{x}_i le corresponde la salida d_i , con d_i perteneciente al conjunto $\{0,1\}$.

En el proceso de entrenamiento el Perceptrón se expone repetidamente a la secuencia de ejemplos hasta que los pesos de la red son ajustados de forma que al final del entrenamiento se obtengan las salidas esperadas para cada uno de los patrones de entrada.

El algoritmo de entrenamiento puede resumirse así:

```

Inicializar los pesos  $w_i$  con valores aleatorios
Repetir
  Para cada par de entrenamiento  $j$  hacer
     $y :=$  salida del Perceptrón para  $\bar{x}_j$ 
    si  $y = d_j$  «no hacer nada. Clasificación correcta de  $\bar{x}_j$ »           (1)
    si  $(y = 0) \text{ and } (d_j=1)$  «hacer  $\bar{w} := \bar{w} + \bar{x}_j$ »           (2)
    si  $(y = 1) \text{ and } (d_j=0)$  «hacer  $\bar{w} := \bar{w} - \bar{x}_j$ »           (3)
Hasta que todos los patrones hayan sido correctamente clasificados
  
```

Debe observarse que las tres expresiones anteriores (1), (2) y (3) pueden escribirse en forma compacta como $\bar{w} := \bar{w} + (d_j - y)\bar{x}_j$

Esta fue la primera regla de aprendizaje que se utilizó para entrenar un Perceptrón, más tarde se agregó un parámetro llamado velocidad de aprendizaje, que aquí denotaré con la letra μ , y que impide cambios bruscos en el vector de peso, un valor adecuado de éste evitará que el sistema oscile continuamente. Este parámetro debe ser fijado de ante mano y habitualmente es un número pequeño, entre 0.1 y 1, de esta manera la fórmula para modificar el vector de pesos se escribe así: $\bar{w} := \bar{w} + \mu(d_j - y)\bar{x}_j$.

Con la adición de este parámetro puede probarse el teorema para la regla de aprendizaje del Perceptrón que asegura la convergencia hacia la solución luego de un número finito de pasos de corrección de los pesos, claro está, sólo en el supuesto que dicha solución sea factible, es decir que el problema de clasificación sea linealmente separable.

Para ilustrar este algoritmo de aprendizaje se desarrollará un pequeño ejemplo. Se desea entrenar a un Perceptrón para que aprenda la siguiente clasificación:

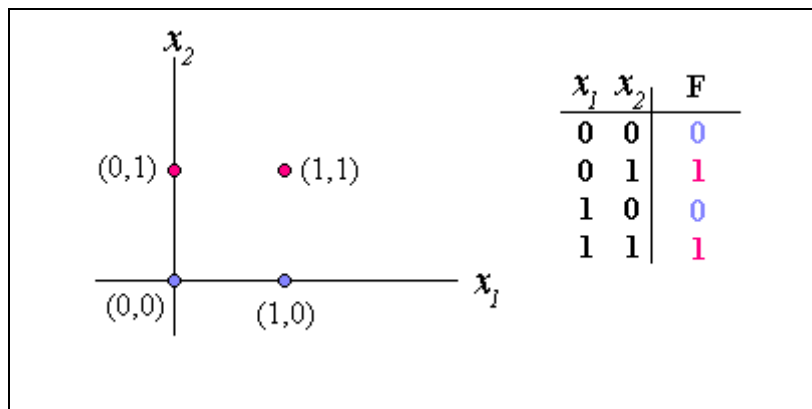


Ilustración 55 La clasificación especificada por F es linealmente separable, así que un Perceptrón será capaz de aprenderla en un número finito de pasos de entrenamiento

En la siguiente tabla se presenta el entrenamiento de un Perceptrón cuyo vector de pesos es inicializado en (0,0,0). El parámetro velocidad de aprendizaje se ha elegido $\mu=0.5$. La secuencia de patrones de entrenamiento será presentada tantas veces como sea necesario hasta que sea clasificada correctamente en toda su totalidad.

Paso	x_0	x_1	x_2	d	w_0	w_1	w_2	y	Actualización de pesos		
									w_0	w_1	w_2
1	1	0	0	0	0	0	0	1	-0,5	0	0
2	1	0	1	1	-0,5	0	0	0	0	0	0,5
3	1	1	0	0	0	0	0,5	1	-0,5	-0,5	0,5
4	1	1	1	1	-0,5	-0,5	0,5	0	0	0	1
5	1	0	0	0	0	0	1	1	-0,5	0	1
6	1	0	1	1	-0,5	0	1	1	-0,5	0	1
7	1	1	0	0	-0,5	0	1	0	-0,5	0	1
8	1	1	1	1	-0,5	0	1	1	-0,5	0	1
9	1	0	0	0	-0,5	0	1	0	-0,5	0	1

En el paso de entrenamiento número 5 el Perceptrón realiza la última actualización del vector de pesos alcanzando el aprendizaje buscado. No obstante sólo se puede estar seguro al comprobar que se clasificaron correctamente todos los patrones de entrada, es por ello que el entrenamiento debió proseguir hasta el paso número 9.

Veamos la interpretación geométrica del aprendizaje en el plano $X_0=1$.

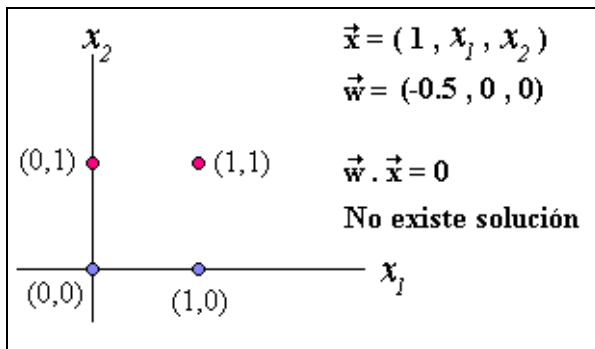


Ilustración 56 Actualización número 1

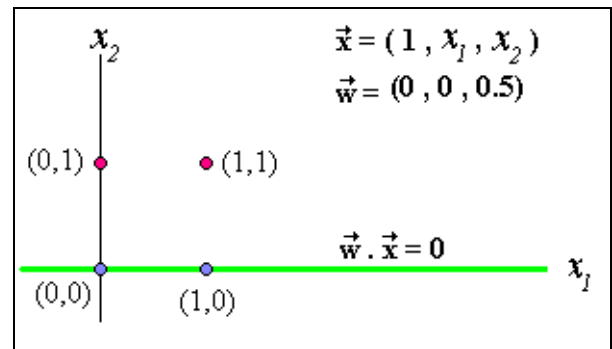


Ilustración 57 actualización número 2

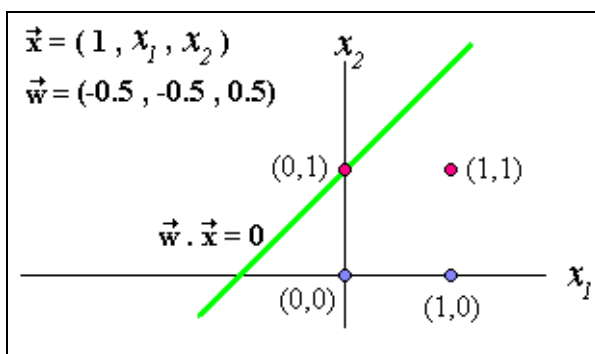


Ilustración 58 Actualización número 3

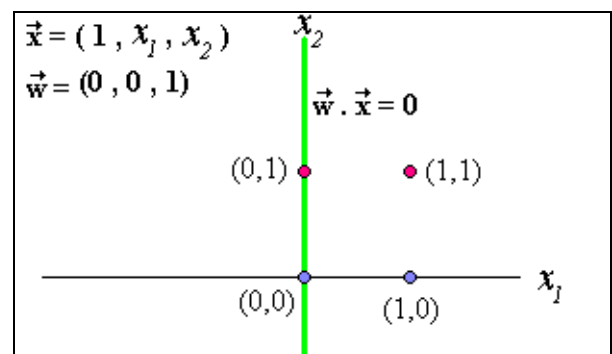


Ilustración 59 Actualización número 4

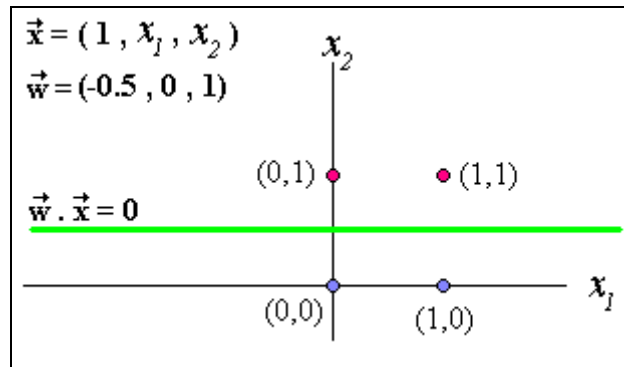


Ilustración 60 Actualización número 5

A partir de la quinta actualización del vector de pesos el dispositivo realiza correctamente la clasificación de todos los patrones de entrada. El entrenamiento termina en el paso número 9, una vez que se comprueba que todos los vectores de entrada han sido bien clasificados.

Perceptrón unicapa

El potencial de un único Perceptrón simple es demasiado limitado, no sólo porque existen clasificaciones que no puede realizar sino porque es capaz de distinguir sólo dos grupos de patrones. Una forma fácil de conseguir mayor potencia es utilizar varios perceptrones simples trabajando en paralelo con las mismas entradas. Esta red unicapa también se la llama asociador lineal y fue inventada por múltiples personas durante el período entre 1968 y 1972. [Hecht-Nielsen91].

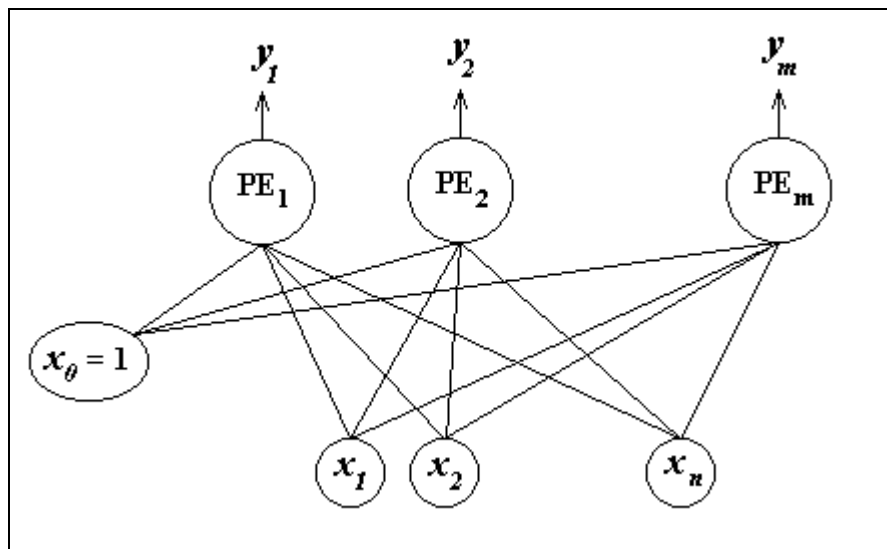


Ilustración 61 Perceptrón unicapa. Este dispositivo puede aprender la correlación existente entre un conjunto de patrones n-dimensional y uno m-dimensional de componentes binarios.

Cada elemento de procesamiento –PE_i– de este dispositivo es un Perceptrón simple que se encarga de aprender la correlación existente entre los patrones de entrada y una componente del vector de salida. Claro está, basta que un solo PE se enfrente a una clasificación no linealmente separable para que todo el dispositivo falle en el aprendizaje. Veamos un ejemplo:

Supongamos que queremos entrenar a un Perceptrón unicapa para que aprenda la función *div2* –parte entera de dividir por dos– de números binarios de hasta tres cifras. Esto significa que la

red debe aprender a realizar una clasificación en cuatro grupos o clases determinada por la siguiente tabla:

\vec{x}	\vec{y}
(0,0,0)	(0,0)
(0,0,1)	(0,0)
(0,1,0)	(0,1)
(0,1,1)	(0,1)
(1,0,0)	(1,0)
(1,0,1)	(1,0)
(1,1,0)	(1,1)
(1,1,1)	(1,1)

Para resolver el problema debe armarse una red como la presentada en la Ilustración 61 compuesta por dos perceptrones simples y entrenarlos hasta lograr el comportamiento esperado. El algoritmo de entrenamiento es el mismo que vimos para el caso de un único PE y puede pensarse como dos entrenamientos independientes donde cada PE_i aprenderá a correlacionar la entrada con la componente y_i del vector de salida.

Perceptrón unicapa		PE ₁				PE ₂			
\vec{X}	\vec{Y}	x_1	x_2	x_3	y_1	x_1	x_2	x_3	y_2
(0,0,0)	(0,0)	0	0	0	0	0	0	0	0
(0,0,1)	(0,0)	0	0	1	0	0	0	1	0
(0,1,0)	(0,1)	0	1	0	0	0	1	0	1
(0,1,1)	(0,1)	0	1	1	0	0	1	1	1
(1,0,0)	(1,0)	1	0	0	1	1	0	0	0
(1,0,1)	(1,0)	1	0	1	1	1	0	1	0
(1,1,0)	(1,1)	1	1	0	1	1	1	0	1
(1,1,1)	(1,1)	1	1	1	1	1	1	1	1

Ilustración 62 El aprendizaje de la función div2 se realiza por medio del entrenamiento de dos perceptrones simples PE₁ y PE₂.

Como se dijo anteriormente el éxito del algoritmo de aprendizaje radica en que ambos PEs deban realizar una clasificación linealmente separable. Puede verse fácilmente que efectivamente, así sucede:

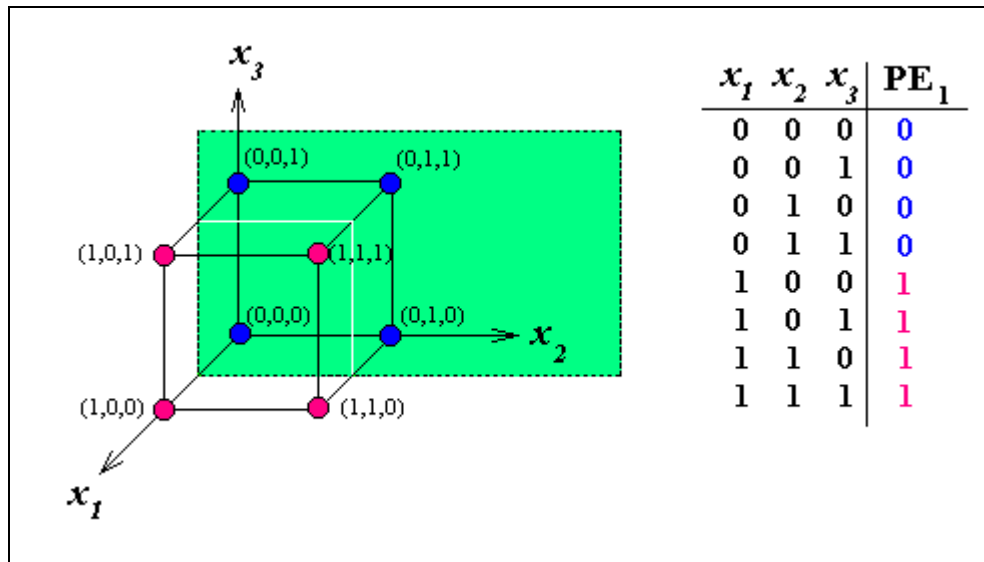


Ilustración 63 La clasificación será aprendida por PE_1 por ser linealmente separable.

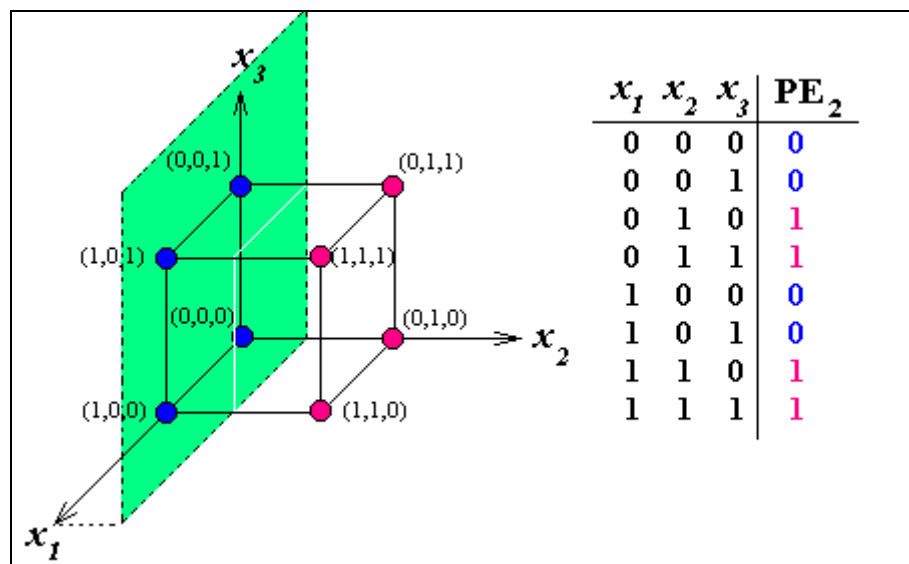


Ilustración 64 La clasificación será aprendida por PE_2 por ser linealmente separable.

Puede concluirse que el Perceptrón unicapa será entrenado con éxito en el aprendizaje de la función *div2*.

El proceso para determinar si un problema es linealmente separable o no, se realiza gráficamente sin problema, cuando los patrones de entrada generan un espacio de dos dimensiones. Sin embargo, esta visualización se dificulta cuando el conjunto de patrones de entrada es de tres dimensiones, y resulta imposible de observar gráficamente cuando los patrones de entrada son de dimensiones superiores. En este último caso se requiere plantear condiciones de desigualdad que permitan comprobar la separabilidad lineal de los patrones. Se debe resolver el sistema formado por tantas inecuaciones como patrones de entrada existan y esto se realiza con base en la ecuación de salida del Perceptrón de la siguiente manera:

$$\sum_{i=0}^n w_i x_i \geq 0, \quad \text{para aquellos patrones cuya salida deseada sea 1}$$

$$\sum_{i=0}^n w_i x_i < 0, \quad \text{para aquellos patrones cuya salida deseada sea 0}$$

Perceptrón multicapa

Volvamos por un instante al problema de clasificación de patrones de la función XOR, que como ya se dijo, es imposible de ser realizado correctamente por un Perceptrón simple. Intentemos aproximar una solución. La Ilustración 53 –pág. 129– sugiere que se podría descomponer correctamente el espacio si tuvieran tres regiones. Una región pertenecería a una de las clases de salida y las otras dos pertenecerían a la segunda clase de salida. Así que si en lugar de utilizar únicamente una neurona de salida se utilizaran dos, se obtendrían dos rectas por lo que podrían delimitarse las tres zonas necesarias. Pero para poder elegir entre una zona u otra de las tres, es necesario utilizar otra capa con una neurona cuyas entradas serán las salidas de las neuronas anteriores, por tanto se ha de utilizar una red de tres neuronas, distribuidas en dos capas para solucionar este problema.

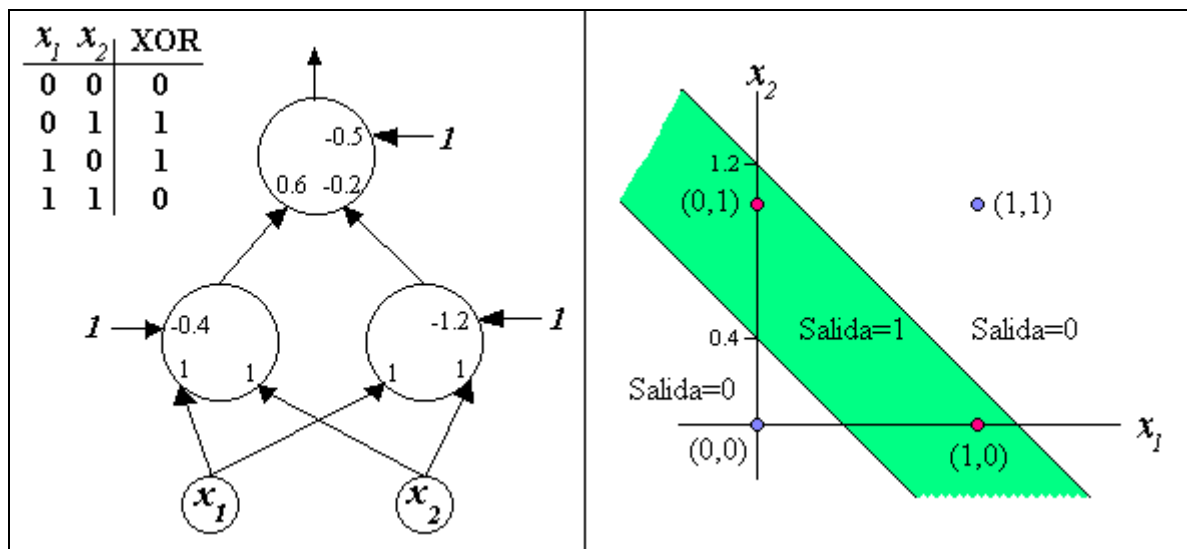


Ilustración 65 Esta red resuelve correctamente el problema XOR. Los números que figuran dentro de los nodos en el gráfico de la izquierda representan los pesos de cada una de las conexiones. La capa oculta proporciona dos rectas que separan el plano en tres regiones. Las dos regiones que contienen los puntos (0,0) y (1,1) se asocian a una salida nula de la red. La región central se asocia a una salida de la red con valor igual a 1.

La adición de dos unidades de capa oculta, o capa intermedia dan a la red la flexibilidad necesaria para resolver el problema del XOR.

Un Perceptrón multicapa es una red con alimentación hacia delante –feedforward–, compuesta de varias capas de neuronas entre la entrada y la salida de la misma, esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el Perceptrón de un solo nivel.


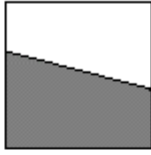
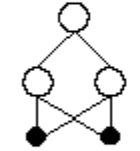
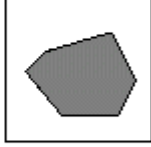
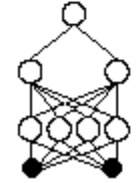

Estructura	Tipos de regiones de decisión	Formas de las regiones de decisión
una capa 	medio plano limitado por un hiperplano	
dos capas 	regiones convexas o cerradas	
tres capas 	arbitraria (limitada por el número de nodos)	

Ilustración 66 Imagen editada a partir de gráfico extraído de [FloresMendez99]

El Perceptrón simple sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de entrada de los patrones. Un Perceptrón con dos capas, puede formar cualquier región convexa en este espacio. Las regiones convexas se forman mediante la intersección de las regiones que definen cada neurona de la capa oculta, cada uno de estos elementos se comporta como un Perceptrón simple, activándose su salida para los patrones de un lado del hiperplano. Si la neurona del nivel de salida implementa la función lógica AND, la región de decisión resulta ser la intersección de todos los semiplanos formados en el nivel anterior. Esta región de decisión será una región convexa con un número de lados a lo sumo igual al número de neuronas de la capa oculta.

A partir de este análisis surge el interrogante respecto a los criterios de selección para las neuronas de las capas ocultas de una red multicapa, este número en general debe ser lo suficientemente grande como para que se forme una región compleja que pueda resolver el problema, sin embargo no debe ser muy grande pues la estimación de los pesos puede ser no confiable para el conjunto de los patrones de entrada disponibles. Hasta el momento no hay un criterio establecido para determinar la configuración de la red y esto depende más bien de la experiencia del diseñador. Otra opción es intentar alguna forma automática para determinar este y otros parámetros de la red. La Neuroevolución –que veremos más adelante– puede constituir una solución aceptable esta problemática.

No se pretende implicar que todas las críticas del Perceptrón podrían tener respuesta añadiendo capas ocultas a la estructura. Lo que se pretende sugerir es que las técnicas siguen avanzando hasta sistemas cuyas capacidades son cada vez mayores. De hecho el Perceptrón multicapa no se transformó en una red exitosa sino hasta la aparición de un algoritmo de aprendizaje adecuado llamado Backpropagation. Actualmente este es el algoritmo que más se emplea en la obtención de los pesos de red multicapa de perceptrones y constituye una generalización del algoritmo de mínimos cuadrados –introducido por la red Adaline– y emplea una técnica de búsqueda del gradiente que minimice el error esperado entre la salida actual y la deseada. Ambos algoritmos se presentan en los apartados subsiguientes.

Adaline

Dos años después que Frank Rosenblatt presentara el modelo del Perceptrón, Bernard Widrow y su estudiante Marcian Hoff introdujeron el Adaline y su regla de aprendizaje llamado algoritmo de *mínimos cuadrados* –LMS por sus siglas en inglés– conocida también como regla *Delta*, o regla de *Widrow-Hoff*.

El Adaline es un dispositivo que consta de un único elemento de procesamiento, así que, al igual que el Perceptrón simple, no constituye técnicamente una red neuronal. Sin embargo, se trata de una estructura muy importante por dos razones: 1) su salida ya no se limita a un valor binario sino por el contrario, puede tomar cualquier valor real; 2) el algoritmo LMS es más potente que la regla de aprendizaje del Perceptrón ya que minimiza el error cuadrático medio y ha servido de inspiración para el desarrollo de otros algoritmos, como por ejemplo el Backpropagation que veremos más adelante.

El término Adaline es una sigla, sin embargo su significado cambió ligeramente a finales de los años sesenta cuando decayó el estudio de las redes neuronales, inicialmente se llamaba ADAPtive LInear NEuron –Neurona Lineal Adaptativa–, para pasar después a ser ADAPtive LInear Element –Elemento Lineal Adaptativo–.

El Adaline es similar al Perceptrón, excepto en que su función de transferencia es la función *Identidad* en lugar de la función escalón que usa el Perceptrón. Es decir, la salida del Adaline, es igual a la activación.

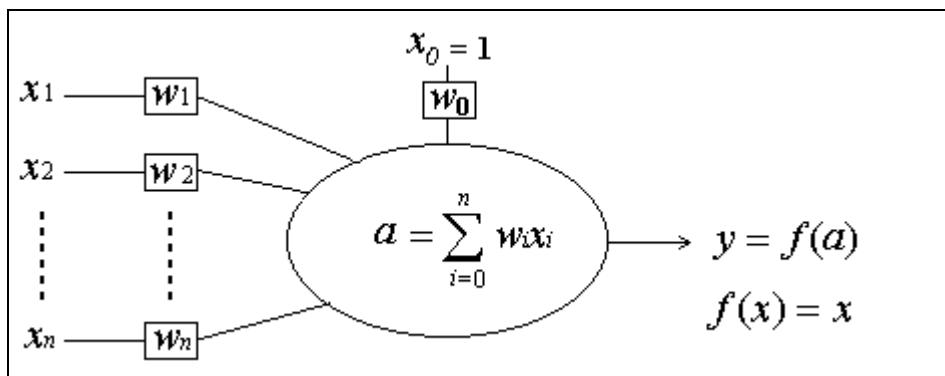


Ilustración 67 Adaline. Obsérvese que a diferencia del Perceptrón no existe la función escalón. La salida es igual a la activación. En ocasiones la función de transferencia *Identidad* se reemplaza por otra bajo la condición que sea continua y diferenciable.

En el gráfico se ha denotado explícitamente que la salida del Adaline es una función de la activación. Inicialmente la función *Identidad* permitió que el dispositivo sea un verdadero combinador lineal, no obstante, con el tiempo se han probado otras funciones como la función sigmoide –que veremos más adelante– que ya no es una función lineal. La posibilidad de utilizar otras funciones de transferencia está dada por la flexibilidad de la regla de aprendizaje que lo contempla siempre que dichas funciones sean continuas y derivables.

El Adaline presenta la misma limitación del Perceptrón en cuanto al tipo de problemas que pueden resolver, ambos dispositivos pueden solo resolver problemas linealmente separables. También al igual que el Perceptrón unicapa, se pueden combinar un cierto número de elementos de procesamiento Adaline en la capa de salida para obtener clasificaciones más potentes. En este caso la regla de aprendizaje se aplica sobre cada uno de los PE de manera individual.

Regla de aprendizaje LMS

El algoritmo de aprendizaje es el mismo que el presentado para el Perceptrón, sólo difiere en que la modificación del vector de pesos ha sido calculada para minimizar el error cuadrático medio como veremos inmediatamente.

Dado el conjunto de entrenamiento $\{(\bar{x}_1, d_1), (\bar{x}_2, d_2), \dots, (\bar{x}_L, d_L)\}$ de alguna función de procesamiento que asocia a los vectores de entrada \bar{x}_i los valores de salida deseados d_i se pretende encontrar el mejor vector de pesos \bar{w} para que el Adaline lleve a cabo esta asociación.

El mejor vector de pesos será aquel que elimine por completo, o en su defecto haga mínima la diferencia entre la salida del Adaline y el valor deseado para todos los vectores de entrada. Una solución para lograrlo es minimizar el error cuadrático medio, o valor esperado del error, definido de la siguiente manera:

$$\xi = \frac{1}{L} \sum_{k=1}^L (d_k - y_k)^2$$

Donde y_k es el valor de salida del Adaline para el k -ésimo vector de entrada y L es la cardinalidad del conjunto de entrenamiento. Claramente ξ es una función de \bar{w} , –recordemos que y_k lo es– y se desea hallar el vector de pesos que lo minimice, para ello se debe conocer la forma de la función ξ . Para un vector de dos dimensiones la gráfica de ξ es un paraboloide. Además tiene que poseer una concavidad hacia arriba, puesto que todas las combinaciones de pesos deben dar lugar a un valor no negativo para el error cuadrático medio ξ . Este resultado es general, y se obtiene independientemente de las dimensiones del vector de pesos. En el caso de que las dimensiones sean más de dos, el paraboloide se conoce con el nombre de hiperparaboloide [Freeman93].

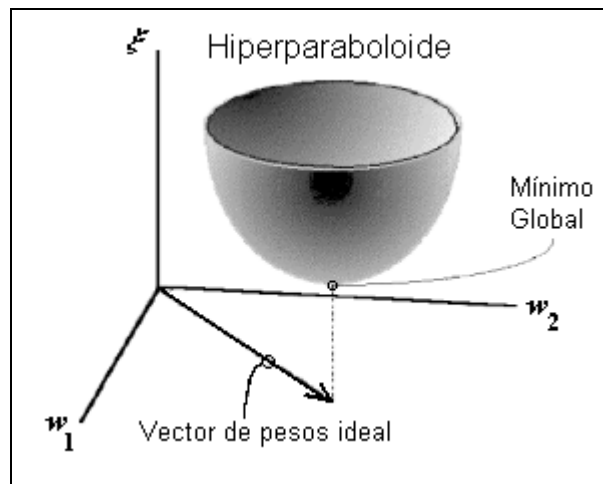


Ilustración 68 Gráficamente, la función ξ es un paraboloide cóncavo hacia arriba.

Analíticamente puede hallarse el vector de pesos ideal igualando las derivadas parciales a cero para obtener el mínimo de la función, no obstante los cálculos son engorrosos y aquí se trata de encontrar un método iterativo por el cual el dispositivo vaya haciendo correcciones en los pesos hasta encontrarlo automáticamente.

El algoritmo de aprendizaje del Adaline, al igual que el del Perceptrón, corrige el vector de pesos en cada paso de entrenamiento, pero a diferencia del dispositivo de Rosenblatt, éste lo hace mediante el método del descenso más pronunciado por la superficie de ξ . Dado que el vector de pesos de este procedimiento es variable en el tiempo lo escribimos como una

función explícita del paso temporal, t . En cada paso, el próximo vector de pesos se calcula según

$$\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) + \Delta \bar{\mathbf{w}}(t)$$

En donde $\Delta \bar{\mathbf{w}}(t)$ es el cambio que sufre $\bar{\mathbf{w}}$ en el t -ésimo instante. Se está buscando la dirección del descenso más pronunciado en cada punto de la superficie, así que es necesario calcular el gradiente de la superficie –que proporciona la dirección de la pendiente más pronunciada hacia arriba–. La dirección opuesta es la que utilizaremos. Para obtener una magnitud del cambio se multiplica por una constante positiva apropiada μ . Así nos queda la siguiente expresión

$$\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) - \mu \bar{\nabla} \xi(\bar{\mathbf{w}}(t))$$

Recordemos que el vector gradiente se obtiene a partir de las derivadas parciales de la siguiente forma:

$$\bar{\nabla} \xi = \left(\frac{\partial \xi}{\partial w_0}, \frac{\partial \xi}{\partial w_1}, \dots, \frac{\partial \xi}{\partial w_n} \right)$$

No obstante, el método utiliza una aproximación de ξ fácilmente calculable con la información conocida de forma explícita en cada iteración, me refiero al error cuadrático del paso actual ε^2 que, para el caso de estar procesando el k -ésimo par de entrenamiento, se calcula así:

$$\varepsilon^2 = (d_k - y_k)^2$$

Recordemos que $y = f(a)$ con $a = \sum w_i x_i$, así las componentes del gradiente de ε^2 se calculan como:

$$\frac{\partial \varepsilon^2}{\partial w_i} = 2(d - y_k)(-1)f'(a).x_i$$

$$\frac{\partial \varepsilon^2}{\partial w_i} = -2(d - y_k)x_i f'(a)$$

De aquí la restricción que f sea diferenciable. Podemos escribir el vector gradiente de ε^2 como:

$$\bar{\nabla} \varepsilon^2 = -2(d_k - y_k)\bar{\mathbf{x}}_k f'(a)$$

Por lo tanto utilizando $\bar{\nabla} \varepsilon^2$ como aproximación de $\bar{\nabla} \xi$ se puede reescribir la ecuación $\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) - \mu \bar{\nabla} \xi(\bar{\mathbf{w}}(t))$ de la siguiente manera:

$$\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) - \mu(-2)(d_k - y_k)\bar{\mathbf{x}}_k f'(a)$$

$$\boxed{\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) + 2\mu(d_k - y_k)\bar{\mathbf{x}}_k f'(a)}$$

Esta última expresión representa la regla de aprendizaje del Adaline. Debe aclararse que esta fórmula ya se haya generalizada para el caso de una función de transferencia diferenciable cualquiera. Para el Adaline original, f es la función identidad y por lo tanto $f'(a)=1$ y su regla de aprendizaje se reduce a:

$$\boxed{\bar{\mathbf{w}}(t+1) = \bar{\mathbf{w}}(t) + 2\mu(d_k - y_k)\bar{\mathbf{x}}_k}$$

El parámetro μ determina la estabilidad y la velocidad de convergencia del vector de pesos hacia el valor de error mínimo. Dado que se ha utilizado una aproximación del gradiente, el camino que sigue el vector de pesos al bajar por la superficie de pesos hacia el mínimo no será una curva suave. Además a medida que nos acerquemos al mínimo el módulo del gradiente será más chico y por lo tanto nos acercaremos asintóticamente, cada vez con pasos más pequeños.

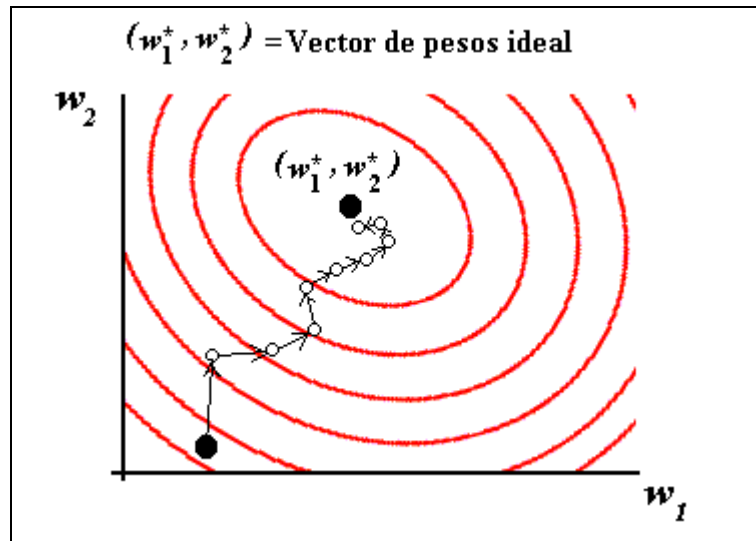


Ilustración 69 Diagrama de contornos de la superficie del error cuadrático medio. Puede observarse que la hipotética ruta que sigue el vector de pesos en su búsqueda del error mínimo utilizando el algoritmo LMS no es una curva suave, porque se está aproximando el gradiente en cada punto. Obsérvese también que el tamaño del paso se vuelve cada vez más pequeño a medida que nos aproximamos a la solución de error mínimo.

Los cambios del vector de peso deben hacerse relativamente pequeños en cada iteración. Si los cambios son demasiado grandes, el vector de pesos podría vagar por la superficie, sin encontrar nunca el mínimo. La misión del parámetro μ es evitar esta búsqueda sin tino. Por lo tanto, determinar su valor más adecuado es fundamental pues tiene efectos significativos sobre el entrenamiento. Según lo expone Freeman en [Freeman93] la experiencia parece ser el mejor profesor a la hora de seleccionar un valor adecuado para μ y plantea tomar un valor tal que los pesos no cambien más que una cierta fracción de su valor actual. También sugiere que, incrementar el valor de μ cuando los cambios en el vector comienzan a hacerse demasiado pequeños –como consecuencia de la proximidad del *fondo* de la superficie– puede acelerar la convergencia al mínimo buscado. Sin embargo ha de tenerse en cuenta que un valor más grande del debido ocasionará que los pesos vayan “*dando saltos en torno al fondo*” de la superficie, lo cual producirá un error global inadmisibles. Nuevamente, la experiencia nos indicará la decisión válida.

Backpropagation

Ya se ha visto en este capítulo, cuando se analizó el Perceptrón de Rosenblatt, la limitación de este dispositivo para clasificar patrones que no sean linealmente separables. También se expuso cómo es posible combinar perceptrones simples en una red multicapa que resuelva satisfactoriamente uno de estos problemas como lo constituye la función XOR. Incluso se ha mencionado que una red Perceptrón con dos capas, puede formar cualquier región convexa de decisión, y con tres puede definir cualquier región arbitraria haciendo posible cualquier clasificación –Ver “Perceptrón multicapa” pág. 136–.

Lo que aún está faltando es el enunciado de algún algoritmo de aprendizaje capaz de encontrar los pesos adecuados para este tipo de redes multicapa. Pues entonces es hora de presentar el algoritmo Backpropagation¹.

El primer algoritmo de entrenamiento para redes multicapa fue desarrollado por Paul Werbos en 1974, este se desarrolló en un contexto general, para cualquier tipo de redes, siendo las redes neuronales una aplicación especial, razón por la cual el algoritmo no fue aceptado dentro de la comunidad de desarrolladores de redes neuronales. Fue solo hasta mediados de los años 80 cuando el algoritmo Backpropagation o algoritmo de propagación inversa fue redescubierto al mismo tiempo por varios investigadores. El algoritmo se popularizó cuando fue incluido en el libro "Parallel Distributed Processing Group" por los psicólogos David Rumelhart y James McClelland. La publicación de este libro trajo consigo un auge en las investigaciones con redes neuronales, siendo la Backpropagation una de las redes más ampliamente empleadas, aun en nuestros días.

La mayoría de los sistemas actuales de computo se han diseñado para llevar a cabo funciones matemáticas y lógicas a una velocidad que resulta asombrosamente alta para el ser humano. Sin embargo la destreza matemática no es lo que se necesita para solucionar problemas de reconocimiento de patrones en entornos ruidosos, característica que incluso dentro de un espacio de entrada relativamente pequeño, puede llegar a consumir mucho tiempo.

Lo que se necesita es un sistema de procesamiento que sea capaz de examinar todos los patrones en paralelo. Idealmente ese sistema no tendría que ser programado explícitamente, lo que haría es adaptarse a sí mismo para aprender la relación entre un conjunto de patrones dado como ejemplo y ser capaz de aplicar la misma relación a nuevos patrones de entrada – generalización–. Este sistema debe estar en capacidad de concentrarse en las características de una entrada arbitraria que se asemeje a otros patrones vistos previamente, sin que ninguna señal de ruido lo afecte. Este sistema fue el gran aporte de la red de propagación inversa, Backpropagation.

La Backpropagation es un tipo de red de aprendizaje supervisado, que emplea un ciclo *propagación–adaptación* de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento.

¹ La bibliografía consultada utiliza el término backpropagation para referirse tanto al algoritmo de aprendizaje que utiliza una red multicapa como al tipo mismo de red neuronal. Así es frecuente encontrar textos donde se habla de “la red backpropagation”. Aquí también se incurrirá en este abuso del lenguaje y al decir “red backpropagation” estaré aludiendo a una red de tipo Perceptrón multicapa feedforward (con propagación hacia delante, sin ciclos) que es entrenada con el algoritmo del mismo nombre.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual han sido entrenadas.

Varias investigaciones han demostrado que, durante el proceso de entrenamiento, la red Backpropagation tiende a desarrollar relaciones internas entre neuronas con el fin de organizar los datos de entrenamiento en clases. Esta tendencia se puede extrapolar, para llegar a la hipótesis consistente en que todas las unidades de la capa oculta de una Backpropagation son asociadas de alguna manera a características específicas del patrón de entrada como consecuencia del entrenamiento. Lo que sea o no exactamente la asociación puede no resultar evidente para el observador humano, lo importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas cuando se le dan las entradas, en el proceso de entrenamiento. Esta misma representación interna se puede aplicar a entradas que la red no haya visto antes, y la red clasificará estas entradas según las características que compartan con los ejemplos de entrenamiento.

Este último concepto es importantísimo, se trata ni más ni menos que de la capacidad de poder generalizar a través de ejemplos. Está relacionado directamente con nuestras capacidades cerebrales superiores. Desde niños aprendemos muchos conceptos a partir de la generalización de algunas muestras concretas. Por ejemplo el concepto “gato” lo adquirimos al conocer no uno sino varios ejemplares de este felino, así aprendemos a reconocer ciertas características comunes que servirán para poder clasificar correctamente a un animal que nunca hemos visto antes, como representante de la especie “gato”.

Estructura y funcionamiento de la red Backpropagation

La Backpropagation es una red formada por capas, con propagación hacia delante, que está completamente interconectada entre capas. Por lo tanto no existen conexiones de retroalimentación, ni conexiones que salten una capa para ir directamente a otra superior.

Se presentará la derivación de la regla GDR para una red formada por una capa oculta –entre la entrada y la capa de salida– pero se pueden admitir más¹.

¹ Freeman en [Freeman93] habla de una red de tres capas –considerando a las neuronas de entrada como la primera capa– no obstante otros autores no consideran a la entrada como una capa puesto que no realiza ningún procesamiento ni interviene en el aprendizaje, simplemente representa los datos de entrada. En el presente trabajo no contamos como una capa a las neuronas de entrada, así al decir “una red de dos capas” nos estaremos refiriendo a una formada por una capa oculta y otra de salida.

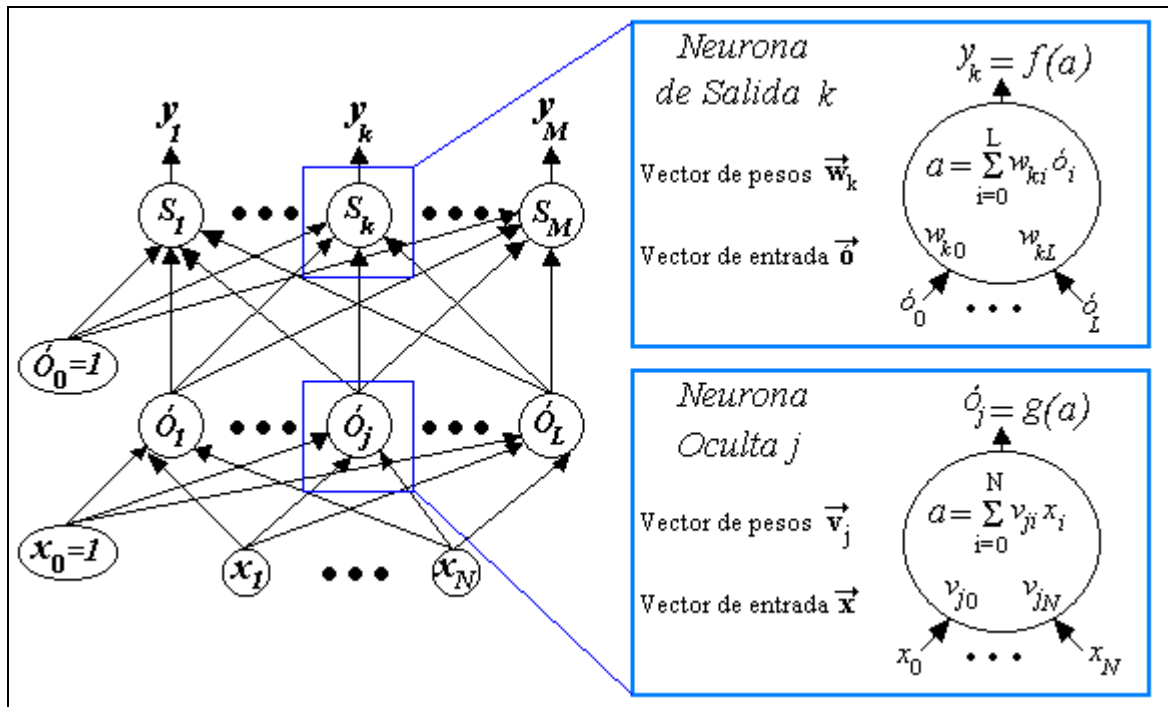


Ilustración 70 Arquitectura de una red Backpropagation de dos capas. Los pesos y unidades de tendencia son opcionales. Se han diferenciado las funciones de transferencia g en la capa oculta y f en la capa de salida para que resulte más clara la derivación de la regla de aprendizaje, aunque por lo general es la misma en toda la red.

La función de transferencia de cada PE tanto en la capa de salida como en la capa oculta puede ser cualquier función derivable. Este requisito excluye la función escalón que se utilizó en el Perceptrón pues no es derivable en todo su dominio. Las funciones más utilizadas son la función Identidad y la sigmoide logística. La selección de esta función depende de la forma en que se pretende representar los datos de salida. Por ejemplo, si se desea que las unidades de salida sean binarias, se utiliza una función sigmoide, porque esta función limita la salida y es casi biestable, pero también es derivable, en otros casos es tan aplicable una función de salida lineal como una sigmoide.

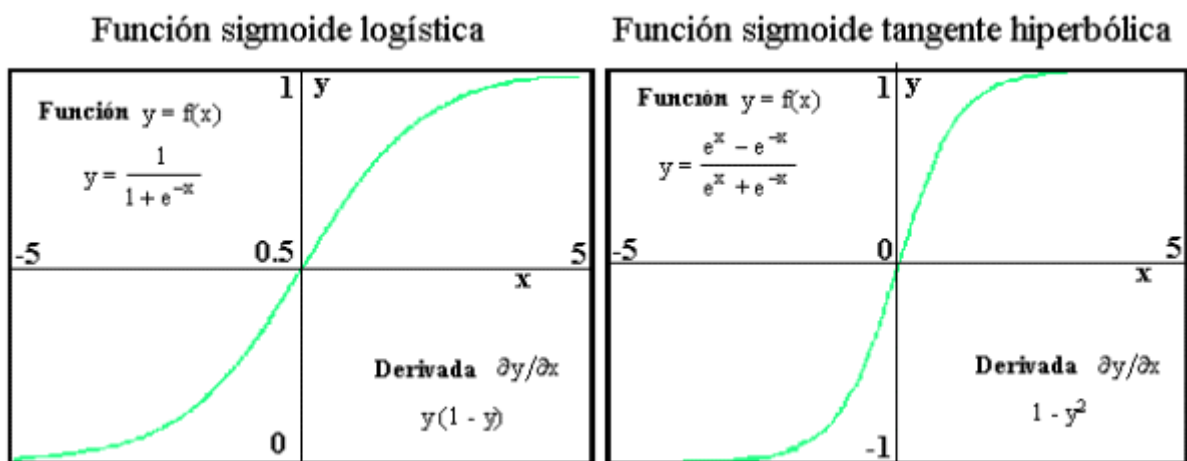


Ilustración 71 Funciones sigmoideas.

Regla de aprendizaje Delta generalizada –GDR–

Supongamos que se tiene un conjunto de pares de vectores de la forma (\vec{x}, \vec{d}) que son ejemplos de una correspondencia funcional que se pretende sea aprendida por la red, con $\vec{x} \in \mathbf{R}^N$ y $\vec{d} \in \mathbf{R}^M$. Se derivará un método para entrenarla, suponiendo que los pares de vectores de entrenamiento se hayan seleccionado adecuadamente y que haya un número suficiente de ellos, más adelante daremos precisiones sobre que significa *adecuadamente* y *suficiente*.

El algoritmo propiamente dicho no difiere del ya presentado para el Perceptrón y el Adaline sino en la forma de actualizar los pesos. Se presenta un vector de entrada, se calcula la salida, se determina el error, se corrigen los pesos de las conexiones y se vuelve a repetir el proceso hasta que el error se reduzca a un valor aceptable. Veamos entonces como se actualizan los pesos de las conexiones de cada capa de la red.

Actualización de pesos de la capa de salida

Téngase presente la Ilustración 70 para la siguiente deducción. Para cada presentación de un vector de entrada se pretende minimizar el error global de la capa de salida que está formado por la suma de aquellos errores de cada unidad de salida. Más precisamente la expresión del error que se desea minimizar es:

$$E = \frac{1}{2} \sum_{k=1}^M (d_k - y_k)^2$$

El factor 1/2 de la ecuación aparece por conveniencia para calcular derivadas más adelante, dado que aparecerá una constante arbitraria μ en el resultado final la presencia de este factor no invalida la deducción de la regla. Además está claro que si minimizamos E también estamos minimizando $\sum (d_k - y_k)^2$.

Puede verse a E como una función de todas las componentes de todos los vectores de pesos de las neuronas de salida. Para determinar el sentido en que se deben cambiar los pesos, se calcula el valor negativo del gradiente de E respecto de todos los pesos w_{ki} .

$$\vec{\nabla} E = \left(\frac{\partial E}{\partial w_{10}}, \frac{\partial E}{\partial w_{11}}, \dots, \frac{\partial E}{\partial w_{1L}}, \frac{\partial E}{\partial w_{20}}, \frac{\partial E}{\partial w_{21}}, \dots, \frac{\partial E}{\partial w_{2L}}, \dots, \frac{\partial E}{\partial w_{M0}}, \frac{\partial E}{\partial w_{M1}}, \dots, \frac{\partial E}{\partial w_{ML}} \right)$$

Calculemos ahora la componente genérica correspondiente a la i -ésima coordenada del vector de pesos de la k -ésima neurona de salida .

$$\frac{\partial E}{\partial w_{ki}} = \frac{1}{2} 2(d_k - y_k)(-1) \frac{\partial y_k}{\partial w_{ki}}$$

$$\frac{\partial y_k}{\partial w_{ki}} = f'(a) \frac{\partial a}{\partial w_{ki}}$$

$$\frac{\partial a}{\partial w_{ki}} = \delta_i$$

$$\therefore \frac{\partial E}{\partial w_{ki}} = -(d_k - y_k) f'(a) \delta_i$$

$$\therefore -\frac{\partial E}{\partial w_{ki}} = (d_k - y_k) f'(a) \delta_i$$

El cambio de peso será en el sentido y proporcional al gradiente negativo. De esta manera los pesos de la capa de salida se actualizan según lo siguiente:

$$w_{ki}(t+1) = w_{ki}(t) + \mu(d_k - y_k)f'(a)\delta_i$$

El factor μ se denomina parámetro de velocidad de aprendizaje, siempre es positivo y suele ser menor que 1, más adelante se darán algunas precisiones más respecto a él.

Si la función de transferencia es la Identidad, la ecuación para la actualización de pesos nos queda:

$$w_{ki}(t+1) = w_{ki}(t) + \mu(d_k - y_k)\delta_i$$

Si por el contrario utilizamos como función de transferencia la función sigmoide logística la actualización es la siguiente:

$$w_{ki}(t+1) = w_{ki}(t) + \mu(d_k - y_k)y_k(1 - y_k)\delta_i$$

Actualización de los pesos de capas ocultas

Se desea repetir para la capa oculta el mismo tipo de cálculo que se ha realizado para la capa de salida. Debe notarse que la expresión del error que se utiliza para calcular la actualización de los pesos de la capa de salida también puede verse como una función de todos los pesos de todas las neuronas de la capa oculta. Veamos que esto es cierto:

$$E = \frac{1}{2} \sum_{k=1}^M (d_k - y_k)^2$$

$$y_k = f(a^s)$$

$$a^s = \sum_{j=0}^L w_{kj} \delta_j$$

$$\delta_j = g(a)$$

$$a = \sum_{i=0}^N v_{ji} x_i$$

Así podemos reescribir E como:

$$E = \frac{1}{2} \sum_{k=1}^M (d_k - f(\sum_{j=0}^L w_{kj} g(\sum_{i=0}^N v_{ji} x_i)))^2$$

El gradiente de E respecto a los pesos de la capa oculta se calcula de la siguiente forma:

$$\vec{\nabla} E = \left(\frac{\partial E}{\partial v_{10}}, \frac{\partial E}{\partial v_{11}}, \dots, \frac{\partial E}{\partial v_{1N}}, \frac{\partial E}{\partial v_{20}}, \frac{\partial E}{\partial v_{21}}, \dots, \frac{\partial E}{\partial v_{2N}}, \dots, \frac{\partial E}{\partial v_{L0}}, \frac{\partial E}{\partial v_{L1}}, \dots, \frac{\partial E}{\partial v_{LN}} \right)$$

Y la componente genérica correspondiente a la i -ésima coordenada del vector de pesos de la j -ésima neurona de salida se calcula así:

$$\frac{\partial E}{\partial v_{ji}} = \frac{1}{2} \sum_{k=1}^M 2(d_k - y_k)(-1) \frac{\partial y_k}{\partial a^s} \cdot \frac{\partial a^s}{\partial \delta_j} \cdot \frac{\partial \delta_j}{\partial a} \cdot \frac{\partial a}{\partial v_{ji}}$$

Cada uno de estos factores puede calcularse explícitamente a partir de las ecuaciones anteriores. El resultado es el que sigue:

$$\frac{\partial E}{\partial v_{ji}} = - \sum_{k=1}^M (d_k - y_k) f'(a^s) w_{kj} g'(a) x_i$$

Se actualizan los pesos de la capa oculta proporcionalmente al valor negativo de la ecuación anterior introduciendo nuevamente el factor μ quien una vez más representa la velocidad de aprendizaje.

$$\Delta v_{ji} = \mu g'(a) x_i \sum_{k=1}^M (d_k - y_k) f'(a^s) w_{kj}$$

Obsérvese que todas las actualizaciones de pesos de la capa oculta dependen de todos los términos de error $(d_k - y_k)$ de la capa de salida. De aquí surge la noción de que los errores conocidos de la capa de salida se propagan hacia atrás, hacia la capa oculta, para determinar los cambios de pesos adecuados en esa capa. Los términos de error de las unidades ocultas deben calcularse antes que hayan sido actualizados los pesos de conexión w_{kj} con las unidades de la capa de salida.

Por lo tanto, los pesos de la capa oculta se actualizan de la manera siguiente:

$$v_{ji}(t+1) = v_{ji}(t) + \mu g'(a) x_i \sum_{k=1}^M (d_k - y_k) f'(a^s) w_{kj}$$

Variantes del algoritmo

Desde que en 1986 se presentara la regla *Backpropagation*, se han desarrollado diferentes variantes del algoritmo original. Estas variantes tienen por objeto acelerar el proceso de aprendizaje. A continuación, se mencionan brevemente los algoritmos más relevantes, citados en [Palmer2001]

- La regla *delta-bar-delta* se basa en que cada peso tiene una tasa de aprendizaje propia, y ésta se puede ir modificando a lo largo del entrenamiento.
- Por su parte, el algoritmo QUICKPROP modifica los pesos en función del valor del gradiente actual y del gradiente pasado.
- El algoritmo de gradiente conjugado se basa en el cálculo de la segunda derivada del error con respecto a cada peso, y en obtener el cambio a realizar a partir de este valor y el de la derivada primera.
- Por último, el algoritmo RPROP –*Resilient propagation*– es un método de aprendizaje adaptativo parecido a la regla *delta-bar-delta*, donde los pesos se modifican en función del signo del gradiente, no en función de su magnitud.

Consideraciones prácticas

En cuanto al conjunto de entrenamiento

Cuando se introdujo la regla GDR prometimos dar precisiones sobre que significa *adecuadamente y suficiente* con respecto a la selección de pares de vectores de entrenamiento para la *Backpropagation*. Desafortunadamente, no hay una definición sencilla que sea aplicable a todos los casos. Al igual que muchos aspectos de los sistemas de redes neuronales, la experiencia suele ser la mejor maestra. Se darán aquí unas cuantas ideas generales.

Frecuentemente sólo se necesita un subconjunto pequeño de los datos de entrenamiento que se disponen. Por lo general el resto se utiliza para probar la red una vez concluido la etapa de aprendizaje.

En [Palmer2001] se sugiere crear tres conjuntos con los datos de entrenamiento, uno de aprendizaje, otro de validación y finalmente uno de test. La motivación es encontrar la red que tenga la mejor ejecución con casos nuevos, es decir, aquella mejor adquiera la capacidad de generalizar.

Durante la etapa de aprendizaje de la red, los pesos son modificados de forma iterativa de acuerdo con los valores del grupo de entrenamiento. Sin embargo, cuando el número de pesos es excesivo en relación al problema, el modelo se ajusta demasiado a las particularidades irrelevantes presentes en los patrones de entrenamiento en vez de ajustarse a la función subyacente que relaciona entradas y salidas, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos –fenómeno del sobreajuste–. En términos coloquiales la red aprende de memoria¹. Obsérvese que se puede disminuir el número de pesos en una arquitectura Backpropagation utilizando menos unidades ocultas. También se logra disminuyendo la cantidad de neuronas de entrada y salida, pero éstas son dependientes del problema, y sólo puede conseguirse cambiando la codificación de los parámetros de entrada y de la salida.

Para evitar el problema del sobreajuste, es aconsejable utilizar el segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de aprendizaje. Durante el aprendizaje la red va modificando los pesos en función de los datos de entrenamiento y de forma alternada se la alimenta con los datos de validación.

Con el grupo de validación se puede averiguar cuál es el número de pesos óptimo –y así evitar el problema del sobreajuste–, en función de la arquitectura que ha tenido la mejor ejecución con los datos de validación.

Por último, si se desea medir de una forma completamente objetiva la eficacia final del sistema construido, no deberíamos basarnos en el error que se comete ante los datos de validación, ya que de alguna forma, estos datos han participado en el proceso de entrenamiento. Se debería contar con un tercer grupo de datos independientes, el grupo de test el cuál proporcionará una estimación acertada del error de generalización.

En otro sentido, Freeman y Skapura [Freeman93] aconsejan no entrenar por completo a la red con vectores de una clase, pasando después a otra clase; la red se olvidará del entrenamiento original, una estrategia de selección aleatoria del orden en que se presentarán los datos de entrada, o el intercalado entre distintas clases pueden ser dos alternativas viables.

En cuanto a las dimensiones de la red

Otra decisión importante para implementar una Backpropagation será la de cuántas capas ocultas y cuántos nodos incluir en cada una de ellas. La cantidad de nodos de la capa de salida generalmente está fijada por el problema particular a resolver y depende si se desean valores analógicos o binarios en las unidades de salida.

¹ Esto me recuerda un cuento de Borges en el que su personaje, Ireneo Funes, luego de un accidente adquiere una increíble capacidad de memoria tan extraordinaria al punto de provocarle grandes dificultades intelectuales. Era capaz de recordar cada hoja de cada árbol y cada vez que la había percibido o imaginado.

“...No sólo le costaba comprender que el símbolo perro abarcara tantos individuos dispares de diversos tamaños y diversas formas; le molestaba que el perro de las tres y catorce (visto de perfil) tuviera el mismo nombre que el perro de las tres y cuarto (visto de frente).”

“...Sospecho, sin embargo, que no era muy capaz de pensar. Pensar es olvidar diferencias, es generalizar, abstraer. En el abarrotado mundo de Funes no había sino detalles, casi inmediatos.”

Jorge Luis Borges. Funes el memorioso.

Utilizar dos capas ocultas a menudo hace que la red aprenda más deprisa, pero habitualmente con una sola alcanza. Determinar el número de unidades que hay que utilizar en la capa oculta puede resultar no muy evidente. La idea consiste en utilizar el menor número posible para ahorrar carga de procesamiento. Para redes de un tamaño considerable –cientos o miles de entradas–, el tamaño de la capa oculta sólo necesita ser una fracción relativamente pequeña del que tiene la capa de entrada. Si la red no llega a converger para llegar a una solución, cabe en lo posible que se necesiten más nodos ocultos. Si converge, se puede probar con un número inferior de nodos ocultos y determinar un tamaño final basándose en el rendimiento global del sistema.

Es posible eliminar unidades ocultas que resulten superfluas. Si se examinan periódicamente los valores de los pesos de las neuronas de la capa oculta durante el entrenamiento, a menudo se detectan pesos que cambian muy poco respecto de sus valores iniciales. Estos nodos pueden no estar participando del aprendizaje y quizá baste con un número menor de unidades ocultas.

En cuanto a los pesos y parámetros de aprendizaje

Los pesos iniciales deberían ser pequeños y aleatorios, por ejemplo entre $-0,5$ y $+0,5$. Más problemático suele ser la elección del parámetro de velocidad de aprendizaje, μ , ya que tiene un efecto significativo en el rendimiento de la red. Normalmente μ debe ser un número pequeño –del orden de $0,05$ a $0,25$ – para asegurar que la red llegue a asentarse en una solución. Suele ser posible incrementar el valor de μ a medida que progresa el aprendizaje. Aumentando a medida que disminuye el error de la red suele acelerar la convergencia incrementando el tamaño del paso conforme el error alcanza un valor mínimo, pero la red puede rebotar, alejándose demasiado del valor mínimo verdadero, si μ llegara a ser demasiado grande.

Otra forma de aumentar la velocidad de convergencia consiste en utilizar una técnica llamada momento. Cuando se calcula el valor del cambio de peso, se añade una fracción del cambio anterior. Este término adicional tiende a mantener los cambios de peso en la misma dirección. Las ecuaciones de cambio de pesos de la capa de salida pasan entonces a ser:

$$w_{ki}(t+1) = w_{ki}(t) + \mu(d_k - y_k) f'(a) \delta_i + \alpha \Delta_{ki}(t-1)$$

con una ecuación similar para la capa oculta. En la ecuación anterior, α es el parámetro de momento, y permite filtrar las oscilaciones en la superficie del error provocadas por la tasa de aprendizaje μ al acercarse al mínimo, y acelera considerablemente la convergencia de los pesos, ya que si en el momento t el incremento de un peso era positivo y en $t+1$ también, entonces el descenso por la superficie de error en $t+1$ será mayor. Sin embargo, si en t el incremento era positivo y en $t+1$ es negativo, el paso que se da en $t+1$ es más pequeño, lo cual es adecuado, ya que eso significa que se ha pasado por un mínimo y los pasos deben ser menores para poder alcanzarlo. El factor momento suele tomar un valor próximo a 1

En cuanto a la función de transferencia de las neuronas ocultas y de salida

El algoritmo *Backpropagation* exige que la función de transferencia sea derivable para poder obtener el error o valor delta de las neuronas ocultas y de salida. Se disponen de dos formas básicas que cumplen esta condición: la función lineal Identidad y la función sigmoide –logística o tangente hiperbólica–.

Debe tenerse en cuenta que para aprovechar la capacidad de las RNA de aprender relaciones complejas o no lineales entre variables, es imprescindible la utilización de funciones no lineales al menos en las neuronas de la capa oculta [Palmer2001]. Las RNA que no utilizan funciones no lineales, se limitan a solucionar tareas de aprendizaje que implican únicamente

funciones lineales o problemas de clasificación que son linealmente separables. Por tanto, en general se utilizará la función sigmoide –logística o tangente hiperbólica– como función de transferencia en las neuronas de la capa oculta.

Por su parte, la elección de la función de transferencia en las neuronas de la capa de salida dependerá del tipo de tarea impuesta. En tareas de clasificación, las neuronas normalmente toman la función de activación sigmoide. Así, cuando se presenta un patrón que pertenece a una categoría particular, los valores de salida tienden a dar como valor 1 para la neurona de salida que representa la categoría de pertenencia del patrón, y 0 ó -1 para las otras neuronas de salida. En cambio, en tareas de predicción o aproximación de una función, generalmente las neuronas toman la función de activación lineal.

Debilidades de la Backpropagation

A continuación se detallarán algunas debilidades o dificultades concernientes a las redes Backpropagation, muchas de ellas recopiladas en [GómezMartínez97]

- Los resultados dependen fuertemente de los valores iniciales aleatorios de las conexiones. Esto hace que sea conveniente entrenar varias redes con distintos valores iniciales y elegir la que mejor funcione.
- A veces se requiere mucho tiempo para obtener soluciones sencillas. Las redes más grandes necesitan más tiempo de procesamiento, pero también hacen falta más ejemplos de aprendizaje, y eso provoca un aumento mucho mayor del tiempo de aprendizaje.
- Incorporar nuevos ejemplos de aprendizaje, una vez que la red haya sido entrenada produce una “interferencia catastrófica” o empeoramiento en el rendimiento del sistema.
- Inestabilidad temporal: Si usamos un coeficiente de aprendizaje elevado, se van a producir incrementos grandes en los pesos, de manera que es fácil pasarse de incremento y tener que tratar de compensarlo en el siguiente ciclo, de manera que se producirían oscilaciones continuas. Esto se soluciona usando un coeficiente pequeño, o para no tener un aprendizaje muy lento, modificar dicho coeficiente adaptativamente –aumentarlo si el error global disminuye, y disminuirlo en caso contrario– como ya comentamos en el apartado anterior.
- El problema de los mínimos locales: El algoritmo de Backpropagation sigue la pendiente más pronunciada hacia abajo en la superficie del error, pero no garantiza alcanzar el mínimo global.

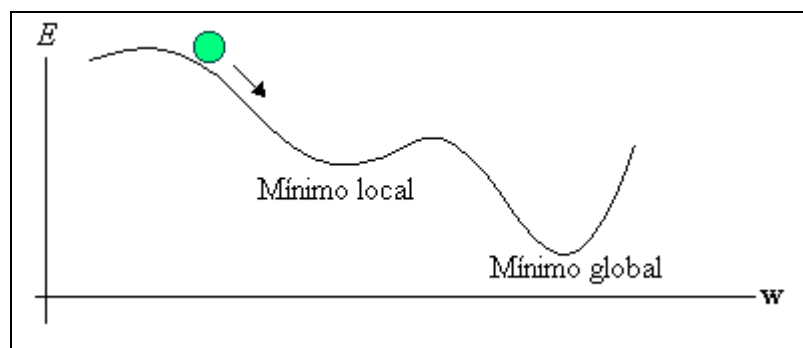


Ilustración 72 El problema de los mínimos locales. Esta gráfica muestra una sección transversal de una superficie de error hipotética dentro del espacio de pesos. A partir de la posición aleatoria inicial en la superficie del error, el algoritmo nos llevará probablemente al mínimo más cercano, pero no necesariamente al más *hondo* –mínimo global–.

Una vez que la red se asienta en un mínimo, sea local o global, cesa el aprendizaje. Si se alcanza un mínimo local, el error de la red puede seguir siendo alto. Afortunadamente, este problema no parece causar grandes dificultades en la práctica. De todas maneras podemos intentar solucionarlo. Una técnica que puede ayudar a no caer en mínimos locales consisten en añadir cierto nivel de ruido a las modificaciones de los pesos de las conexiones. Otra alternativa es simplemente volver a empezar con un conjunto distinto de pesos originales. Como contra, estas medidas aumentan el tiempo de aprendizaje.

Hasta aquí, los ejemplos puntuales de redes neuronales artificiales. Ahora retomaré la perspectiva más genérica para completar la caracterización de las RNA que se inició tempranamente en el presente capítulo. Ya tenemos nociones precisas acerca de cómo “aprende” una red a través del entrenamiento o como se organizan y conectan los nodos para formar una arquitectura característica. Si bien se han visto estas propiedades con cierto detalle para el Perceptrón, Adaline y Backpropagation, lo cierto es que existen muchas otras arquitecturas –topologías– y distintos tipos de aprendizajes que no se verán en detalle aquí. Esto no impide el tratamiento general de aquellos temas que completarán una visión más acabada de este paradigma, justo antes de abordar el interesante resultado de su combinación con algoritmos evolucionarios, conformando lo que se llama *neuroevolución* que prácticamente ya se está en condiciones de presentar.

Arquitectura neuronal

Ya se ha visto como conectando varias neuronas de un determinado modo se consigue una red neuronal completa. La manera de realizar esas conexiones determina la arquitectura de la misma. Existen muchas variaciones de topologías, pero pueden clasificarse según ciertos criterios como el número de niveles o capas, el número de neuronas por nivel, o las formas de conexión.

En cuanto a la cantidad de capas, puede decirse que una red podrá estar formada por un mínimo de una capa y un máximo de n capas¹. El número de neuronas por nivel puede variar desde 1 hasta n y depende de la topología de cada modelo en particular.

La forma en que la salida de una determinada capa de neuronas está conectada a la entrada de otra, determina la conectividad de la red. Aquí existen dos formas básicas:

- *Redes Feedforward*: Propagación hacia delante. Las salidas de las neuronas de una capa sólo se propagan hacia la capa siguiente. Un ejemplo de este tipo de redes es la Backpropagation.
- *Redes FeedBack*: Propagación hacia atrás. Son redes donde existe algún tipo de realimentación de las salidas a las entradas anteriores o del mismo nivel. También suelen denominarse *redes recurrentes o recursivas*.

¹ Ya se ha mencionado, y queremos ratificarlo, que en la bibliografía consultada no existe un criterio unificado para contar los números de capas de una red. Algunos autores consideran los datos de entrada como la primera capa, que si bien no realiza procesamiento alguno transfiere los datos al siguiente nivel. Otros no la consideran una capa propiamente dicha por no participar ni del procesamiento ni del aprendizaje.

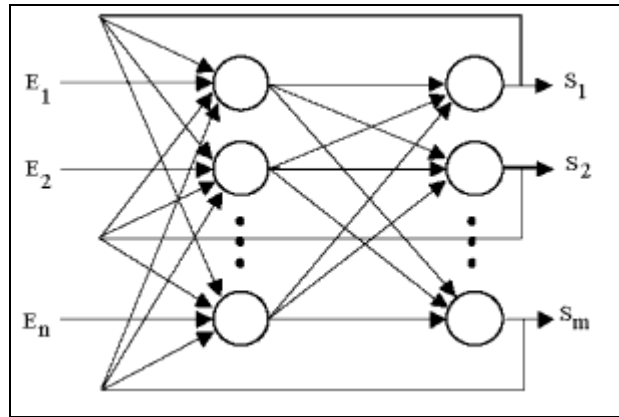


Ilustración 73 Ejemplo de red recurrente.

Las redes feedforward suelen ser más rápidas, en contraste las recurrentes son más potentes porque en el sistema circulan datos relacionados con entradas presentadas a la red en instantes anteriores. Esto le da a estas redes la capacidad de *memoria de corto plazo*.

Cada una de las características enumeradas no dice demasiado por sí sola, por lo que debe analizarse junto al modelo de red que la adopta, sin embargo, dicho análisis excede los objetivos del presente capítulo.

Según el número de conexiones se pueden clasificar las redes neuronales en dos categorías

- *Redes totalmente conectadas*: Cuando la salida de las neuronas de una capa se conecta con todos los PEs de la capa siguiente.
- *Redes localmente conectadas*: Cuando la salida de las neuronas de una capa se conecta sólo con un cierto número de PEs –una región– de la capa siguiente

Tipos de aprendizaje

En cualquier red neuronal tradicional¹ cabe distinguir la fase o proceso de aprendizaje, opcionalmente una fase de test, y la fase de aplicación.

La modificación dinámica de los pesos de las conexiones es la verdadera esencia del aprendizaje en una red neuronal.

Podría aproximarse una definición de aprendizaje diciendo que es un proceso por el cual los parámetros libres de una red neuronal son ajustados a través de un proceso continuo de estimulación por parte del entorno en donde se sitúa el sistema. El tipo de aprendizaje es determinado por la forma o manera que tienen lugar dichos cambios.

El modo de aprendizaje más sencillo consiste en la presentación de patrones de entrada junto a los patrones de salida deseados para cada patrón de entrada, por eso se llama *aprendizaje supervisado*. A su vez el aprendizaje supervisado puede ser de tres subtipos: *por Corrección de Error*, *por Refuerzo* y *Estocástico*.

¹ Se utilizará el término *tradicional*, para distinguir las redes neuronales que estamos estudiando de aquellas que se generan a través de un proceso de evolución a las que llamaremos *redes neuronales evolutivas* y pertenecen al paradigma *neuroevolución* que veremos más adelante.

Si no se le presentan a la red los patrones de salida deseados, se dice que se trata de *aprendizaje no supervisado*, ya que no se le indica a la red cuáles son los resultados que debe dar, sino que se le deja seguir alguna regla de autoorganización.

Aprendizaje Supervisado por corrección de error

Este es el aprendizaje supervisado por excelencia, a tal punto que en mucha bibliografía sólo se usa el término aprendizaje supervisado. Debe entenderse en tal caso que se está haciendo referencia al aprendizaje por corrección de error.

Los ejemplos visitados: Perceptrón, Adaline y Backpropagation utilizan este tipo de aprendizaje. Se presentan a la red una serie de patrones de entrada junto a los patrones de salida deseados. El aprendizaje consiste en la modificación de los pesos de las conexiones en el sentido de reducir la discrepancia entre la salida obtenida y la deseada.

En los años 80 se popularizaron el algoritmo de Backpropagation y la regla delta generalizada que ya se ha expuesto en el presente capítulo. La mayoría de las reglas empleadas actualmente en el aprendizaje supervisado por corrección de error son modificaciones de la regla delta generalizada. Se calcula que el 80% de las aplicaciones tecnológicas en redes neuronales se fundamentan en el algoritmo de Backpropagation y en alguna versión de la regla delta generalizada.

Aprendizaje Supervisado por Refuerzo

Es más lento que el anterior, pero tiene la ventaja de no necesitar conocer por completo el comportamiento deseado, es decir, puede que no se conozca la salida deseada exacta para cada entrada. Lo que se conoce es cómo debería ser el comportamiento de manera general ante diferentes entradas.

La función del supervisor aquí es más la de un crítico que la de un maestro. Consiste en poner en funcionamiento a la red y evaluar el éxito o fracaso de la salida, se produce entonces una señal llamada de refuerzo que mide el buen funcionamiento del sistema. Esta señal se caracteriza por el hecho de que es menos informativa que en el caso de aprendizaje supervisado por corrección de error. Simplemente puede ser una indicación de que el dispositivo funcionó correctamente o no.

Los pesos se ajustan en base a la señal de refuerzo basándose en un mecanismo de probabilidades. Si una acción tomada por el sistema de aprendizaje es seguida por un estado satisfactorio, entonces la tendencia del sistema a producir esa particular acción es reforzada. En otro caso, la tendencia del sistema a producir dicha acción es disminuida.

Debe aclararse que existe bibliografía que sitúa al aprendizaje por refuerzo a medio camino entre el supervisado y el no supervisado. De hecho, la mayor parte de la bibliografía especializada en neuroevolución consultada en este trabajo hablan simplemente de aprendizaje supervisado, en alusión al aprendizaje supervisado por corrección de error, y de aprendizaje por refuerzo, para referirse al aprendizaje supervisado por refuerzo.

Aprendizaje Supervisado Estocástico

Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidades.

Utiliza el paradigma *Solidificación Simulada –Simulated Annealing–* que se presentó austeramente en otro capítulo –pág. 81– y del cual no se darán más precisiones.

El estado de mínima energía corresponde a los valores de los pesos con los que la red se ajusta al objetivo deseado, es decir que la energía representa una medida del error producido. El proceso es el siguiente:

- Se realiza un cambio aleatorio en los pesos.
- Se determina la nueva energía de la red alguna consideración sobre la distancia al objetivo deseado.
- Si la energía decrece se acepta el cambio, si no decrece se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades.

Dos modelos de redes que han utilizado este tipo de aprendizaje son *Boltzmann Machine* y *Cauchy Machine*.

Aprendizaje no supervisado

Los investigadores han diseñado varios procedimientos de aprendizaje no supervisado, que son capaces de extraer inductivamente suficiente información de los datos presentados a la red neuronal como para permitir posteriormente su recuperación o clasificación de acuerdo a características similares.

En el aprendizaje no supervisado no se requiere presentar patrones de salida deseados. Sin embargo el algoritmo y la regla de modificación de las conexiones produce vectores de salida consistentes, esto es, la presentación de un patrón aprendido o parecido a el, produce siempre la misma salida. El proceso de aprendizaje extrae las propiedades estadísticas de los ejemplos de aprendizaje y los agrupa en clases de patrones similares. No se sabe a priori que salida corresponderá a cada grupo o categoría de patrones de entrada.

Existen diferentes interpretaciones que se le puede dar a la salida generada por una red utilizando este tipo de aprendizaje y que depende de la estructura y el algoritmo. Por ejemplo:

- *Grado de Familiaridad o Similitud*: Entre la información actual y la información pasada.
- *Clusterización*: Establecimiento de categorías o clases. La red se encarga de encontrar las características o propiedades propias de cada clase.
- *Codificación*: La salida se interpreta como una codificación de la entrada.
- *Mapeo de Características*: Los elementos de procesamiento de la capa de salida se disponen geoméricamente representando un mapa topográfico de las características de los datos de entrada. Es decir a entradas parecidas le corresponde la activación de neuronas próximas en la capa de salida.

Un ejemplo de red que utiliza este tipo de aprendizaje, cuya interpretación de la salida es un mapeo de características lo constituye la red SOM –Self organizing map– de Kohonen.

Un concepto asociado al aprendizaje no supervisado es el *aprendizaje competitivo*.

Aprendizaje Competitivo

En el aprendizaje competitivo, un gran número de unidades ocultas pugnan entre sí, con lo que finalmente se utiliza una sola unidad oculta para representar un patrón de entrada determinado o un conjunto de patrones similares. La unidad oculta seleccionada es aquella cuyos pesos de conexión se asemejan más al patrón de entrada. El aprendizaje consiste en reforzar las conexiones de la unidad ganadora y debilitar las otras, para que los pesos de la unidad ganadora se asemejen cada vez más al patrón de entrada.

Modelos de RNA

Como se ha mencionado anteriormente, existen varios modelos diferentes de redes neuronales. Cada una de ellos ha sido diseñado para fines más o menos específicos, no obstante, existen varios que han ganado gran popularidad. Se presenta a continuación una tabla con los modelos más conocidos junto a sus características relevantes [Hilera95].

Nombre	Año	Aplicaciones	Comentarios	Limitaciones	Creador/es
Perceptrón	1957	Reconocimiento de caracteres impresos	La red más antigua	No puede reconocer caracteres complejos	Frank Rosenblatt
ADALINE/M ADALINE	1960	Filtrado de señales. Ecuador adaptativo. Modems.	Rápida, fácil de implementar con circuitos analógicos	Solo es posible clasificar espacios linealmente separados.	Bernard Widrow
Avalancha	1967	Reconocimiento de habla continua. Control de robots.	Ninguna red sencilla realiza estas tareas	No es fácil alterar la velocidad o interpolar el movimiento	Stephen Grossberg
Cerebellatrón	1969	Control del movimiento de los brazos de un robot	Semejante a Avalancha	Requiere complicadas entradas de control	David Marr, James Albus, Andres Pellionez
Backpropagation	1974-1985	Síntesis de voz desde texto. Control de robots. Predicción	La red más popular. Numerosas aplicaciones con éxito. Facilidad de aprendizaje. Potente.	Necesita mucho tiempo para el aprendizaje y muchos ejemplos	Paul Werbos, David Parker, David Rumelhart
Brain-State-in-a-Box	1977	Extracción de conocimiento de bases de datos	Posiblemente mejor realización que las redes de Hopfield	Realización y potenciales aplicaciones no estudiadas totalmente	James Anderson
Neocognitrón	1978-1984	Reconocimiento de caracteres manuscritos	Insensible a la translación, rotación y escala	Requiere muchos elementos de proceso, niveles y conexiones.	K. Fukushima
Self Organizing Map (SOM)	1980-1984	Reconocimiento de patrones, codificación de datos, optimización	Realiza mapas de características comunes de los datos aprendidos	Requiere mucho entrenamiento	Teuvo Kohonen
Hopfield	1982	Reconstrucción de patrones y optimización	Puede implementarse en VLSI. Fácil de conceptualizar	Capacidad y estabilidad	John Hopfield
Memoria asociativa bidireccional	1985	Memoria heteroasociativa de acceso por contenido	Aprendizaje y arquitectura simples	Baja capacidad de almacenamiento. Los datos deben ser codificados	Bart Kosko
Máquinas de Boltzman y Cauchy	1985-1986	Reconocimiento de patrones (imágenes, sonar y radar). Optimización	Redes simples. Capacidad de representación óptima de patrones	La máquina de Boltzman necesita un tiempo muy largo de aprendizaje	Jeffrey Hinton, Terry Sejnowski
Counter propagation	1986	Compresión de imágenes.	Combinación de Perceptrón y TPM	Numerosas neuronas y conexiones	Robert Hecht-Nielsen
Teoría de resonancia adaptativa (ART)	1986	Reconocimiento de patrones (radar, sonar, etc)	Sofisticada, poco utilizada	Sensible a la translación, distorsión y escala	Gail Carpenter, Stephen Grossberg

Modelos de Redes [Hilera95]

Aplicaciones de RNA

En la tabla anterior pueden observarse varias aplicaciones en las que se han utilizado con éxito las redes neuronales. No obstante, antes de presentar las redes neuronales artificiales evolucionarias –próximo tema a abordar– realizaré una pequeña enumeración de algunos logros obtenidos por el paradigma de las RNA.

- *Conversión de texto escrito a lenguaje hablado*: Se han desarrollado redes neuronales que transforman el texto escrito en los códigos elegidos para representar los fonemas correspondientes. Mediante la ayuda de un sintetizador se transforman los códigos en fonemas. La red aprende a hacer distinciones difíciles como pronunciar una *c* suave o fuerte según el contexto. Si bien esto se había conseguido antes, la novedad más importante reside en que no es necesario definir y programar muchas reglas complejas, pues la red extrae automáticamente el conocimiento necesario.
- *Aprendizaje de gramáticas*: Se han utilizado también de manera exitosa redes neuronales para aprender el pasado de los verbos ingleses. A través del entrenamiento el sistema fue mejorando y al final fue capaz de generalizar y conjugar verbos desconocidos.
- *Compresión de imágenes*: Utilizando redes neuronales se han conseguido codificar imágenes con una relación de compresión de hasta 8:1 con alta fidelidad en la reconstrucción sin tener que idear ninguna regla.
- *Reconocimiento de escritura manual*: Se han empleado redes neuronales por ejemplo para reconocer kanji –escritura japonesa–, eliminando la gran dificultad que presenta este lenguaje para introducirlo en la computadora. El Neocognitrón, por ejemplo, consigue un reconocimiento muy avanzado de patrones con gran capacidad de abstracción y generalización, que lo hacen capaz de reconocer patrones con distinta orientación y altos niveles de distorsión.
- *Problemas de combinatoria*: Las redes neuronales artificiales están ofreciendo ciertas esperanzas en el área de problemas algorítmicamente tan complejos como los NP-completos; por ejemplo el problema del viajante de comercio –Hopfield, J. & Tank, D.–
- *Reconocimiento de patrones en imágenes*: En este campo se han desarrollado numerosas aplicaciones como la clasificación de imágenes de sonar y radar, la detección de células cancerosas, lesiones neurológicas y cardíacas, prospecciones geológicas, etc. Son muy útiles para procesar imágenes de las que no se sabe bien cuáles son las características esenciales o diferenciales, ya que las redes no necesitan disponer de reglas explícitas previas para realizar la clasificación, sino que extraen el conocimiento necesario.
- *Visión artificial en robots industriales*: Se han utilizado redes neuronales para inspección de etiquetas, clasificación de componentes, etc. Supera a otros sistemas de visión, además minimiza los requerimientos de operadores y facilita el mantenimiento.
- *Predicción de señales*: Se han obtenido mejores resultados a la hora de predecir series “caóticas” usando Backpropagation que mediante métodos lineales y polinomiales.
- *Filtro de ruido*: Las redes neuronales artificiales son mejores preservando la estructura profunda y el detalle que los filtros tradicionales cuando eliminan el ruido. El Adaline que vimos con cierto detalle en este capítulo constituyó la primera aplicación profesional de las redes neuronales y se utilizó como filtro para eliminar ruido en las líneas telefónicas.
- *Modelado y predicción de indicadores económicos*: Se obtienen mejores resultados que con cualquier otro método conocido. Se ha aplicado por ejemplo a la predicción de tasas de interés, déficits comerciales, precios de stock, etc.

- *Servocontrol*: Compensación adaptativa de variaciones físicas en servomecanismos complicados como el control de ángulos y posiciones de los brazos de un robot.

Aquí se termina este acercamiento al paradigma tradicional de las Redes Neuronales Artificiales. Creo que se han expuesto características importantes que sugieren grandes potencialidades. También se han mencionado algunas de sus limitaciones. Y son precisamente estas potencialidades y limitaciones las que convierten a las Redes Neuronales Artificiales en atractivos objetos de investigación.

Ultimamente ha surgido una alternativa atractiva para resolver algunas dificultades que supone la utilización de las RNA y que consiste en codificarlas dentro de un cromosoma para someterlas a un proceso de evolución. A continuación expondré los lineamientos principales de esta interesante área de investigación.

Redes Neuronales Artificiales Evolucionarias –RNAE–

Las redes neuronales artificiales evolucionarias –RNAE– son una clase especial de RNA donde la evolución representa la forma fundamental de adaptación, aunque se concibe también aprendizaje con entrenamiento tradicional como los que se han visto en este capítulo.

Ya se explicó anteriormente que la combinación de las RNA con algoritmos evolucionarios definen el paradigma que se acostumbra denominar *Neuroevolución*. Nótese que no utilicé el término Algoritmos Genéticos, ya que estos últimos son sólo una rama de la Computación Evolutiva, y las RNAE no se limitan sólo a ellos. El término algoritmos evolucionarios lo emplearé para hablar de manera general sobre cualquier proceso de evolución enmarcado en la Computación Evolutiva.

El presente trabajo corresponde a un desarrollo de Neuroevolución, en el cual se combinaron redes neuronales artificiales con Algoritmos Genéticos. Aquí sí me refiero a esta clase particular de algoritmos. Se ha explicado en el capítulo titulado “Computación Evolutiva” que dichos algoritmos ocupan un papel importante en esta rama de la Inteligencia Artificial subsimbólica –puede consultarse el apartado “Los Algoritmos Genéticos y su rol en la Computación Evolutiva” en la pág. 78 para más información al respecto– .

¿Por qué evolucionar redes neuronales?

Las redes neuronales artificiales son exitosas y han resuelto adecuadamente muchos problemas, pero presentan ciertas dificultades. Entre algunas otras citaremos:

- Tienen limitaciones en su capacidad de aprendizaje.
- No siempre es posible encontrar la topología adecuada.
- No es simple determinar las características adecuadas de la entrada.
- Existen muchos algoritmos de entrenamientos basados en métodos de gradiente y éste a veces es incalculable o muy costoso de hacerlo.
- Para utilizar estos algoritmos basados en el gradiente deben imponerse restricciones sobre la arquitectura de la red –funciones de transferencia y de error derivables–.
- Para asegurar un buen desempeño, la velocidad de aprendizaje deber ser pequeña –entrenamientos prolongados–.

- La utilización de parámetros de velocidad de aprendizaje más grandes –para acelerar el tiempo requerido del entrenamiento– puede llevarlas a un estado de inestabilidad con oscilaciones continuas, lo que degrada el aprendizaje.
- Son proclives a ser “engañadas” cayendo en mínimos locales.
- No es fácil saber a priori –salvo recurriendo a la experiencia– cuántas capas y unidades ocultas son necesarias para resolver adecuadamente el problema.
- Poseen el inconveniente del sobreajuste –ya explicado anteriormente–.
- Generalmente poseen fuerte dependencia con los valores iniciales de los pesos escogidos aleatoriamente.
- Frecuentemente el entrenamiento requiere mucho tiempo de procesamiento.

Todas estas dificultades han llevado a los investigadores al estudio de alternativas para paliar algunos de estos inconvenientes. Así se han propuestos muchísimas estrategias para combinar el poder de las RNA con el de los algoritmos evolucionarios resolviendo satisfactoriamente muchos de los inconvenientes mencionados.

La Computación Evolutiva constituye un método de búsqueda estocástico que no tiene problemas de selección de paso, es decir parámetro de velocidad, ni de mínimos locales, y tiene la característica deseable de trabajar en paralelo. Estas cualidades hacen de este paradigma una buena opción para el entrenamiento de redes neuronales, pues al no utilizar información de gradiente, los algoritmos evolucionarios son en general más rápidos, más simples de implementar y tienen menos probabilidad de caer en un mínimo local que los algoritmos de entrenamiento tradicionales.

La evolución se ha utilizado en diversas formas para conseguir establecer varios aspectos de las redes neuronales artificiales, tales como los pesos de conexión, el diseño de la arquitectura, el valor de los parámetros iniciales, las reglas de aprendizaje, etc.

Una característica distinguible de las RNAE es su capacidad de adaptabilidad a un ambiente dinámico [XinYao99]. Las RNAE pueden adaptarse a un ambiente a medida que éste va cambiando. Mientras que las RNA tienen una única forma de adaptación –aprendizaje–, las RNAE poseen dos formas de adaptación –evolución y aprendizaje– ya que se pueden evolucionar redes que luego sean sometidas a un entrenamiento tradicional, esto sucede por ejemplo cuando se evoluciona la inicialización de los parámetros de la red, la idea es obtener la mejor inicialización para que el entrenamiento sea más eficiente¹.

El contar con dos formas de adaptación hace que las RNAE sean mucho más efectivas y eficientes que las RNA para adaptarse a un ambiente dinámico sin intervención humana.

A continuación haré un breve recuento de distintas formas de neuroevolución.

Evolución de pesos de conexión

Ya he mencionado los inconvenientes de la utilización del gradiente en los algoritmos de entrenamiento. Una forma de evitar estas dificultades consiste en prescindir de estos algoritmos. En su lugar, se puede recurrir a los algoritmos evolucionarios para encontrar el

¹ Esto constituye una forma de hibridación de técnicas de búsqueda, donde la evolución realiza la tarea exploratoria y un algoritmo de aprendizaje tradicional realiza la tarea explotatoria. Ya se ha realizado una discusión sobre la exploración vs. la explotación en el contexto de los algoritmos genéticos –para su relectura diríjase a la página 74–.

conjunto de pesos de conexión adecuados que haga funcionar correctamente a la red en la realización de la tarea pretendida.

Para una arquitectura fija predefinida, se codifican los pesos de las conexiones en un cromosoma, se arma la población y se somete a evolución. Para calcular el fitness de los individuos se construyen las redes respectivas con los pesos de conexión determinados en los cromosomas y se prueba la misma para obtener una calificación acerca de su desempeño. Por ejemplo el fitness de la red podría ser una función inversamente proporcional al error cuadrático medio calculado con un conjunto de prueba. Debe quedar claro que al probar la red no se realiza modificación de pesos alguna, sólo se calcula el error para poder asignarle un valor de fitness.

Debido a que sólo se somete a evolución el conjunto de pesos de conexión quedando todos los otros parámetros de la red fijos, a esta estrategia se la denomina *entrenamiento evolucionario* pues hace lo mismo que cualquier algoritmo de entrenamiento tradicional. No obstante posee algunas ventajas pues no necesita información de gradiente, no está limitado a ningún tipo de arquitectura en particular, tiene menos probabilidad de caer en un óptimo local y se puede utilizar un algoritmo genético simple sin ningún esfuerzo adicional.

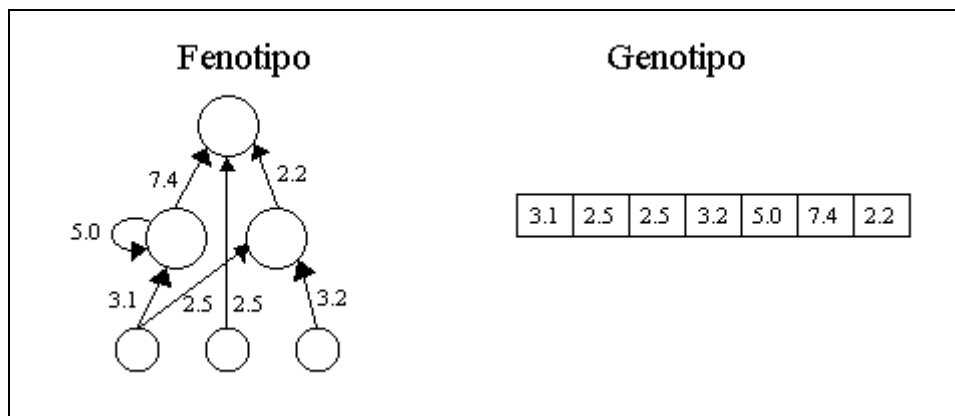


Ilustración 74 Entrenamiento Evolucionario. Sólo se somete a proceso evolutivo el conjunto de pesos de conexión. Obsérvese que la arquitectura de la red –fijada previamente– corresponde a una topología arbitraria, el método no asume ninguna restricción sobre la misma.

En cuanto a la codificación elegida para representar en el cromosoma los valores reales de los pesos de conexión valen las mismas consideraciones que discutimos al ver los Algoritmos Genéticos en otro capítulo –ver “Codificación” en la pág. 65–.

Si se pretende respetar el Algoritmo Genético canónico se utilizarán strings binarios, así cada peso será representado por un número determinado de bits. La red neuronal quedará codificada simplemente por la concatenación de los pesos de conexión –recordemos que la arquitectura de la red es fija–. Algunos investigadores han utilizado los códigos de Gray –ver en pág. 67– con buenos resultados. Otros han preferido utilizar un alfabeto de mayor cardinalidad proponiendo la utilización de números reales en lugar de binarios para codificar los cromosomas. El ejemplo de la Ilustración 74 justamente utiliza esta última representación. Recordemos que esta discusión excede el ámbito de la Neuroevolución perteneciéndole al paradigma más amplio de la Computación Evolutiva –para más información ver el apartado “Codificando números reales” en la pág. 68–.

Entrenamiento evolucionario vs. entrenamiento basado en gradiente

Ya se indicó que el entrenamiento evolucionario es atractivo pues puede manejar el problema de la búsqueda global mejor en superficies del error complejas, multimodales y no diferenciables. Por lo tanto es preferible cuando la información del gradiente es inexistente, o costosa de conseguir o estimar. Existen numerosos ejemplos donde se ha utilizado una aproximación evolucionaria para entrenar redes recurrentes de alto orden. Además el mismo algoritmo puede ser usado para entrenar cualquier tipo de red sin tener que considerar si ellas son feedforward, recurrentes o de alto orden.

El entrenamiento evolucionario puede ser más lento para algunos problemas en comparación con las más rápidas variantes del algoritmo Backpropagation y el algoritmo del gradiente conjugado. Sin embargo los algoritmos evolucionario son menos sensibles a las condiciones iniciales de entrenamiento. Ellos siempre buscan soluciones óptimas globales, mientras que los algoritmos de gradiente descendente pueden sólo buscar una solución óptima local en la vecindad de la solución inicial.

Existe un conjunto grande de problemas en donde el entrenamiento evolucionario puede ser significativamente más rápido y confiable que el Backpropagation. Estos casos han sido estudiados y expuestos por varios investigadores entre ellos Prados, Bartlett, y Downs [XinYao99]. Sin embargo, otros resultados bastante distintos fueron presentados por Kitano que encontró que el método AG-BP –una técnica híbrida que corre un Algoritmo Genético primero y luego un Backpropagation– es tan bueno como las variantes rápidas de Backpropagation para redes pequeñas, pero menos eficiente en grandes redes. Sin embargo ha habido varios otros artículos que reportan excelentes resultados usando hibridación de entrenamiento evolucionario y algoritmos de gradiente descendente.

Esta discrepancia puede ser atribuida, al menos parcialmente, a que han sido comparados distintos algoritmos evolucionarios y distintas variantes de Backpropagation. Pero esta diferencia también señala que no existe todavía un claro ganador en términos del mejor algoritmo de entrenamiento, sino que sus desempeños se ven afectados por la naturaleza del problema. Esto es ciertamente verdadero de acuerdo al teorema de *No-Free-Lunch*¹. En general los algoritmos híbridos tienden a realizar mejor la tarea para un gran número de casos.

Una característica de las RNA que puede degradar el entrenamiento evolucionario es la equivalencia funcional de distintas redes. Por ejemplo: si alteramos el orden de las neuronas de la capa oculta de una red multicapa feedforward totalmente conectada, aunque distintas, producirán los mismo resultados. A esto se lo denomina *problema de permutación* y causa el hecho indeseable que genotipos distintos produzcan fenotipos que funcionan de igual forma. Según Xin Xao en [XinYao99] el problema de permutación disminuye la eficiencia del crossover.

En un intento de reducir el impacto negativo del problema de permutación, varios investigadores han abandonado el crossover como operador genético utilizando esencialmente el operador de mutación o alguna variante –evolución primitiva–. Así se han apartado de los Algoritmos Genéticos que tienen al crossover como principal operador. Codificando los pesos de conexión en el cromosoma como valores reales, estos trabajos se enmarcan en la

¹ Este teorema desarrollado por Wolpert y Macready (1995; 1997) ataca la supuesta robustez de los Algoritmos Genéticos, indicando la necesidad de introducir conocimiento del problema dentro del algoritmo de búsqueda para mejorar su rendimiento (ver Críticas a la Base Teórica de los Algoritmos Genéticos –pág. 63–).

rama denominada Estrategias Evolutivas –ver “Otros Paradigmas de la Computación Evolutiva” en la pág. 78–.

Entrenamiento híbrido

Para una función unimodal, es decir que posea un solo extremo –máximo o mínimo según se trate de una función de fitness o de una función de error–, la mayoría de los algoritmos evolucionarios no son tan eficientes como los que usan el método del gradiente descendiente, para encontrar rápidamente el punto óptimo.

Sin embargo en una función complicada, multimodal, donde existen muchos máximos o mínimos locales, los algoritmos evolucionarios realizan una mejor búsqueda global. Esto es especialmente verdadero para los Algoritmos Genéticos. Por lo tanto, la eficiencia del entrenamiento evolucionario puede ser mejorada significativamente incorporando procedimientos de búsqueda local dentro de la evolución. Así los algoritmos evolucionarios encontrarán una buena región en el espacio para que un procedimiento de búsqueda local halle la mejor solución en dicha región. Este algoritmo local podría ser por ejemplo el algoritmo Backpropagation si la arquitectura de la red lo permite.

Muchos investigadores han utilizados Algoritmos Genéticos para buscar un conjunto de pesos de conexión cercano al óptimo, una vez hallada esta región, un algoritmo Backpropagation completa la tarea realizando una búsqueda local desde esos pesos iniciales. Sus resultados muestran que la hibridación AG-BP es más eficiente que cualquiera de los métodos de búsqueda por separado. También se han realizado similares trabajos de evolución de pesos iniciales en redes neuronales de aprendizaje competitivo y redes de Kohonen.

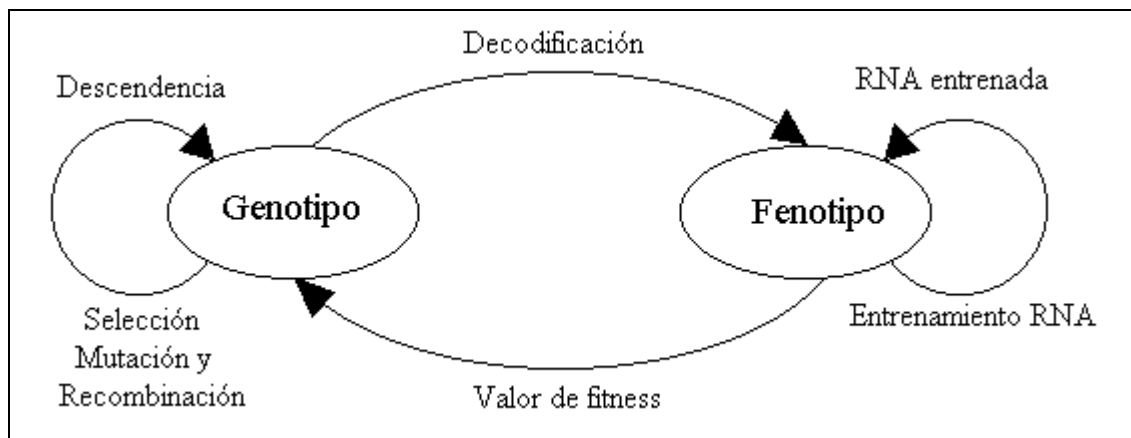


Ilustración 75 Esquema de un de entrenamiento híbrido.

Otra vez se está cuestionando la robustez de los Algoritmos Genéticos al presentar mejores resultados con técnicas híbridas, donde es necesario incluir información específica del problema para mejorar la solución.

Evolución de arquitectura

Aquí se discute el diseño de arquitecturas de RNA. La arquitectura incluye su topología –conectividad y función de transferencia de cada nodo–. La arquitectura es crucial puesto que posee gran impacto sobre las capacidades de procesamiento de la red. Dada una tarea de aprendizaje una red neuronal con solo unas pocas conexiones y nodos lineales puede que no sea capaz de realizar toda la tarea. Mientras que una red con un gran número de conexiones y nodos no lineales puede sobreajustarse al ruido en los datos de entrenamiento y fallar en la adquisición de una buena capacidad de generalización.

Todavía el diseño de la arquitectura depende mucho del trabajo del humano experto y del tedioso proceso de ensayo y error. No hay una forma sistemática y automática para diseñar un arquitectura cercana a la óptima para una tarea dada. Las investigaciones sobre *algoritmos constructivos* y *destructivos* representan un esfuerzo hacia el diseño automático de arquitecturas. A grandes rasgos, un algoritmo constructivo comienza con una red minimal – mínimo número de capas ocultas, nodos y conexiones– y va agregando nuevas capas, nodos y conexiones durante el entrenamiento. Un algoritmo destructivo utiliza la estrategia opuesta, comienza con una red maximal y va borrando capas, nodos y conexiones innecesarias durante el entrenamiento. Sin embargo estos métodos son susceptibles de “caer” en una estructura óptima local.

El diseño de arquitecturas óptimas de RNA puede formularse como un problema de búsqueda en el espacio de las arquitecturas, donde cada punto representa una arquitectura completa, y dado algún criterio para calificar su performance –menor error en el entrenamiento, menor tiempo de aprendizaje, menor complejidad de la red etc.– pueden utilizarse algoritmos evolucionarios para resolver el problema.

Al igual que como ocurre con la evolución de los pesos de conexión, la evolución de arquitecturas tiene dos grandes fases: la representación del genotipo y el algoritmo elegido para llevar adelante la evolución. Uno de los puntos clave en la codificación genética es decidir cuanta información acerca de la arquitectura se codificará en el cromosoma. En un extremo, todo los detalles, es decir toda conexión y nodo de la arquitectura puede ser especificado por un cromosoma. Esta clase de representación es llamada *codificación directa*. En el otro extremo, solo los más importantes parámetros de una arquitectura, como el número de capas ocultas y nodos ocultos en cada capa son codificados. Otros detalles sobre la arquitectura son dejados para que los decida el proceso de entrenamiento. Esta clase de representación es llamada *codificación indirecta*.

Una vez seleccionado el esquema de codificación, la evolución de arquitecturas puede progresar siguiendo el siguiente ciclo:

1. Decodificar cada individuo de la generación corriente obteniendo así la arquitectura de la RNA –fenotipo–.
2. Entrenar varias veces cada una de las RNA con una regla de aprendizaje predefinida, comenzando desde diferentes conjuntos aleatorios iniciales de pesos de conexión.
3. Computar el fitness de acuerdo al resultado promediado de cada entrenamiento y otros criterios de performance.
4. Seleccionar los padres y generar la siguiente generación.

Debe notarse que la evolución de una arquitectura óptima de red neuronal artificial, debe complementarse con algún método de entrenamiento tradicional puesto que los pesos de la conexión no se someten al proceso evolutivo. Por lo tanto es un nuevo caso de hibridación, de hecho obsérvese que los pasos descritos arriba siguen el esquema general presentado en la Ilustración 75. Si pretendemos olvidarnos definitivamente de los entrenamientos tradicionales de RNA podemos evolucionar conjuntamente la arquitectura y los pesos de conexión – algoritmo evolucionario puro– pero esto lo veremos más adelante en este mismo capítulo.

Una investigación considerable sobre evolución de arquitecturas de RNA ha sido realizada en los años recientes. La mayoría se ha concentrado en evolucionar la topología y algunos pocos evolucionaron la función de transferencia de cada nodo.

Esquema de codificación directa

Se han desarrollado muchos tipos distintos de codificación directa, es decir aquella que codifica exhaustivamente todos los parámetros de la red. Por ejemplo una matriz $C^{N \times N}$ de elementos c_{ij} binarios puede representar una arquitectura con N nodos indicando ausencia o presencia de la conexión desde el nodo i al nodo j según c_{ij} contenga o no el valor nulo. El cromosoma será la concatenación de cada fila o columnas de la matriz.

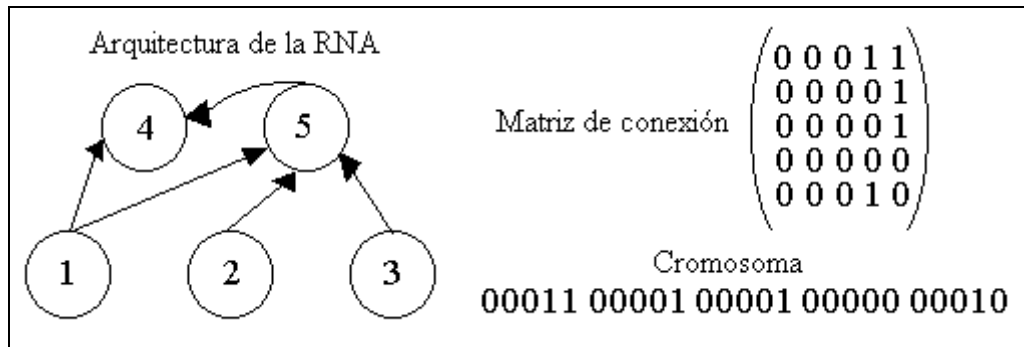


Ilustración 76 Codificación directa de la topología de una RNA

Generalmente se utiliza el error y el tiempo de aprendizaje para calcular la función de fitness. La medida de complejidad como el número de nodos y conexiones es también utilizada en el fitness. Schaffer¹ ha presentado un experimento con el cual muestra que una RNA diseñada por evolución tiene mejor habilidad de generalización que una entrenada con Backpropagation usando una arquitectura diseñada por el hombre.

Un problema potencial de la codificación directa es la escalabilidad. Una red grande requiere una matriz muy grande. Una forma de cortar la matriz es usar conocimiento del dominio para reducir el espacio de búsqueda. Por ejemplo, si se quiere evolucionar una red completamente conectada feedforward puede codificarse en el cromosoma sólo el número de capas ocultas y los nodos en cada capa oculta.

El problema de permutación existe también en la evolución de arquitecturas disminuyendo la probabilidad de producir descendencia mejor adaptada por recombinación genética. Por lo tanto algunos investigadores evitan el crossover y adoptan sólo mutación en la evolución de arquitecturas.

Sin embargo se ha mostrado que el crossover puede ser útil e importante para incrementar la eficiencia de evolución para algunos problemas. Además, Hancock² sugiere que el problema de permutación puede no ser tan severo como se ha supuesto. Aparentemente este es otro caso de discusión que aún permanece abierta.

Esquema indirecto de codificación

La codificación indirecta ha sido utilizada por varios investigadores para reducir la longitud de los genotipos de las arquitecturas, codificando sólo algunas características de ellas en el cromosoma. Los detalles acerca de cada conexión en una RNA son o bien predefinidos de acuerdo a previo conocimiento o especificados por un conjunto determinístico de reglas de desarrollo.

¹ Citado por Xin Yao en [XinYao99]

² Citado por Xin Yao en [XinYao99]

Este tipo de codificación puede producir genotipos compactos pero no es muy buena para conseguir RNA compactas con buen nivel de generalización. Algunos argumentan que la codificación indirecta es más plausible que la directa en la biología, aparentemente es imposible codificar en genes cada una de las características de nuestras redes neuronales, puesto que sencillamente no hay lugar en los cromosomas para albergar tanta información.

La arquitectura de las RNA puede ser codificada en un cromosoma a través de un conjunto de parámetros tales como el número de capas ocultas, el número de nodos ocultos en cada capa, el número de conexiones entre dos capas, etc. La ventaja es que disminuye la longitud de los cromosomas, la desventaja es que también reduce el espacio de búsqueda al que podrá acceder el algoritmo. Estos recortes se realizan por un conjunto de suposiciones que deben imponerse. Por ejemplo, se puede codificar sólo el número de nodos ocultos en la capa intermedia, claro está asumiendo que se trata de una RNA feedforward completamente conectada con una sola capa oculta. En general la representación paramétrica es conveniente cuando conocemos qué clase de arquitectura estamos tratando de encontrar, de lo contrario estaremos truncando a ciegas el espacio de búsqueda.

Otro método utilizado consiste en codificar reglas de desarrollo para construir las arquitecturas neuronales. El efecto destructivo del crossover¹ es minimizado puesto que la representación de reglas de desarrollo es capaz de preservar los “*Building Blocks*” prometedores –ver “Paralelismo implícito y la Hipótesis de construcción de bloques” en la página 62–.

Una regla de desarrollo usualmente se describe por una ecuación recursiva o una regla de producción. Veamos un ejemplo utilizando reglas de producción.

Conjunto de reglas de desarrollo

$$S \rightarrow \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad A \rightarrow \begin{pmatrix} a & a \\ a & a \end{pmatrix} \quad B \rightarrow \begin{pmatrix} i & i \\ i & a \end{pmatrix}$$

$$C \rightarrow \begin{pmatrix} i & a \\ a & c \end{pmatrix} \quad D \rightarrow \begin{pmatrix} a & e \\ a & e \end{pmatrix} \quad \dots$$

$$a \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad c \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad i \rightarrow \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad e \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \dots$$

Ilustración 77 Reglas de desarrollo como conjunto de reglas de producción o reescritura. Cada regla puede representarse en el cromosoma por cuatro alelos que representan la parte derecha de la misma. La parte izquierda de la regla puede representarse implícitamente por la posición dentro del cromosoma. S es el *start simbol* y por donde comienza el proceso de reescritura.

¹ Me refiero a la probabilidad que una operación de crossover destruya un esquema dado (ver “Teorema fundamental – Teorema de esquemas” pág. 59)

El patrón de conectividad de la arquitectura en forma de matriz es construido desde el *start simbol*, por la aplicación repetida de las reglas de desarrollos a los elementos no terminales hasta que la matriz contenga sólo elementos terminales.

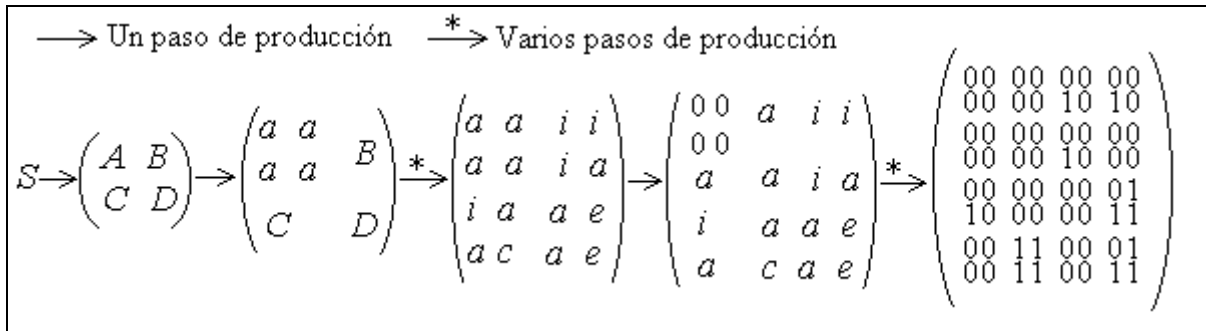


Ilustración 78 Desarrollo de la conectividad de la arquitectura por la aplicación sucesiva de las reglas de desarrollo de la Ilustración 77.

La matriz de conexión desarrollada en la Ilustración 78 define una arquitectura particular de RNA.

Algunos buenos resultados con representación de reglas de desarrollo fueron reportados. Sin embargo, el método tiene varias limitaciones. A menudo necesita predefinir el número de pasos de reescritura. No permiten reglas recursivas. Un genotipo compacto no implica un fenotipo compacto es decir RNA compactas. Generalmente esta representación separa la evolución de arquitecturas de la de pesos de conexión lo que crea algunos problemas en la evolución.

Evolución de función de transferencia de los nodos

En el punto anterior se asume que la función de transferencia en cada nodo es fija y predefinida por un experto humano. Generalmente se asume que esta función es la misma para todos los nodos al menos de la misma capa.

Algunos investigadores han realizado evoluciones de topología y función de transferencia al mismo tiempo codificándolas en los cromosomas. Se han propuestos ciertas variantes, como por ejemplo fijar dos tipos de funciones de transferencia, una sigmoide y una gaussiana, comenzar el proceso evolutivo con un determinado porcentaje de nodos con un tipo de funciones y el resto con el otro. La evolución intenta determinar el porcentaje óptimo de nodos con cada tipo de función.

Se han reportado algunos buenos resultados con la evolución de funciones de transferencia.

Evolución simultánea de arquitectura y pesos de conexión

La evolución de la arquitectura de las RNA sin los pesos hace que ésta sea ineficiente. El problema es que para evaluar el fitness de la arquitectura conseguida se debe entrenar la red para lo cual se inicializan aleatoriamente los pesos de conexión. Pero los algoritmos de entrenamiento dependen mucho de los valores iniciales de los pesos, así un genotipo G_1 puede ser de mejor calidad que G_2 y sin embargo su fitness ser peor debido a los pesos iniciales elegidos durante su evaluación. Habitualmente se subsana este problema realizando varios entrenamientos con distintos valores iniciales y la medida de aptitud es tomada como un promedio de estos entrenamientos. Obviamente esto trae aparejado un mayor tiempo de procesamiento. Esta es una de las mayores razones por la cual se han evolucionado arquitecturas de redes pequeñas con esta estrategia.

Por lo tanto se puede concluir que la evolución de arquitecturas sin información de los pesos dificulta una certera evaluación del fitness y consecuentemente la evolución se hace muy ineficiente.

Una forma obvia de solucionar este problema es evolucionar simultáneamente arquitecturas y pesos de conexión.

Una cuestión importante en la evolución de RNA es la elección de los operadores de búsqueda utilizados en el algoritmo evolutivo. Tanto crossover como mutación han sido utilizados. Sin embargo, como ya se dijo antes, el uso de crossover está cuestionado por algunos investigadores pues contradice las ideas básicas detrás de las RNA. Ya vimos en otro capítulo que este operador trabaja bien cuando existen buenos bloques constructivos¹, pero esto no es claro que ocurra en las RNA puesto que ellas realizan una representación distribuida del conocimiento entre todos los pesos de la Red. La recombinación de una parte de la red con otra parte de otra es probable que destruya ambas.

Sin embargo si la red no usa una representación distribuida sino más bien localizada, como ocurre con algunos tipos de redes neuronales, entonces el crossover puede ser muy útil.

Evolución de reglas de aprendizaje

Las investigaciones en evolución de reglas de aprendizaje están aun en una temprana etapa de desarrollo [*XinYao99*].

Los parámetros del algoritmo Backpropagation –como velocidad de aprendizaje y momento– ajustados por evolución puede ser considerado como el primer intento de evolucionar reglas de aprendizajes. Varios autores han evolucionado estos parámetros junto con la arquitectura. Otros han evolucionado los parámetros pero con una arquitectura fija predefinida. Estos parámetros evolucionados tienden a optimizar dicha arquitectura más que a evolucionar una regla genérica de aprendizaje.

El mismo problema descrito antes con respecto a la inicialización de pesos aleatorios se repite aquí y también se resuelve con varios entrenamientos para calcular un promedio. Si la arquitectura es fija, entonces la regla de aprendizaje evolucionará hacia la óptima para esa arquitectura. Si se desea evolucionar una regla de aprendizaje cercana a la óptima para varias arquitecturas, el fitness se basa en el promedio de entrenamientos de varias arquitecturas.

Un ciclo típico de evolución de reglas de aprendizaje para varias arquitecturas puede ser descrito así.

1. Decodificar cada regla de aprendizaje.
2. Construir un conjunto de RNA con arquitecturas y pesos iniciales aleatorios y entrenarlos usando la regla decodificada.
3. Calcular la función de fitness para cada individuo de acuerdo al promedio del resultado.
4. Seleccionar padres para generar la próxima generación.

Generalmente se asume que la regla de aprendizaje es la misma para todas las conexiones de la red y que la adaptación de los pesos depende solo de información local –valores de entrada, activación de la salida del nodo, los pesos de conexión actuales, etc. –.

¹ Los buenos bloques constructivos son esquemas de corta longitud de definición, bajo orden y cuyo fitness es mayor al promedio –para más información ver “Paralelismo implícito y la Hipótesis de construcción de bloques” Pág. 62–

Una regla de aprendizaje se considera que es una función lineal de estas variables locales y sus productos. De manera general se puede escribir así:

$$\Delta w(t) = \sum_{k=1}^n \sum_{i_1, i_2, \dots, i_k=1}^n \left(\theta_{i_1 i_2 \dots i_k} \prod_{j=1}^k x_{i_j}(t-1) \right)$$

donde t es el tiempo, Δw es el cambio de peso, x_1, \dots, x_n son variables locales, y los θ_i son coeficientes reales que serán determinados por la evolución. En otras palabras, la evolución de reglas de aprendizaje es equivalente a evolucionar vectores de valores reales θ_i . Diferentes θ_i determinan diferentes reglas de aprendizaje.

Debido al largo número de posibles términos en esta ecuación la evolución se hace muy lenta y muchas veces resulta impracticable. Una cuestión importante aquí es determinar un subconjunto de términos descriptos en esta ecuación. Para citar un ejemplo Chalmers¹ define una regla de aprendizaje como una combinación lineal de 4 variables, codifica los valores reales en binario y usa arquitectura fija. Después de 1000 generaciones la evolución descubrió la bien conocida regla delta y algunas variantes. Este experimento, aunque simple, demostró el potencial de evolución de reglas de aprendizaje.

De esta manera finaliza la aproximación general al paradigma de las RNAE o Neuroevolución. Ya se ha mencionado en la introducción de este capítulo que existen muchas otras estrategias neuroevolutivas que no voy a citar aquí. No obstante, en el presente trabajo, se han implementado algunas de ellas, y serán expuestas más adelante bajo el título “Estrategias de Neuroevolución implementadas” –pág. 194– .

Así doy por terminado este capítulo esperando que, junto a los anteriores, se constituya en una fuente útil de información donde el lector encuentre aquello necesario para hacer de este trabajo de grado una exposición razonablemente autocontenida.

¹ Citado por Xin Yao en [XinYao99]

Criaturas virtuales comandadas por RNAE

Introducción

Sin entrar en demasiados detalles, comentaré que los temas a tratar en este y en los subsiguientes capítulos se refieren a cuestiones relacionadas con la implementación, prueba, evaluación y comparación de distintas estrategias neuroevolutivas sobre diversas arquitecturas neuronales.

Respecto a estas arquitecturas neuronales que se someterán a evolución, debo mencionar que son varios los tipos implementados. Se desea determinar virtudes y deficiencias de las mismas en relación al dominio de problemas elegido y a su impacto sobre el algoritmo genético. Se realizarán pruebas empíricas con redes recurrentes y feedforward, sin restricciones de conectividad, con y sin término de tendencia, y con posibilidad de evolucionar la función de transferencia en cada nodo de la red.

Si bien no he invertido poco tiempo en el estudio realizado sobre la arquitectura neuronal, el presente trabajo de grado se interesa sobre todo en los desarrollos neuroevolutivos implementados, testeados y analizados que, como se mencionará en su momento, pueden considerarse extensiones de estrategias introducidas en el área por diversos investigadores.

Estas estrategias, siendo alguna de ellas muy conocidas y frecuentemente citadas en la literatura especializada, suelen presentar limitaciones en cuanto al tipo de red neuronal soportada, habiendo sido diseñadas para una clase en particular. Por lo tanto, ninguna de ellas se adecua completamente a toda la gama de redes implementadas en este trabajo, no consideran la conexión de tendencia ni realizan evolución de función de transferencia. En todos los casos ha sido necesario extender estos métodos no sólo para cubrir estas características sino para trabajar tanto con redes del tipo feedforward como con redes recurrentes.

También se presenta una estrategia de neuroevolución propia que llamaré “Grupos Neuronales Evolutivos” o simplemente GNE, en la que se evolucionan conjuntos de redes neuronales, para resolver colectivamente una tarea. Esto constituye un intento más cercano de emular al cerebro humano, al que es difícil concebir como una única gran red neuronal. Los estudios en el ámbito de las neurociencias han demostrado una gran localización funcional en zonas cerebrales precisas –ver “Zonas del cerebro. Distintas funciones” en la pág. 104– sugiriendo que este evolucionado órgano está constituido por un conjunto complejo de centros neuronales asociados a distintas actividades, aunque posiblemente interrelacionados.

Así con GNE pretendo conseguir grupos formados por varias redes neuronales artificiales que se organicen con alguna división funcional para llevar a cabo una tarea compleja. De acuerdo a la información de entrada, recibida en cada instante, será necesaria alguna estrategia de selección para determinar cuál o cuáles redes neuronales deben activarse para dar respuesta a la estimulación recibida durante la ejecución de la tarea encomendada.

En todos los casos se ha logrado dotar a ciertas criaturas virtuales, agentes de aquí en adelante, de *aprendizaje evolucionario por refuerzo*, sobre una clase genérica de problemas denominada “*sequential decision task*”.

Aprendizaje evolucionario por refuerzo es aquel que se consigue a través de la evolución de redes neuronales artificiales. “Sequential decision task” es una clase genérica de problemas donde una secuencia de decisiones debe ser realizada antes de poder medir cuál ha sido el efecto de cualquiera de ellas. Pero de estos temas me ocuparé luego.

El presente capítulo está organizado de la siguiente forma. A continuación, y como parte de esta introducción, dedicaré unas líneas para ubicar este trabajo en el contexto de la Inteligencia Artificial. Luego, ya en el ámbito de lo específico, se expondrán los detalles del dominio de problemas utilizado para la prueba y evaluación de las distintas arquitecturas neuronales y estrategias neuroevolutivas implementadas. He elegido enfrentar problemas de control de procesos, a través de la generación de controladores neuronales para comandar agentes que se mueven en un ambiente virtual de dos dimensiones esquivando obstáculos y alcanzando objetivos. Se describirá el campo de prueba, la interacción con los obstáculos y objetivos, los pormenores de la simulación como así también las características de los agentes. Se comentará con precisión la forma de calcular el fitness –necesario para cualquier proceso evolucionario–, junto con la descripción de la clase de problemas “Sequential decision tasks” que condicionan la manera de calcularlo.

Luego se presentarán los distintos tipos de arquitecturas neuronales que se evolucionarán y finalmente se comentará en detalle cada una de las seis estrategias de neuroevolución implementadas.

Se analizará un tipo de neuroevolución simple en la que se utiliza un algoritmo genético convencional junto a una representación cromosómica tradicional. Luego una dosis de elitismo será considerada en la segunda estrategia, cuyos resultados mostrarán su superioridad respecto de la primera.

SANE y ESP, dos estrategias desarrolladas por el Departamento de Ciencias de la Computación de la Universidad de Texas, serán estudiadas junto a las extensiones necesarias implementadas para soportar toda la gama de arquitecturas neuronales desarrolladas para este trabajo.

Una combinación de estos dos últimos métodos se presentará bajo el título de “ESP con población de blueprints”. Y para finalizar se expondrá GNE –Grupos Neuronales Evolutivos–, la nueva estrategia que se propone para atacar la solución a problemas complejos por medio de más de una red neuronal y de un mecanismo de evolución incremental. Comentaré algunos artículos, el planteo general, los primeros intentos y la versión final del método que fue implementada.

El presente trabajo en el contexto de la Inteligencia Artificial

Deseo hacer hincapié en que los desarrollos aquí presentados se inscriben en el paradigma de la Neuroevolución que surge de la combinación de otros dos: la Computación Evolutiva y las Redes Neuronales Artificiales.

El paradigma de las Redes Neuronales Artificiales es frecuentemente referenciado como paradigma Conexionista o simplemente Conexionismo.

La Computación Evolutiva y el Conexionismo conforman el enfoque subsimbólico de la Inteligencia Artificial, en oposición al enfoque simbólico conocido generalmente como Inteligencia Artificial Clásica o simplemente Inteligencia Artificial. De este último enfoque los sistemas expertos son los más representativos. Esta identificación del nombre de la

disciplina con sólo una rama de ella, seguramente se deba al desprestigio que sufrió por muchos años el Conexionismo mientras el simbolismo estaba en pleno apogeo¹.

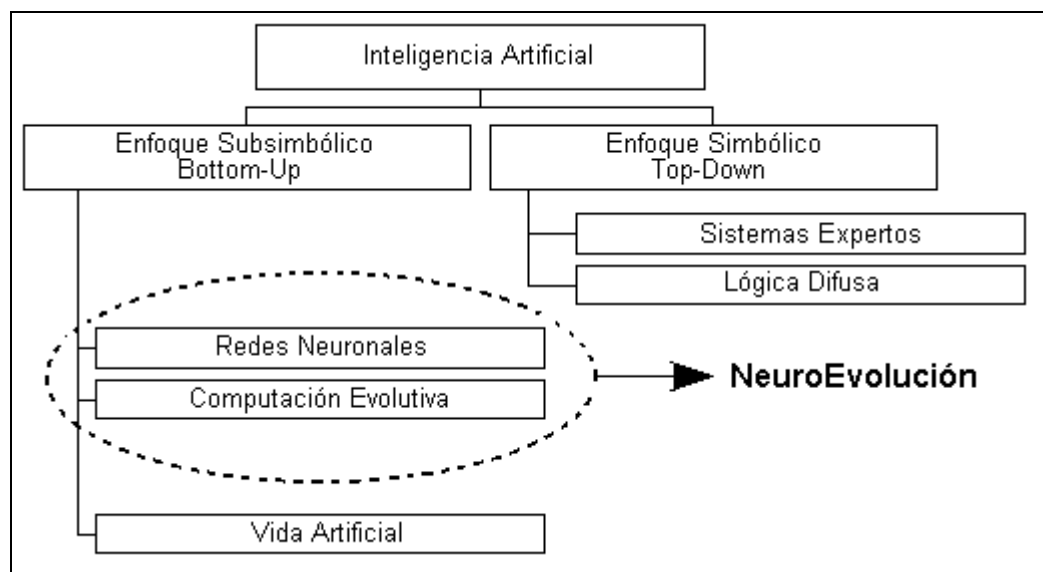


Ilustración 79 La Neuroevolución dentro del contexto de la Inteligencia Artificial.

Ya se ha mencionado algo sobre el enfoque subsimbólico en el capítulo titulado “Computación Evolutiva” cuando se hizo una clasificación del paradigma según [Gascón] – ver “Clasificación” en la pág. 78–. Allí se comentó que la rama subsimbólica de la Inteligencia Artificial se caracteriza por crear sistemas con capacidad de aprendizaje. Éste se puede obtener a nivel de individuo imitando el cerebro, como es el caso de las Redes Neuronales Artificiales, o a nivel de especie imitando la evolución, como lo hacen los algoritmos evolucionarios. En este capítulo debo agregar: “o ambas formas a la vez”, como se hace en el paradigma de la Neuroevolución. En todos los casos, se parte de representaciones de bajo nivel –neuronas y cromosomas– que se manipulan para conseguir el objetivo propuesto –de alto nivel de abstracción–, por eso se dice que, a diferencia de la Inteligencia Artificial Simbólica, el enfoque es de abajo hacia arriba (Bottom–Up).

Por el contrario el enfoque simbólico utiliza la estrategia de arriba para abajo (Top-Down): se comienza al nivel de la psicología de sentido común, se trata de imaginar el proceso mediante el cual se resuelve un problema particular y si la tarea no se puede lograr en un solo paso, entonces se procede a desmenuzarlo en partes más simples hasta integrarlo al nivel de hardware o software.

Según Marvin Minsky [Minsky], críticos miopes se han quejado de que el progreso de la investigación en Inteligencia Artificial simbólica se hace cada día más lento. Esto es, según él, comprensible puesto que las primeras etapas en cualquier campo se hacen más arduas conforme se solucionan los problemas más fáciles, y cada vez es más difícil hacer avances significativos. No obstante estas críticas hacia el simbolismo han ocasionado un creciente interés por el enfoque subsimbólico.

¹ Recordemos que a partir de la aparición del libro “Perceptrons: An Introduction to Computational Geometry” en 1969 escrito por Marvin Minsky y Seymour Papert, tratando en detalle la limitación de los perceptrones, se produjo un abandono masivo de la investigación en redes neuronales.

Pero aparentemente ambos tienen virtudes y deficiencias. Minsky reconoce la necesidad de sistemas integrados que exploten las ventajas de ambos, buscando algún lugar intermedio entre los dos extremos.

Dominio de problemas a resolver

En el presente trabajo se ha elegido, para probar varios métodos de neuroevolución y algunas arquitecturas neuronales, un conjunto atractivo de problemas relacionados con la evolución de comportamientos típicos en agentes que se mueven en un entorno de dos dimensiones interactuando con obstáculos y objetivos.

Este esquema de problema es frecuentemente utilizado por los investigadores del área. La razón es que aunque puede considerarse un *problema de juguete* lo cierto es que cae en la categoría de *problemas de control*, para los cuales las redes neuronales han demostrado excelente desempeño.

El objetivo es conseguir comportamiento inteligente en estos agentes que se mueven libremente en dos dimensiones dentro de los confines de un entorno virtual 2D, de forma rectangular, cuyos límites no pueden sobrepasar.

Otros objetos como obstáculos, que deberán aprender a evitar, y objetivos móviles o estáticos, a los que deberán alcanzar, interactúan con los agentes en este ambiente virtual. En cada instante dichos agentes estarán intentando moverse en alguna dirección y sentido definido por dos componentes que representan un vector de movimiento, que indica el desplazamiento sobre la superficie en que se mueven. No obstante, los obstáculos presentes y los límites del entorno pueden impedir que el movimiento se lleve a cabo.

Estas componentes del vector movimiento estarán aportadas por una –en el caso más general– o más de una –en el método GNE desarrollado– red neuronal artificial generada por evolución. Básicamente –luego ampliaré el concepto– en todas las estrategias implementadas, exceptuando GNE, una red neuronal controla al agente recibiendo como entrada una serie de señales –estímulos– que activan la RNA que al evaluarse determina la respuesta conformada por sus neuronas de salida cuyos valores se reflejan directamente en una acción de movimiento.

Típicamente cada red controladora tendrá dos neuronas de salidas que corresponderán a las componentes x e y del vector de movimiento, es decir, representan los desplazamientos Δx y Δy que realizará el agente en el entorno.

He construido un entorno de simulación que puede considerarse una simplificación del simulador para Khepera (Michel 1995), un pequeño robot de forma circular –5 centímetros de diámetro– diseñado con fines didácticos y de investigación¹. Para más datos sobre el entorno de simulación desarrollado aquí, puede consultarse el próximo capítulo –pág. 238– en donde se describen las utilidades implementadas en el software que acompaña este trabajo de grado.

El objetivo aquí es implementar, evaluar, medir y comparar distintos métodos neuroevolutivos y no el de desarrollar una fiel simulación física. Por ello considero que un entorno como el

¹ Pese a su reducido tamaño, el robot Khepera no es fácil de controlar. Posee una serie de sensores infrarrojos y de luces ubicados alrededor de su circunferencia y dos ruedas controladas por dos motores separados. El simulador contiene utilidades para visualizar la red neuronal que controla al robots, así puede observarse su arquitectura y activaciones durante la simulación.

construido alcanza holgadamente para cumplimentar los objetivos propuestos. De hecho varios investigadores han presentado sus desarrollos evaluando distintas estrategias neuroevolutivas en entornos similares como por ejemplo Gómez y Miikkulainen que en [Gomez97] utilizaron un ambiente muy parecido al que se implementó aquí –aunque mucho más simple– para testear un método de neuroevolución incremental junto a una nueva estrategia neuroevolutiva llamada ESP –Enforced subpopulation– que también se ha implementado en este trabajo y que se expondrá luego.

He experimentado evolucionando distintos comportamientos como captura de presa, evasión de obstáculo, búsqueda de salida, alcance de objetivo. Todos ellos se consiguen definiendo un escenario adecuado, conformado por la forma y dimensión del entorno como así también la disposición y cantidad de obstáculos en este ambiente. Otros parámetros como la cantidad de objetivos existentes en el ambiente, si estos objetivos son estáticos o móviles, completan la definición de los distintos comportamientos a resolver.

He extendido varios métodos neuroevolutivos, relativamente nuevos, que vieran la luz por primera vez en estos últimos 6 años, para conseguir ciertos comportamientos “inteligentes” en los agentes para distintos escenarios. Como ya comenté, fue necesario extender cada uno de ellos para permitir evolución de función de transferencia y soportar una gama más amplia de arquitecturas neuronales con las cuales trabajar¹.

Por supuesto, también se ha utilizado el nuevo modelo propuesto de neuroevolución GNE para conseguir conjuntos en el que más de una red intervienen en la decisión del movimiento a seguir por el agente. Sin embargo los detalles de estos desarrollos se presentarán más adelante en este mismo capítulo. Ahora me interesa comentar en qué consiste la evaluación de las redes neuronales evolucionadas, necesaria para la asignación de fitness que requiere el Algoritmo Genético.

Las redes neuronales de cada generación son evaluadas en una simulación donde el tiempo de la misma es dividido en instantes discretos que llamaré pasos de simulación. Así en cada paso el agente recibe las señales de estímulo que excitan las neuronas de entrada de su RNA, desencadenando su evaluación y como consecuencia la salida de la misma que se traduce en un movimiento determinado por el vector que definen las neuronas de salida de la Red.

Veamos los elementos que conforman el entorno virtual y la manera que los agentes evolucionados interrelacionan con ellos.

El campo de prueba

El entorno 2D que representa el mundo virtual en el que se desempeñan los agentes, está constituido por un área plana, de forma rectangular, en donde se producirá la interacción con los obstáculos, los límites del entorno y los objetivos.

¹ En el presente trabajo se han implementado redes neuronales recurrentes y feedforward, sin restricciones de conectividad, con y sin conexión de tendencia. También se utilizan distintas funciones de transferencia en cada nodo. Para más información ver “Tipos de Redes Neuronales Artificiales implementadas” en la pág. 184.

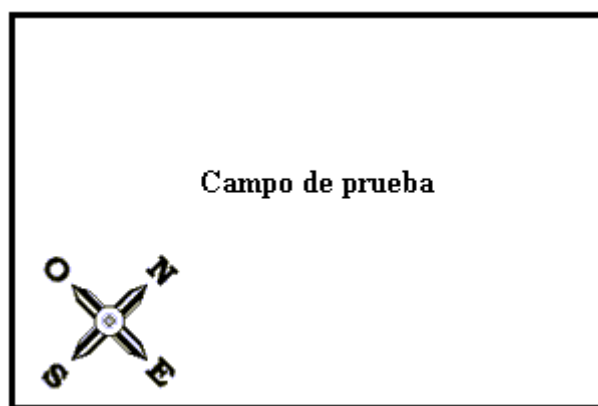


Ilustración 80 El campo de prueba donde tiene lugar la simulación es un recinto rectangular en el que se mueven los agentes intentando alcanzar los objetivos propuestos.

Se han marcado los puntos cardinales en el campo de prueba para utilizarlos como referencia en el momento de explicar la configuración de las neuronas de entrada de las redes neuronales. En el software presentado como parte de este trabajo, al inspeccionar una red neuronal –que puede hacerse en cualquier momento de la simulación– las unidades de entrada afectadas a la detección de obstáculos están debidamente etiquetadas con los puntos cardinales.

Las dimensiones del área son configurables como así también la cantidad de objetivos, su disposición en el campo de prueba, y si son móviles o estáticos. Asimismo puede determinarse el punto de partida de los agentes.

Finalmente, la disposición, orientación, tamaño y cantidad de obstáculos son también configurables, permitiendo así una gran variedad de escenarios distintos en el que los agentes deberán desempeñarse evolucionando diferentes comportamientos.

El accionar de los agentes en este campo de prueba puede interpretarse como una simulación en la que se presenta el desempeño de ciertos robots dotados con la capacidad de girar sobre su propio eje un ángulo de 0° hasta 360° y avanzar una determinada distancia –posiblemente distinta cada vez– ambas acciones en un solo paso de simulación.

Los obstáculos

Los obstáculos están representados por segmentos de rectas de posiciones, direcciones, y tamaños previamente definidos, que se disponen en el entorno formando diversos escenarios para las simulaciones. Estos obstáculos no pueden ser sobrepasados por los agentes, pueden ser detectados únicamente si la posición del agente es cercana al mismo y es posible que colisionen contra ellos.

Los objetivos –presa o destino–

Los objetivos, al igual que los agentes y los obstáculos, son objetos que ocupan posiciones en el campo de prueba. Éstos pueden ser estáticos o móviles con desplazamientos aleatorios y pueden haber más de uno.

En caso de conformar un escenario con objetivos móviles, éstos pueden considerarse como presas descuidadas –porque no huyen ante la cercanía del agente que desempeña el rol del depredador–.

Si por el contrario, se utilizan objetivos estáticos estaremos evolucionando el comportamiento de movimiento hacia un destino geográfico definido. La combinación de objetivos fijos, junto

con la colocación precisa de obstáculos y punto de partida de los agentes, nos lleva por ejemplo a situaciones en la que estos últimos deban resolver el problema de salida de un laberinto. Como veremos más adelante, en todos los casos se tratará de laberintos simples porque se ha limitado bastante la información captada por los agentes, de esta manera, se logra conseguir grandes retos en escenarios muy simples.

Las simulación

Para comenzar la simulación es necesario crear el escenario de la misma colocando en el campo de prueba los objetivos, obstáculos y el agente en la posición de partida. Como ya se mencionó antes, el escenario para la simulación queda determinado por las dimensiones del campo de prueba, la disposición y cantidad de obstáculos, la cantidad de objetivos estáticos o móviles y el punto de partida de los agentes.

Como ya se anticipara, el tiempo de simulación está dividido en instantes discretos llamados pasos de simulación. Así, la duración de la misma queda definida por medio de un parámetro que debe ser establecido antes de comenzar el proceso evolutivo, y que se refiere a la cantidad de estos pasos.

Para cada paso de simulación el *controlador neuronal* del agente realiza una evaluación de la información de entrada provista por sus sensores en ese instante, y promueve el correspondiente movimiento. Prefiero utilizar la frase “controlador neuronal” en lugar de “red neuronal” debido a que en el método GNE el agente será comandado no por una red sino por un grupo de éstas. Así por controlador neuronal debe entenderse red neuronal en todos los casos de neuroevolución implementados, salvo en GNE donde el controlador es un grupo neuronal formado por varias RNA.

No sólo los agentes pueden moverse en cada paso de simulación, también lo harán los objetivos, en el caso de ser móviles. Estos últimos pueden considerarse como agentes “descerebrados”, que en lugar de poseer un controlador neuronal que analiza la entrada de información, utilizan un generador aleatorio de señales para el movimiento. Para impedir un comportamiento sumamente inestable con poco recorrido espacial, los objetivos móviles, con alta probabilidad, elegirán repetir el movimiento realizado en el paso de simulación anterior.

El accionar del agente durante la simulación no produce alteración alguna sobre el entorno. Incluso en el caso de evaluar el comportamiento de captura de presa, una vez que el agente ha alcanzado su objetivo, éste no desaparece, pues interesa simular sólo la captura que termina cuando la presa es alcanzada. Debido a esto es posible realizar una única simulación común para todos los agentes que deban ser testeados. Esta fue la opción que se eligió en el software desarrollado, pero debe tenerse en cuenta que en realidad representa un conjunto de simulaciones individuales para cada agente visualizadas de manera simultánea. En ningún momento los agentes detectan la presencia de sus congéneres.

La realización simultánea de los tests produce una visualización más atractiva y brinda información de referencia para un análisis comparativo visual del desempeño de los agentes que no puede hacerse con una serie de simulaciones en secuencia.

Como ya se mencionó la simulación está formada por un número de pasos en los que agentes y objetivos pueden repositionarse. Por lo tanto, el movimiento se evidencia en la sucesión de estos pasos de simulación. Como veremos más adelante en este capítulo cada neurona de salida de los controladores neuronales que comandan a los agentes dará como resultado un valor perteneciente al intervalo $[0..1]$. Se puede decir entonces que la salida de las RNA de los controladores están conformadas por un vector de 2 dimensiones (x_1, x_2) con $x_i \in [0..1]$.

Para cada instante t de la simulación se realiza una evaluación del controlador neuronal y su salida se interpreta como el movimiento que realiza el agente en ese instante. Más precisamente cada componente x_i del vector de salida se corresponde a un valor de desplazamiento en la coordenada i en el campo de prueba. Como son necesarios también los desplazamientos negativos y x_i sólo toma valores entre 0 y 1, aplicamos la siguiente conversión a la salida del controlador neuronal.

$$\Delta x_i = 2(x_i - 0.5)$$

Así obtenemos un valor de desplazamiento entre -1 y 1 para cada coordenada de nuestro entorno virtual 2D.

Se puede calcular la posición del agente de la siguiente manera: Sean $POS(t)$ una función vectorial paramétrica que devuelve la posición del agente en el instante t , y $C(t)$ el vector de salida del controlador neuronal que comanda al agente en el instante t . Si $POS(t-1) = (X_1, X_2)$ y $C(t) = (x_1, x_2)$, la posición del agente en el instante t se define como:

$$POS(t) = (X_1 + 2(x_1 - 0.5), X_2 + 2(x_2 - 0.5))$$

De esta manera la mayor velocidad que podrá desarrollar el agente será aquella conseguida al aplicar un movimiento de longitud $\sqrt{2}$, y sólo podrá alcanzarla en una dirección desplazada en ángulo de 45° de los ejes coordenados.

Debo remarcar que la posición del agente dentro del campo de prueba está determinada por un par ordenado de números reales. Por ello un agente puede moverse lentamente con desplazamientos muy inferiores a 1. Esto ocasiona que si ejecutamos “paso a paso” una simulación en la herramienta de software desarrollada para este trabajo, observaremos que algunos agentes necesitarán varios pasos de simulación para realizar un pequeño movimiento de un pixel. Esto se debe al efecto inevitable de la discretización del espacio para poder graficarlo en una pantalla de computadora.

Los agentes

Como se dijo anteriormente se puede considerar a los agentes como robots, con una determinada cantidad de sensores por la que reciben estímulos que serán procesados en su RNA para estimular algún componente mecánico y conseguir así un determinado movimiento en reacción al estímulo recibido.

Dejemos de lado por un momento las particularidades concernientes al método GNE. Asumamos, por simplicidad que sólo una RNA es la encargada de controlar al agente. Más adelante bajo el título “Grupos Neuronales Evolutivos –GNE–” en la pág. 224 se explicará detalladamente como funcionan los grupos neuronales.

En la herramienta de software desarrollada como parte del presente trabajo se ha dejado la posibilidad de que el operador decida una gran cantidad de características parametrizables para realizar distintos ensayos neuroevolutivos. Entre estas características se encuentra la cantidad de neuronas de entrada de las RNA de los agentes, lo que determina la cantidad de sensores de los mismos. Para la presente explicación se fijará a modo de ejemplo en 14 el número de neuronas de entrada.

De la cantidad de neuronas de entrada 6 se utilizan para detección de obstáculos y otros aspectos que explicaremos luego, mientras que el resto queda libre para la detección de objetivos.

Así para 14 entradas las primeras 8 son sensores de objetivos. Para la simulación se podría suponer que los objetivos son fuentes luminosas y los agentes poseen algún tipo de sensor de

luz. En este ejemplo 8 sensores distribuidos uniformemente alrededor del agente. Estos sensores son algo precarios y proporcionan sólo tres valores: 0 ; 1 ó 5 indicando respectivamente ausencia, presencia detrás de algún obstáculo, o visualización directa de un objetivo en la dirección correspondiente. Veamos una ilustración donde se ejemplifica el valor que toman las neuronas de entrada de la RNA para una determinada situación.

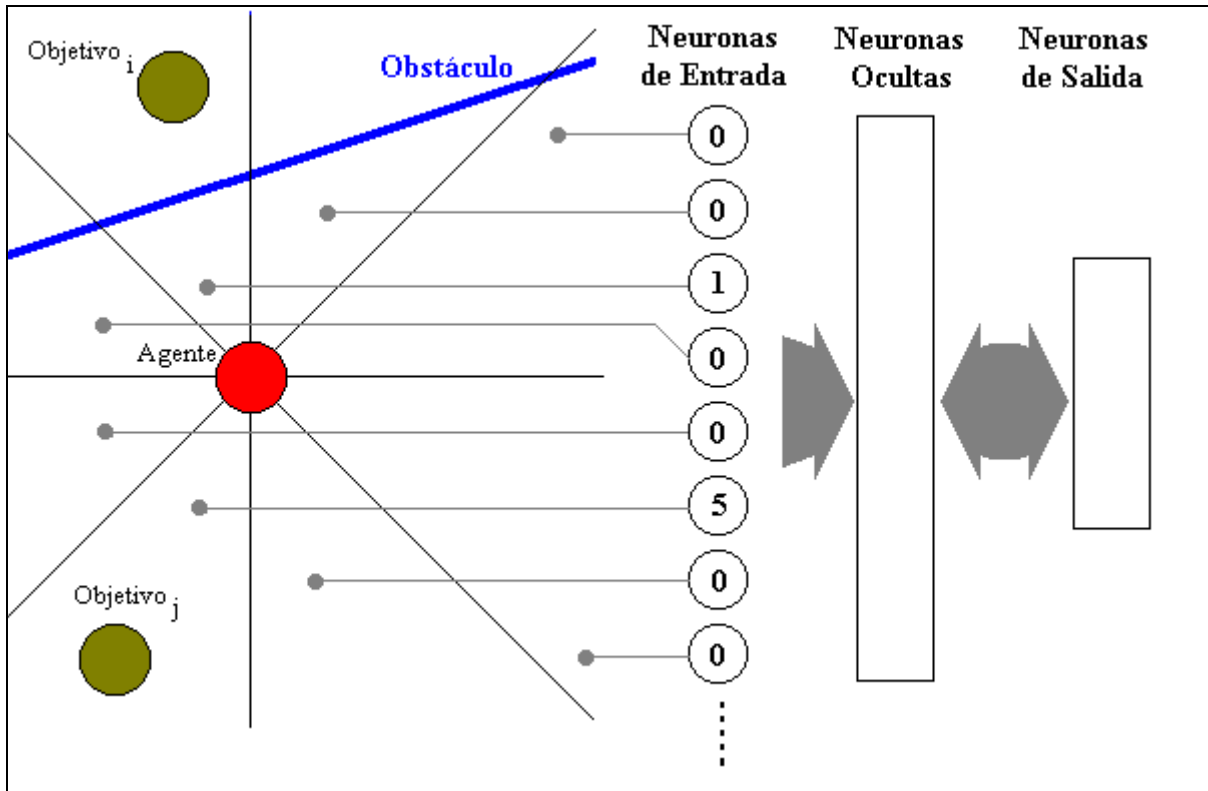


Ilustración 81 Neuronas de Entrada encargadas de recibir los estímulos provenientes de los objetivos. Puede observarse que la circunferencia del agente es dividida en sectores iguales –8 en este ejemplo– a los que le corresponde una neurona receptora de la RN que lo controla. También puede observarse que el Objetivo $_i$ se encuentra detrás de un obstáculo mientras que el camino al Objetivo $_j$ se encuentra libre. La doble flecha existente entre la capa oculta y la capa de salida se debe al hecho que se admiten redes neuronales recurrentes.

La razón de los valores ha surgido por la experimentación y no se descarta la posibilidad de utilizar otros. Sí he encontrado útil que el valor de un objetivo a la vista sin obstáculos sea bastante mayor que los demás porque en general, llegar a la instancia de observar el objetivo sin obstáculos presupone un cambio de comportamiento drástico en la mayoría de los casos.

Las restantes 6 neuronas de entrada están comprometidas a la detección de obstáculos –o bordes del campo de prueba, que son tratados de la misma manera que los obstáculos– , detección de colisión y de tiempo de simulación.

A diferencia de lo que ocurre con los objetivos, se han provisto a los agentes de una limitada capacidad para detectar los obstáculos. Sólo pueden detectarlos en cuatro direcciones correspondientes a los puntos cardinales del campo de prueba y sólo si se encuentran relativamente cercanos –a no más del doble del propio diámetro del agente medido desde el centro de éste–.

Se puede pensar que los agentes poseen cuatro prolongaciones sensibles al tacto por las cuáles pueden apreciar la presencia de un obstáculo si se encuentra lo suficientemente cerca. Por lo

tanto se necesitan cuatro neuronas receptoras para la detección de obstáculos cercanos, y se ha agregado una quinta que detectará colisión contra un obstáculo o borde del campo. Estas cinco neuronas de entrada recibirán en cada instante –paso de la simulación– un valor 1 ó 0 dependiendo de presencia o ausencia de obstáculos cercanos. En el caso de la quinta neurona el valor 1 representará que el agente ha colisionado contra un borde u obstáculo.

Obsérvese que una forma indirecta de conocer sobre la existencia de un obstáculo lejano será a través de las neuronas comprometidas en la detección de objetivos que si reciben un valor 1 indica que se encuentran más allá de uno o más obstáculos en la dirección correspondiente a dicha neurona de entrada.

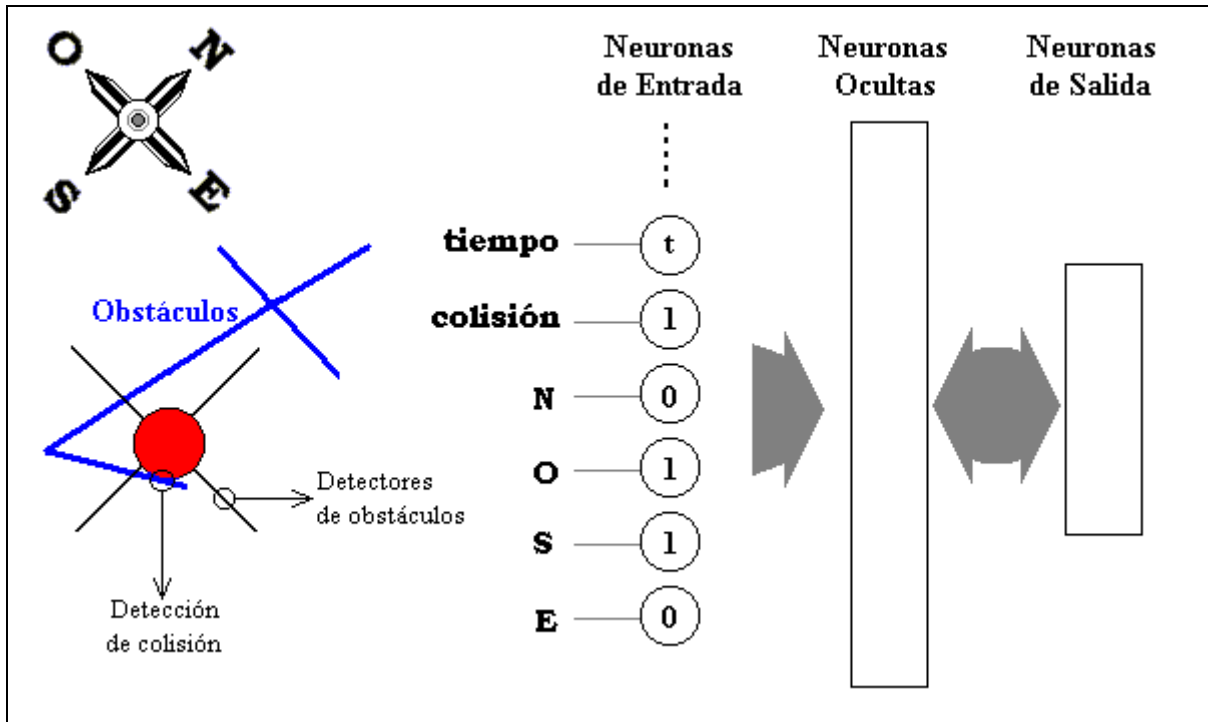


Ilustración 82 Neuronas de entrada afectadas a la detección de obstáculos, bordes, colisión y tiempo de simulación.

En la ilustración anterior es posible apreciar la presencia de una neurona de entrada que recibe como estímulo algún parámetro temporal. Este valor es una apreciación del tiempo de simulación, y pertenece al intervalo real $[0..1]$. Así el 0 representa el inicio de la simulación – antes del primer paso– y el 1 corresponde al último paso de simulación. Es decir que para una simulación de n pasos, si p es el paso corriente, el valor de esta neurona será p/n .

Esta información temporal puede ser útil en algunas circunstancias donde se requiera un cambio en el comportamiento del agente aún sin que cambien los estímulos recibidos desde el exterior. Por ejemplo, en general los agentes no serán capaces de resolver situaciones como esta:

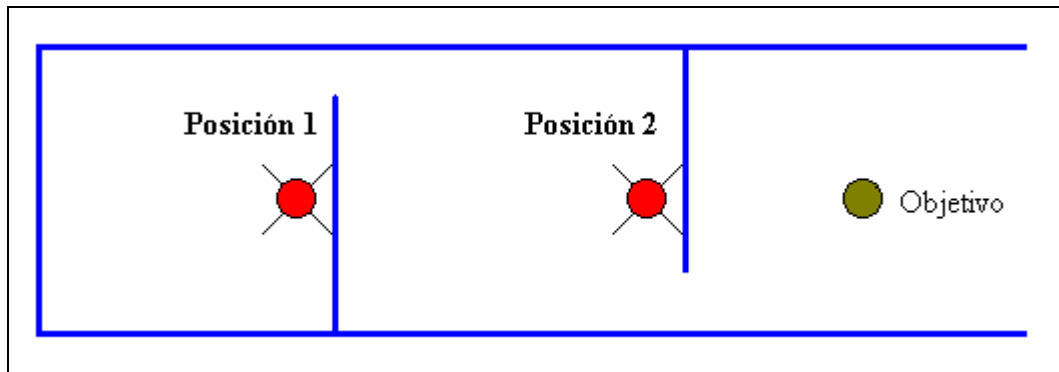


Ilustración 83 Escenario dificultoso. Obsérvese que tanto en la posición 1 como en la posición 2 la información captada desde el exterior es idéntica, sin embargo el agente debe tomar decisiones diferentes para alcanzar el objetivo.

Por ello se agrega una entrada a la red que representa el tiempo de simulación, o lo que es lo mismo, la cantidad de pasos realizados. Si fuese un robot, algún mecanismo interno debería incrementar esta entrada a la red. Esto posibilita un cambio de opinión, es decir después de un determinado tiempo la actitud del agente podría variar aún recibiendo el mismo estímulo exterior. No obstante situaciones como la anterior siguen siendo difíciles de resolver pues se apuesta a que un cambio en una sola neurona de entrada produzca un comportamiento totalmente distinto. Aún así se han realizado exitosamente ensayos evolutivos con escenarios como el presentado en la ilustración anterior.

Sin embargo pueden definirse escenarios que representen dificultades infranqueables para los agentes debido a la austera información de entrada con que se nutren sus redes neuronales. Esta dificultad no es grave pues se han definido otros escenarios que representan retos de mayor o menor dificultad pero todos ellos alcanzables. Así se puede medir las posibilidades e inconvenientes de las distintas estrategias neuroevolutivas estudiadas en el presente trabajo, que conforma el objetivo propuesto.

Nuevamente deseo remarcar que la simplicidad del entorno y la limitada capacidad de detección de obstáculos de los agentes no hace más fácil la tarea de la evolución. Ésta debe generar redes neuronales capaces de comandar a estas criaturas de escasa capacidad sensorial. Es notorio observar como se arriban a soluciones que tienden a posicionar al agente en aquellas zonas límites donde se produce variación en los datos de entrada. Esto se corresponde con una estrategia por demás conocida que todos alguna vez hemos utilizado en alguna habitación a oscuras: el tanteo.

Asignación de fitness

Veamos ahora la manera en que se asigna un valor de fitness a los agentes durante los ensayos evolutivos. Debo aclarar que en un principio se utilizó una forma de calcular este valor pero luego decidí cambiarla por las razones que expondré más adelante.

Todas las estrategias de neuroevolución implementadas comparten la misma forma de calcular el fitness durante la fase de evaluación de la red o grupo neuronal. Esto es a través de la simulación correspondiente en el campo de prueba para un escenario previamente definido, es decir, para una determinada configuración de cantidad y disposición de obstáculos y objetivos.

Una posible manera de asignar un valor de aptitud a cada RNA o grupo testado es hacerlo paso a paso en la simulación según el grado de conveniencia de cada movimiento efectuado por el agente que controla. Esta fue la primera elección, y luego de unas cuantas decisiones

sobre premios y castigos¹ se logró evolucionar con éxito los tipos de comportamientos buscado.

No obstante, comprendí luego que era más interesante evolucionar estos comportamientos sin la necesidad de asignar un valor de bondad a cada movimiento, simplemente porque existen otros muchos ejemplos de problemas en donde tal estrategia es más difícil o imposible de implementar. Tales problemas pueden enmarcarse en una clase genérica que ha merecido la atención de los investigadores en Inteligencia Artificial desde hace muchos años y que en la década pasada recibiera el nombre de *Sequential decision tasks* –Barto et al. 1990 y Grefenstette et al. 1990–.

Aprovechando la ocasión comentaré brevemente esta clase de problemas, mencionando cuáles son las motivaciones para desarrollar métodos que los resuelvan y esperando se comprenda la razón por la que he decidido cambiar la primera implementación de la función de fitness por otra más interesante desde el punto de vista de la clase de problemas a resolver..

Sequential decision tasks

Un jugador de ajedrez realiza cerca de medio centenar de movimientos durante una partida simple, y sin embargo, la mayoría de las veces no se puede evaluar la bondad de cada uno de ellos en el preciso momento en que se realizan. Por el contrario, es necesario observar cómo avanza el juego a lo largo de varios movimientos más.

Una partida de Ajedrez, representa una batalla de ideas. El objetivo final de cada bando es la conquista del Rey adversario, lo que se denomina Jaque Mate. Durante el transcurso de la partida se presentan oportunidades y escaramuzas que permiten ganar material o posiciones, lo que en general es una forma indirecta de concurrir al objetivo final.

Intentar descubrir cuáles de todos los movimientos de una partida de ajedrez son los responsables de un resultado –y en qué medida lo son– es extremadamente difícil, tanto como determinar cuáles no han contribuido al desenlace alcanzado. Quizá el jugador pueda ir evaluando paso a paso qué tan bien va logrando sus objetivos, pero para un observador que no conoce ni las tácticas, ni la estrategia asumida, la tarea de calificar con precisión cada movimiento es simplemente imposible.

No obstante cualquier observador entendido en el juego, podrá evaluar con algún grado de certeza cuál es la situación de cada jugador luego de una secuencia de varios movimientos, y en el extremo, al concluir el juego, sabrá si alguna de las secuencias de movimientos llevadas a cabo por ambos jugadores ha sido mejor que la otra.

Esta situación ha planteado varios interrogantes a los investigadores en Inteligencia Artificial refiriéndose el principal a cómo asignar un crédito –valor de bondad o fitness en el contexto de la Computación Evolutiva– a las decisiones individuales basado en la evaluación de una secuencia de decisiones. Esta problemática se conoce con el nombre de *credit assignment problem* –problema de asignación de crédito–.

¹ La evaluación de cuán bueno es un movimiento, muchas veces se torna difícil –más adelante se ampliará el concepto– y por lo tanto en esta primera elección de cálculo de fitness fue necesario contemplar varios aspectos simultáneamente. Por un lado se premiaron aquellos movimientos que acercaban al agente a alguno de los objetivos de la simulación –presas o sitios destino– pero al mismo tiempo se penalizaban, reduciendo su valor de aptitud, acciones conducentes a colisiones con obstáculos, la detención de movimiento –por colisión o propia elección– y los movimientos en círculos inconducentes.

El ajedrez y otros tantos juegos de tablero –damas, otello, go, etc.–, son representantes del tipo más genérico de problemas que nos ocupa: *sequential decision task*. Podemos decir que en los problemas de esta clase, una secuencia de decisiones debe ser realizada antes de poder medir cuál ha sido el efecto de cualquiera de ellas.

Algunos ejemplos del mundo real de problemas de secuencia de decisiones lo constituyen el enrutado de información en los routers de la Internet, el control de flujo químico en un reactor químico, el control de tráfico aéreo, etc. En todos estos casos, el efecto de una decisión simple se evidencia transcurrido un lapso de tiempo, y aún así, frecuentemente no queda claro cuáles decisiones fueron las responsables, y en qué medida, de lo acontecido. [Moriarty97_1]

El hecho de que las decisiones tomadas tengan tanto consecuencias inmediatas como futuras, hacen más difícil su tratamiento. A menudo la mejor estrategia a largo plazo presupone tomar acciones con perjudiciales consecuencias inmediatas. En el ajedrez, por ejemplo, la pérdida de una pieza, puede transformarse en beneficio a largo plazo, si se sabe aprovechar un posible flanco abierto en la defensa del oponente ocurrido como consecuencia de la captura. Un ejemplo de tácticas para ganar posiciones cediendo piezas son los denominados *gambitos*¹ dentro del mundo ajedrecístico. Sin ir más lejos en nuestro entorno virtual, un agente puede tener que decidir alejarse de su objetivo para tener posibilidades de alcanzarlo.

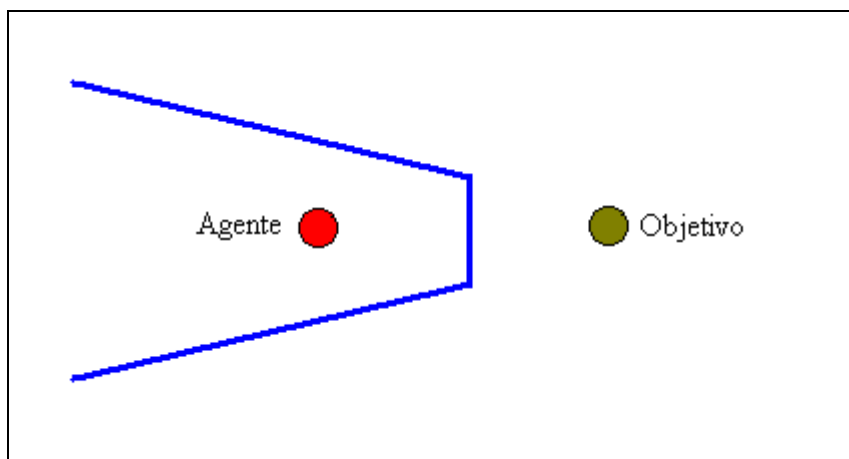


Ilustración 84 Situación que refleja la necesidad de alejarse del objetivo para poder alcanzarlo en algún momento. Las rectas de color azul representan obstáculos infranqueables para el agente que debe optar por retroceder para así poder rodearlo y despejar el camino hacia el objetivo.

Según David Moriarty en [Moriarty97_1] esta complejidad para tomar decisiones en esta clase de problemas, ha causado que muchas tareas del mundo real permanezcan sin automatizar o a lo sumo parcialmente automatizadas. Es por ello que abordar el estudio de estrategias de aprendizaje máquina –*learning machine*– para conseguir sistemas que resuelvan este tipo de problemas, siempre resulta atractivo.

¹ La palabra *gambito* viene del italiano *gambetta*, que significa literalmente zancadilla, en el sentido de trampa o celada puesta ante el adversario. Consiste en el sacrificio de un peón en la apertura para obtener ventaja en el desarrollo y en otros aspectos de la partida. Hay gambitos verdaderos y aparentes; los aparentes son aquellos en los que el peón que se sacrifica se recupera con facilidad, o no puede tomarse por ser evidente que la captura lleva a la derrota. Los verdaderos son los que implican un auténtico sacrificio, o sea, aquellos en los que la pérdida no puede recuperarse inmediatamente y en los que es necesario probar, con un juego exacto, que la compensación por el peón es suficiente. Creo que los gambitos son un claro ejemplo de los problemas a los que se enfrenta la Inteligencia Artificial en la resolución de problemas del tipo *sequential decision tasks*.

Ya se ha hablado en el capítulo “Redes Neuronales Artificiales y Neuroevolución” acerca de los distintos tipos de aprendizaje en el paradigma de las RNA. –ver “Tipos de aprendizaje” en la pág. 152–. Claramente el aprendizaje supervisado por corrección de error queda excluido en este tipo de problemas pues requiere un conjunto de ejemplos de comportamiento correcto, y como el lector puede advertir, tal conocimiento es aquí muy costoso de conseguir o simplemente imposible de hacerlo. No obstante, el aprendizaje por refuerzo¹ parece ajustarse correctamente a las exigencias de este dominio de problemas, y la evolución puede transformarse en una excelente estrategia para conseguir este aprendizaje, donde la señal de refuerzo no se obtiene a cada instante, sino por el contrario en forma muy esporádica.

Los algoritmos evolucionarios, y por supuesto entre ellos los genéticos, combinados con las RNA en el paradigma de la Neuroevolución, son ideales para atacar la resolución de problemas de este tipo. Utilizando una terminología más cercana a la utilizada en la literatura existente sobre aprendizaje, que a la propia de la Computación Evolutiva, diré que la neuroevolución es capaz de lograr un tipo de aprendizaje llamado *aprendizaje evolucionario por refuerzo* –AER– .

Para no extenderme demasiado en este tema, sólo agregaré que Moriarty concluye que la superioridad de AER respecto a otras aproximaciones de aprendizaje por refuerzo radica en su condición de estrategia más robusta de asignación de créditos².

Las motivaciones antes mencionadas, junto con la ya probada capacidad de la Neuroevolución para atacar este tipo de problemas, ha sido la causa para redefinir la estrategia de asignación del fitness de cada RNA evaluada. Por lo tanto, se ha enmarcado el dominio de problemas en la clase “*sequential decision task*” optando por no hacer ningún tipo de valoración sobre el desempeño del agente, sino hasta concluida completamente la simulación. De esta manera, los cientos –a veces miles– de pasos que conforman una simulación constituyen la secuencia de decisiones que será evaluada de acuerdo al resultado obtenido luego de su finalización.

Debo mencionar que este cambio en la obtención del valor de aptitud simplificó la tarea de implementación de la función de fitness, debido a que se pudo prescindir de una larga serie de consideraciones que eran necesarias realizar en el enfoque anterior para determinar el grado de pertinencia de cada decisión elegida. Como ejemplo véase la Ilustración 84 en donde se debe premiar la decisión de ir en sentido contrario al objetivo que se desea alcanzar. Se puede decir que esta situación demuestra que la primera manera de calcular el fitness no era la más acertada, pues la naturaleza de los problemas que me he planteado resolver en el presente trabajo corresponde precisamente a la de secuencia de decisiones que en un primer momento no pude apreciar con claridad.

Función de Fitness.

Como se decía anteriormente, este valor se calcula sólo después de concluida la simulación. Así como es de esperar tal magnitud estará relacionada con la distancia al objetivo –o los objetivos– resultante luego de realizar su último movimiento.

¹ Recordemos que mientras en el aprendizaje supervisado por corrección de error es necesaria la figura del “supervisor” que en todo momento especifica cuál debe ser la respuesta óptima del sistema, en el aprendizaje por refuerzo existe la figura del “juez” que evalúa la respuesta del sistema determinando el grado de adecuación de la misma.

² Puede consultarse [*Moriarty97_1*] para más detalle

Si existe más de un objetivo, se calcula el fitness en relación a cada uno de ellos tomando el mayor de los valores calculados como resultado final.

Una primera aproximación será decir que el valor de fitness en relación a un objetivo será inversamente proporcional a la distancia comprendida entre el agente y dicho objetivo. De esta manera se puede escribir el fitness del agente a –en realidad de la RNA o grupo que lo controla– como:

$$Fitness(a) = (D-d)/D$$

Donde D representa la distancia máxima posible entre dos objetos en el entorno virtual –la diagonal del campo de prueba– y d la distancia existente entre el agente y el objetivo. De esta manera el fitness será 0 (cero) si el agente termina la simulación a la mayor distancia posible del objetivo –uno en una esquina del campo de prueba y el otro en la opuesta–, situación poco probable, y será 1 (uno) si el agente termina sobre el objetivo al finalizar la simulación.

Esta aproximación es demasiado simplista pues no considera la presencia de obstáculos. Sabiendo que para lograr alcanzar el objetivo, un agente debe sortear antes cualquier obstáculo que se interponga entre él y su objetivo, aunque esto signifique alejarse de él, podemos mejorar la función de fitness de la siguiente manera:

$$Fitness(a) = \begin{cases} (D-d)/D, & \text{si no existe interposición de obstáculo} \\ 0 & \text{,si existe interposición de obstáculo} \end{cases}$$

Esto aún no resuelve todo, pues es necesario animar a los agentes a que sorteen los obstáculos. La función de fitness anterior posee un efecto indeseable en algunos escenarios. Es probable que en la primera generación ningún agente logre sortear los obstáculos necesarios como para obtener un valor de aptitud distinto de cero. En tal caso el algoritmo genético se transforma en una simple búsqueda aleatoria hasta que algún agente logre un fitness mayor que cero. Tal situación fue resuelta agregando dos submetas al escenario a resolver. Estas submetas son completamente desconocidas por los agentes ya que no forman parte del entorno virtual durante la simulación, sólo se utilizan al concluir la misma para el cálculo del fitness. Veamos un ejemplo de un escenario para clarificar el concepto.

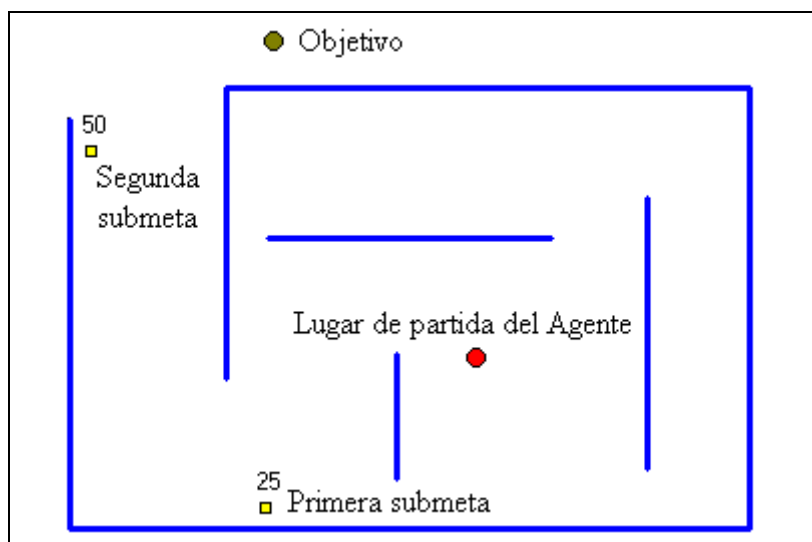


Ilustración 85 Escenario en el que pueden apreciarse dos submetas definidas. Los números 25 y 50 que identifican a la primera y segunda submeta respectivamente se refieren al monto del fitness con que contribuyen. El objetivo se considera que contribuye con un valor de 100.

Se desea que aquellos agentes que al terminar la simulación se encuentran con el camino libre hacia la primera submeta puedan recibir hasta 25 puntos de fitness –25 si se hallan justo sobre ella–, los que hayan logrado alcanzar un lugar desde el cual no existan obstáculos interponiéndose con la segunda submeta, aspiren a recibir hasta 50 puntos de fitness, y finalmente, aquellos que hayan logrado situarse en un punto desde el cual se encuentra libre de obstáculos el camino hacia el objetivo puedan llegar hasta los 100 puntos de fitness.

Sin embargo aún falta un último refinamiento. En el presente trabajo me he planteado evolucionar el comportamiento de “alcanzar el objetivo lo más rápidamente posible” y no el de “alcanzar y permanecer sobre el objetivo”. Por lo tanto aquel agente que logre alcanzar un objetivo debe recibir un fitness alto independientemente del lugar en que sea sorprendido al finalizar la simulación. Además se desea otorgar un valor más alto al agente que más rápido alcance el objetivo. Esto se puede implementar fácilmente de la siguiente forma: Si p es el número de pasos utilizados para alcanzar el objetivo, y P el número total de pasos de la simulación la expresión $(P-p)/P$ tomará el valor cero en caso de ser alcanzado en el último paso de la simulación y tomará el valor uno si la simulación comienza con el agente posicionado sobre el objetivo. Así se obtiene un valor entre 0 y 1 proporcional a la velocidad con que se ha alcanzado el objetivo.

Veamos entonces más formalmente cómo queda definida la función de fitness. Sea d_1 , d_2 y d las distancias a la primera submeta, segunda submeta y al objetivo respectivamente, D la distancia máxima posible entre dos objetos en el campo de prueba, p el número de pasos utilizados para alcanzar el objetivo, en caso de hacerlo, P el número total de pasos de la simulación. Y sean los predicados

$O(s_1)$ = “Se interponen obstáculos entre a y la primera submeta”

$O(s_2)$ = “Se interponen obstáculos entre a y la segunda submeta”

$O(ob)$ = “Se interponen obstáculos entre a y el objetivo”

$A(ob)$ = “Objetivo alcanzado”

La función de Fitness se define por casos de la siguiente manera:

Condición	<i>Fitness(a)</i>
Not $A(ob)$ And $O(s_1)$ And $O(s_2)$ And $O(ob)$	0
Not $A(ob)$ And Not $O(s_1)$ And $O(s_2)$ And $O(ob)$	$25(D-d_1)/D$
Not $A(ob)$ And Not $O(s_2)$ And $O(ob)$	$25 + 25(D-d_2)/D$
Not $A(ob)$ And Not $O(ob)$	$50 + 50(D-d)/D$
$A(ob)$	$100 + 50(P-p)/P$

Así el fitness será un valor real del intervalo [0..150]. El siguiente gráfico muestra la situación de los agentes en distintas franjas de valores de fitness.

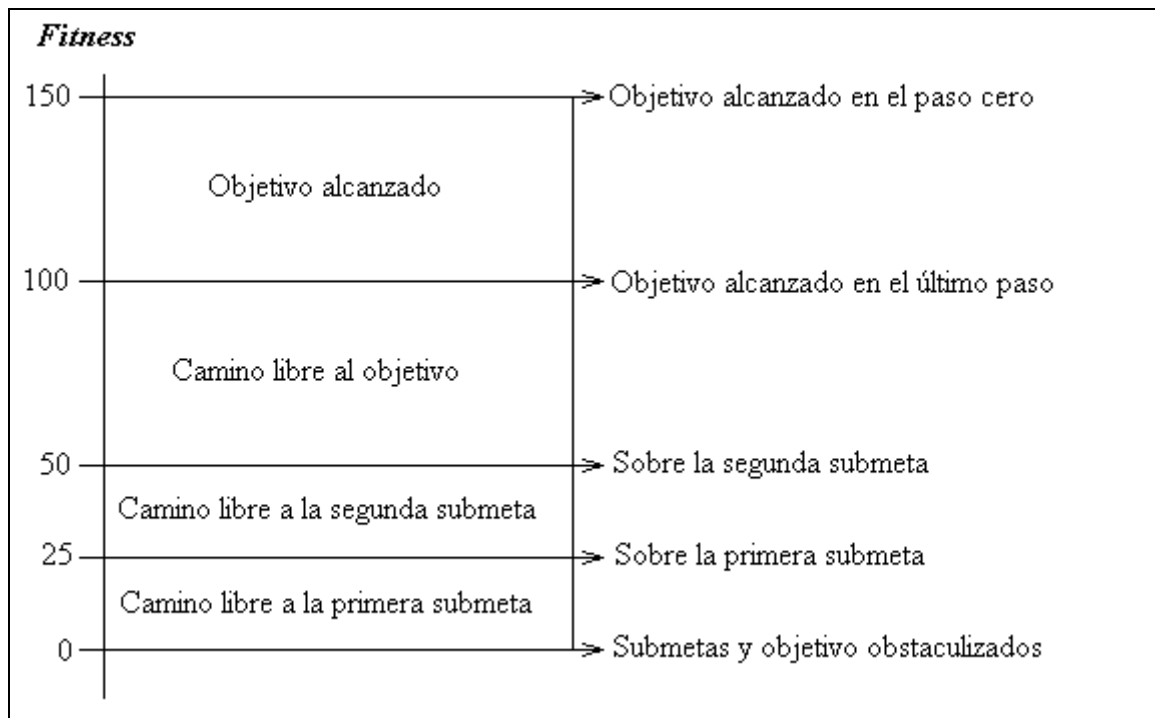


Ilustración 86 Esquema de asignación de fitness donde puede observarse la relación entre el valor de aptitud y la situación alcanzada por el agente durante la simulación

Tipos de Redes Neuronales Artificiales implementadas

Ya se ha mencionado en otro capítulo que existen ciertos inconvenientes asociados a las RNA tradicionales que dependen fuertemente de la experiencia y/o habilidades humanas para ser superados exitosamente. Sin lugar a dudas, uno de los más importantes, se relaciona con la dificultad de encontrar una arquitectura neuronal adecuada para la resolución de una tarea particular. En este aspecto la Neuroevolución ha demostrado que puede convertirse en una alternativa viable de diseño automático de arquitecturas con mejor desempeño que los algoritmos constructivos y destructivos que se han comentado muy brevemente en el apartado “Evolución de arquitectura” –pág. 161–.

Por arquitectura se entiende básicamente la topología expresada a través de la conectividad, pero también debe incluirse la función de transferencia y término de tendencia, si existe, de cada nodo perteneciente a la red.

La elección de una arquitectura apropiada resulta decisiva en la Computación Neuronal pues su efecto impacta directamente en las capacidades de procesamiento de la red. Una tarea compleja, seguramente no podrá llevarse a cabo por una red neuronal de baja conectividad o con todas sus funciones de transferencias lineales. En el otro extremo una red de muy alta conectividad puede sobreajustarse produciendo una deficiente capacidad de generalización.

Aunque básicamente el presente trabajo posee como objetivo la implementación, evaluación y comparación de distintos –métodos– o estrategias– de neuroevolución, se han implementado más de un tipo de RNA para realizar los ensayos neuroevolutivos. Estas clases de RNA basan sus diferencias principalmente en aspectos de arquitectura neuronal.

En lo que concierne a la conectividad debe mencionarse que se han implementado los dos esquemas existentes en la clasificación de topologías neuronales: Redes feedforward y Redes recurrentes –también denominadas recursivas o redes feedback–.

Asimismo se han implementado dos variantes en relación a la función de transferencia. En el primer tipo, no se la somete a evolución, sino que se utiliza la misma función sigmoide en todos los nodos de la red. Por el contrario en el otro tipo cada neurona podrá aplicar una de entre cuatro funciones sigmoides distintas, codificadas en el cromosoma y sometidas a control evolutivo.

Finalmente, dos alternativas más en relación a la existencia o no de término de tendencia en cada neurona de la red completan el conjunto total de tipos de RNA implementados. Así de la combinación de estas tres características –con dos variantes cada una– resulta la posibilidad de evolucionar $2^3=8$ tipos diferentes de RNA.

1. RNA FeedForward, sin término de tendencia, con sigmoide fija.
2. RNA FeedForward, sin término de tendencia, multisigmoide.
3. RNA FeedForward, con término de tendencia, con sigmoide fija.
4. RNA FeedForward, con término de tendencia, multisigmoide.
5. RNA Recurrentes, sin término de tendencia, con sigmoide fija.
6. RNA Recurrentes, sin término de tendencia, multisigmoide.
7. RNA Recurrentes, con término de tendencia, con sigmoide fija.
8. RNA Recurrentes, con término de tendencia, multisigmoide.

Se utilizarán dos neuronas de salida en casi todos los casos, representando el valor de cada una de las componentes del vector movimiento que determina la acción del agente. Sólo en el método GNE serán necesarias redes neuronales con 3 neuronas de salida. Explicaré la utilización de la tercera neurona de salida, más adelante en este capítulo, cuando se aborde esta estrategia evolutiva bajo el título de “Grupos Neuronales Evolutivos –GNE–”, en la página 224 .

La cantidad de neuronas de entrada es un valor parametrizable que puede elegirse en el momento de la realización de una prueba evolutiva. No obstante se requiere un mínimo de 10 neuronas para cumplir con el esquema que ya se presentó en este capítulo. De estas 10 neuronas 4 de ellas se conectan a los sensores de objetivos, 4 a los sensores de obstáculos, una al sensor de colisión y la otra recibe el tiempo de simulación.

La elección de un número mayor de neuronas de entrada determina una mayor cantidad de sensores involucrados en la detección de objetivos. Así, si se eligen evolucionar redes con 20 neuronas de entrada estaremos destinando 14 de ellas a detectar la presencia de un objetivo en un sector de ángulo $\pi/7$ radianes alrededor del agente.

La cantidad de neuronas ocultas también es un valor que se elige en el momento de la realización de un ensayo neuroevolutivo. Debo aclarar que al definir la cantidad de neuronas de entrada u ocultas, se están eligiendo la cantidad máxima y no el número exacto de ellas. Esto es debido a que el algoritmo genético puede generar redes con neuronas totalmente desconectadas o con todas sus conexiones con valor de peso igual a cero. En tal caso, estas neuronas no pueden contribuir en absoluto al procesamiento de la red y se consideran como no pertenecientes a la misma.

No obstante, se debe aclarar que para favorecer la evolución de redes con menor cantidad de neuronas del máximo especificado, es necesario incluir alguna valoración sobre la conectividad en la función de fitness que no se realiza en el presente trabajo, focalizado en la comparación de distintos métodos neuroevolutivos más que en la simplificación estructural de las RN halladas.

RNA Feedforward

Como se mencionara en un capítulo anterior, las redes neuronales del tipo feedforward, o de propagación hacia delante son aquellas en las que las salidas de las neuronas de una capa sólo se propagan hacia la capa siguiente. Un ejemplo de este tipo de redes lo constituye el Perceptrón multicapa que ganara popularidad luego del redescubrimiento del algoritmo de Backpropagation¹.

En el presente trabajo se han implementado redes del tipo feedforward de dos capas –además de la capa de entrada– localmente conectadas. Recordemos que la clasificación de arquitecturas respecto del número de conexiones reconoce dos tipos de RNA: aquellas totalmente conectadas, donde la salida de las neuronas de una capa se conecta con todos los nodos de la capa siguiente, y las localmente conectadas donde la salida de las neuronas de una capa se conecta sólo con un cierto número de neuronas –una región– de la capa siguiente. No se asume aquí ninguna restricción respecto del tamaño de dicha región, pudiendo ser tan amplia como todas las neuronas de la capa siguiente. De esta manera se pueden considerar a las RNA totalmente conectadas como un subconjunto propio de las RNA localmente conectadas.

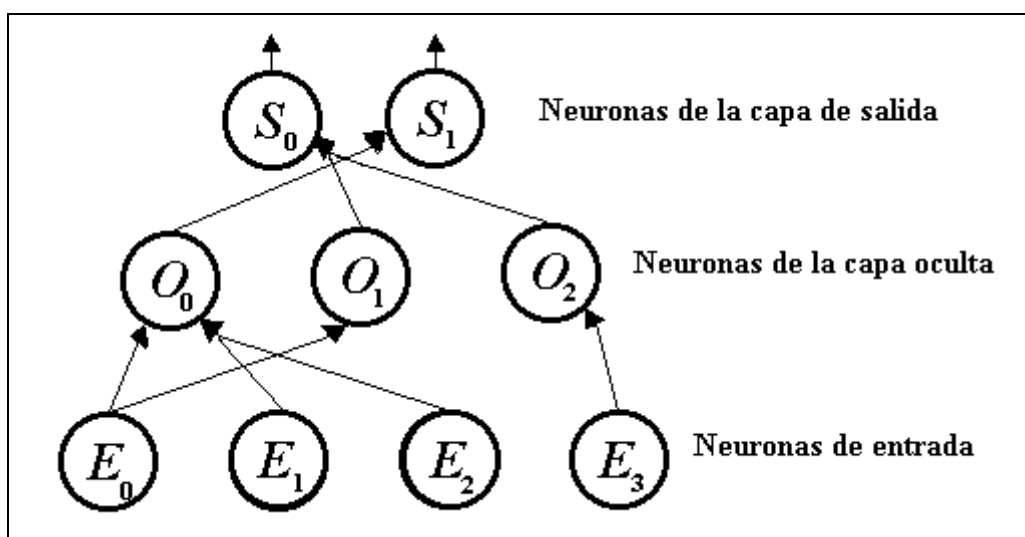


Ilustración 87 Ejemplo de RNA FeedForward de dos capas. Puede observarse que se trata de una red localmente conectada y la evolución podrá generar especímenes con distinto número de conexiones. Debe recordarse que aunque no existe un claro consenso en la literatura especializada, para designar el tipo de RNA a la que pertenece esta arquitectura no se contabiliza la capa de entrada. Así estamos en presencia de una red FeedForward de dos capas.

RNA Recurrentes

Las redes neuronales recurrentes son aquellas donde se permiten conexiones de salida que retroalimentan neuronas anteriores o del mismo nivel dando lugar a la existencia de ciclos en la estructura. Aquí se hace más difícil hablar de capas pues los nodos y conexiones se disponen a modo de grafo genérico con sólo algunas restricciones. Debe notarse que este tipo

¹ Para mayores precisiones pueden consultarse los siguientes apartados: “Perceptrón multicapa” en la pág. 136, “Backpropagation” en la pág. 141 y “Arquitectura neuronal” en la pág. 151.

de redes corresponde a una clase más general que incluye de manera propia al conjunto de redes feedforward.

Las redes recurrentes suelen ser más lentas que las feedforward y consumir mayor cantidad de recursos de cómputo. En contraste suelen ser más potentes porque en el sistema circulan datos relacionados con entradas presentadas a la red en instantes anteriores. De esta manera la red responde a las entradas influenciada por la historia inmediata anterior de estimulaciones recibidas.

A modo de ejemplo comentaré que experimentos sobre comportamiento de “captura de presa” han sido realizados con éxito utilizando redes recurrentes por Gómez y Miikkulainen en [Gomez97] logrando que el agente –depredador– recuerde la dirección por la que la presa escapó a su radio de visión para comenzar la persecución. Demostraron así que este tipo de redes puede implementar memoria a corto plazo. Para ello utilizaron RNA recurrentes con neuronas totalmente interconectadas.

A diferencia de la investigación antes mencionada, en el presente trabajo se han implementado redes recurrentes localmente conectadas. Es decir, el modelo aquí es más genérico, dejando al algoritmo genético la responsabilidad de decidir “libremente” cómo conectar las neuronas de estas redes. Así no se desecha la posibilidad de que la evolución genere estructuras que resuelvan satisfactoriamente el problema y al mismo tiempo ahorre conexiones innecesarias –y con ello recursos de cómputo–.

Por lo tanto, en el presente trabajo se define la topología de las redes recurrentes como un grafo dirigido dónde los nodos representan neuronas y los link sus conexiones, con las siguientes restricciones:

1. Deben existir al menos una neurona de entrada y una neurona de salida.
2. No pueden existir conexiones hacia neuronas de entrada.

Estas restricciones se imponen para dejar fuera del alcance de la definición a ciertas estructuras neuronales carentes de sentido o inútiles. Una red sin neuronas de entrada o de salida es francamente una red sin ninguna utilidad. De nada sirve una gran capacidad de procesamiento si no hay forma de introducir la información a procesar. Semejante ineficacia también puede conseguirse si no existe ninguna comunicación hacia el exterior por la ausencia total de neuronas de salida.

Como ya se mencionara, las neuronas de entrada no pueden considerarse como verdaderas neuronas artificiales y no realizan procesamiento alguno. Su función es la de un subsistema de entrada por el cual la red se nutre de información externa. Así una conexión hacia una neurona de entrada carece de sentido y la evitaremos.

La siguiente ilustración presenta algunos ejemplos de redes que se ajustan a estas restricciones.

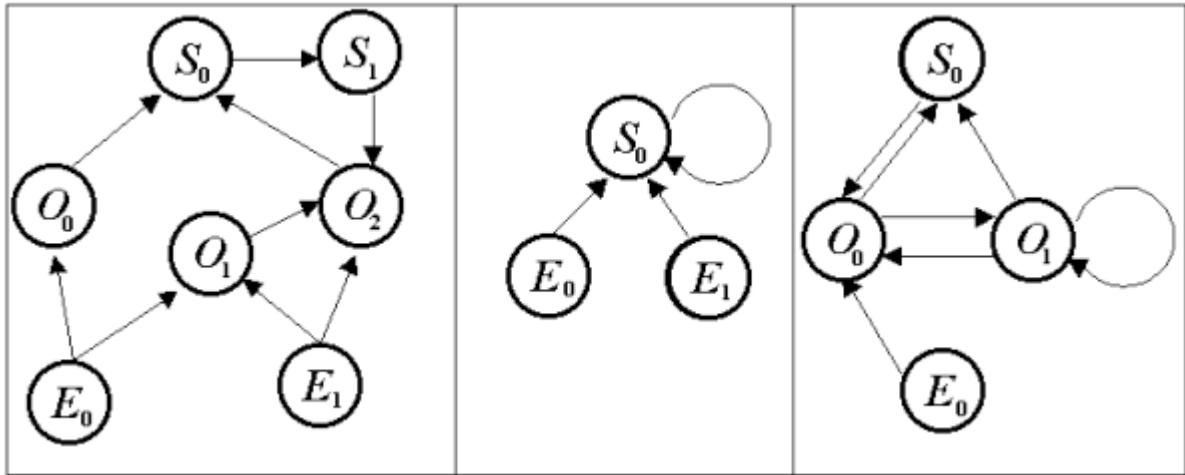


Ilustración 88 Ejemplo de redes neuronales recurrentes válidas conforme a la definición dada.

Sin embargo los siguientes ejemplos de redes neuronales recurrentes no pueden considerarse válidos pues no cumplen con alguna de las restricciones que se han impuesto.

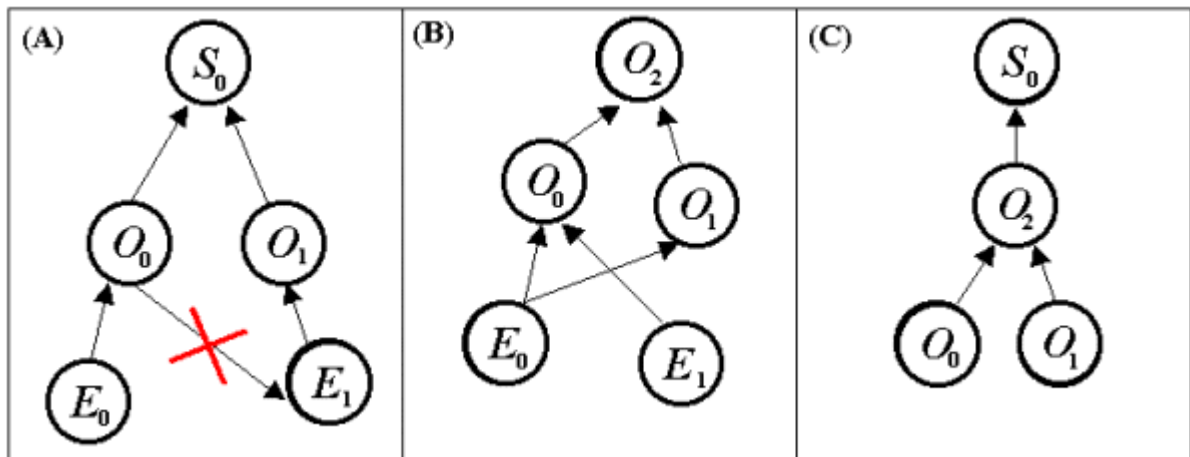


Ilustración 89 Ninguna de las tres estructuras presentadas puede considerarse una red neuronal recurrente válida. Como se puede apreciar, en el caso (A) existe una conexión inválida a una neurona de entrada, lo que carece de sentido pues ya se ha mencionado que estas neuronas no realizan procesamiento alguno y sólo sirven para ingresar valores de estímulo a la red. El caso (B) es una red sin neuronas de salida por lo tanto inútil debido a la incapacidad de comunicar ningún resultado. En el otro extremo, en el caso (C) se observa otro tipo inútil de estructura en la que no existen neuronas de entrada y por lo tanto se hace imposible estimularla.

Ya se ha mencionado el hecho que la evolución podrá generar redes con menor cantidad de neuronas o conexiones cuya cota máxima debe especificarse como parámetro para la estrategia evolutiva elegida. Esto sucede cuando se generan estructuras con una o más neuronas desconectadas o que integran algún circuito que no tiene influencia en la salida de la red. Para clarificar la situación veamos el siguiente gráfico.

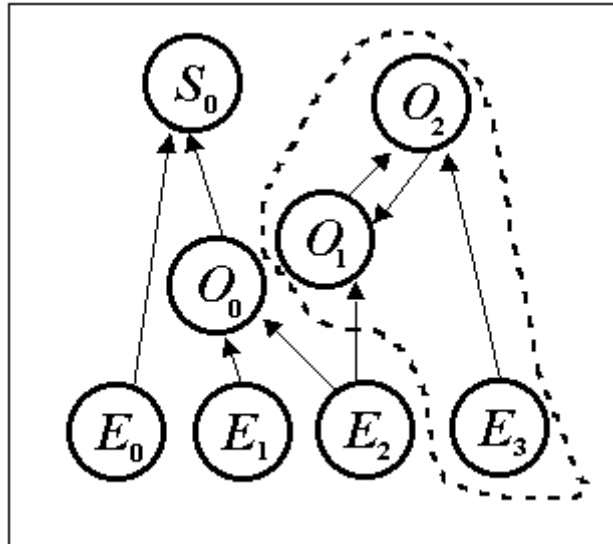


Ilustración 90 Este es el caso de una red neuronal válida con un conjunto de neuronas inútiles. Puede observarse que las neuronas contenidas en el área definida por la línea punteada no realizan ninguna contribución a la salida de la red y pueden considerarse como ajenas a la estructura. Esta situación lejos de ser indeseable, si es que no degrada el fitness mejora el rendimiento de las soluciones alcanzadas al reducir la complejidad de la red y con ello el consumo de recursos computacionales.

Para finalizar este tema haré un breve comentario acerca de la manera de evaluar una red recurrente. Extrañamente no he encontrado precisiones al respecto en la bibliografía consultada.

El inconveniente que se presenta es determinar el orden adecuado en que deben ser evaluadas las neuronas pertenecientes a la red y cómo resolver la presencia de ciclos en la estructura. Este problema no existe en las redes Feedforward donde claramente deben evaluarse todas las neuronas –posiblemente en paralelo– de una capa antes de evaluar la siguiente.

La evaluación en una primera fase de todas las neuronas de una red, para luego pasar a la fase de propagación de resultados a través de las conexiones, que en principio observé como alternativa viable, enseguida demostró no adecuarse al modelo de red neuronal pretendido provocando un efecto indeseable de retardo de respuesta a un estímulo dado. Para simplificar la explicación supongamos como ejemplo una red organizada en capas. La información detectada y presentada a la red por las neuronas de entrada podrá afectar la salida de la misma sólo luego de varias evaluaciones –tantas como capas tenga la red– que irán empujando el flujo de información hacia las capas superiores.

En una red recurrente la evaluación de una neurona puede requerir como entrada la salida de otra neurona a quién ella misma estimula.

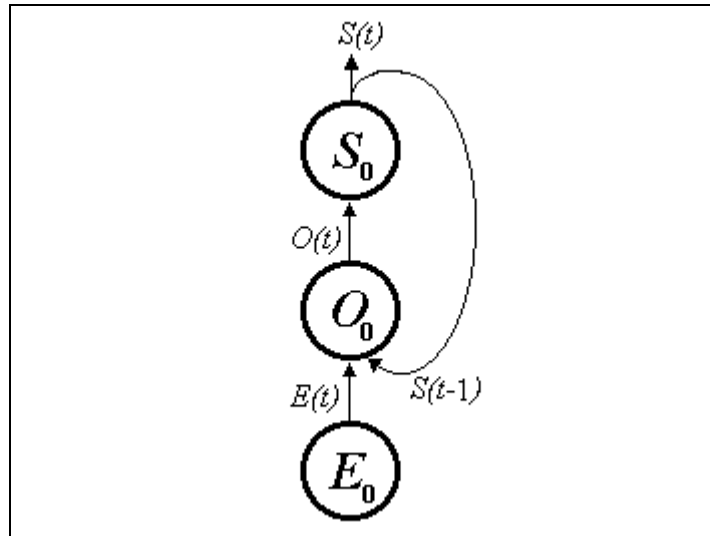


Ilustración 91 Ejemplo sencillo de red recurrente donde se aprecia un ciclo.

Al colocar un valor en la neurona de entrada E de la red presentada en la ilustración anterior, ésta puede evaluarse siguiendo una simple regla: *procesar cada neurona en el momento que tenga todas sus entradas disponibles*. Así, para el instante t , la señal de entrada $E(t)$ llega a la neurona O que en el instante anterior ya había recibido la señal de retroalimentación $S(t-1)$. Ahora con su entrada completa puede evaluarse produciendo la señal $O(t)$ que estimula la neurona S . Finalmente $S(t)$ es producida por la evaluación de S , que no sólo representa la respuesta de la red a la entrada $E(t)$, sino que además se prepara para estimular en el próximo instante $(t+1)$ a la neurona O .

Para la primera evaluación de la red ($t=0$), se considera que las conexiones recursivas de retroalimentación están comunicando el valor cero.

En este sencillo ejemplo, queda claro cuál es el orden en que deben procesarse las neuronas, no obstante otros pueden resultar más ambiguos.

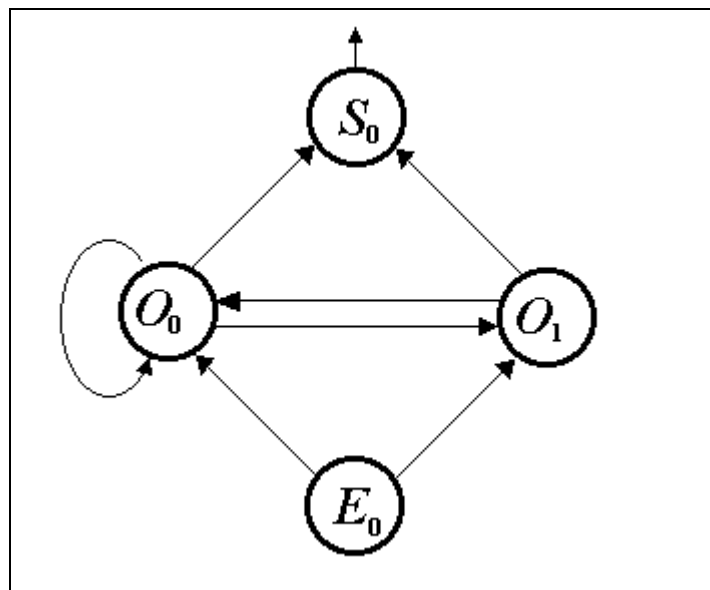


Ilustración 92 En esta estructura no queda especificado de manera única cuál deba ser el orden en que se evalúen los nodos ocultos.

Obsérvese que en la ilustración anterior no existe criterio a simple vista que nos diga qué neurona oculta debe evaluarse primero, ya que cada una de ellas recibe estímulo y a la vez excita a la otra. La elección del orden de evaluación afecta el procesamiento de la red produciendo resultados distintos para ordenamientos distintos.

En situaciones como esta, donde más de un orden es factible, no hay más alternativa que fijar uno de ellos para evaluar la red siempre de la misma manera. En el presente trabajo se determina el orden de evaluación de la siguiente manera.

En este algoritmo no entran en juego las neuronas de entrada que siempre serán evaluadas en primer lugar. En un principio se colocan en una cola las neuronas de salida. Luego se recorre la misma de adelante hacia atrás consultando a cada neurona por la identidad de sus congéneres que la estimulan directamente. Estas neuronas presinápticas de aquella consultada son agregadas al final de la cola, en caso de no encontrarse ya encoladas. Si más de una debe ser agregada, por estimular a la misma neurona, se utiliza el subíndice de las mismas para decidir cuál agregar primero. Al procesar la última neurona de la cola se habrá conseguido el orden de evaluación deseado que resulta establecido por el recorrido de la cola de atrás hacia delante.

De esta manera también se logra que aquellas neuronas que no contribuyen a la salida de la red no estén presente en el ordenamiento y por lo tanto no serán evaluadas.

Conexión de tendencia

Más adelante en este mismo capítulo presentaré los métodos neuroevolutivos implementados. Para su realización me he apoyado particularmente en los desarrollos publicados por el Departamento de Ciencias de la Computación de la Universidad de Texas [*Moriarty96*] [*Moriarty97_1*] [*Moriarty97_2*] [*Gomez97*] [*Gomez99*] [*PerezBergquist01*] [*HanYong01*] [*Bruce01*]. He podido implementar SANE y ESP –dos importantes estrategias neuroevolutivas desarrolladas en este centro de investigación– con bastante fidelidad, gracias a la documentación detallada conseguida, especialmente sobre SANE que se explica con mucha precisión en los artículos de Moriarty. Pero estos métodos, no contemplan conexión de tendencia en las estructuras evolucionadas.

En el presente trabajo, se han extendido estos métodos en varios aspectos, uno de los cuales se refiere a la capacidad para soportar el término de tendencia del que no se hace mención alguna en los artículos de estos investigadores.

Es una realidad que, en la mayoría de los casos, se utiliza el término de tendencia si la comprobación empírica sugiere que con ello se consigue alguna mejora con respecto a la no utilización del mismo. No he podido encontrar bibliografía con un buen análisis acerca de la utilidad de la conexión de tendencia. Me refiero a algún estudio de naturaleza teórica que determine cuando resulta beneficioso usarla y cuando no. No obstante, varios indicios acerca de su conveniencia han sido ya expuestos en este trabajo.

En el capítulo anterior se han brindado algunas precisiones acerca de cómo surgió la utilización de la conexión de tendencia por el beneficio obtenido al someter a entrenamiento el umbral de la función de transferencia de la neurona de McCulloch y Pitts. Mejor que fijar un valor adecuado para este umbral, es incluirlo como el peso de esta nueva conexión. De esta forma, al igual que cualquier peso sináptico, su valor será determinado adecuadamente por el entrenamiento –ver “Perceptrón” en la pág. 127–. Allí se expuso que un Perceptrón simple no podría clasificar correctamente ciertas funciones, tan sencillas como el AND lógico, sin un valor de umbral adecuado o, en su defecto, la conexión de tendencia, pues la recta de decisión

utilizada para la clasificación debe desplazarse del origen de coordenadas –ver Ilustración 52 en la pág. 128–.

Cuando se analizó el Perceptrón Multicapa en la pág. 136 se explicó que con dos capas alcanza para que el dispositivo sea capaz de definir cualquier región de decisión convexa en el espacio. Esto se consigue si la neurona del nivel de salida implementa la función lógica AND de los semiplanos formados por cada neurona en el nivel anterior. Esta región de decisión será una región convexa con un número de lados a lo sumo igual al número de neuronas de la capa oculta. Está claro que para conseguir cualquier región arbitraria, es necesario que las rectas de decisión de cada neurona posean la libertad de no pasar por el origen, libertad que se asegura si utilizamos conexión de tendencia.

Como último dato a favor de la conexión de tendencia se debe mencionar que la red Backpropagation, una de las más exitosas del paradigma RNA, contempla su utilización.

Aunque en los desarrollos de SANE y ESP presentados por Moriarty y Gómez respectivamente parece ser que la utilización de conexión de tendencia no ha sido necesaria, en función de lo antes mencionado no me parece aconsejable descartarla completamente.

Finalmente, para este trabajo, se han implementado redes neuronales con conexión de tendencia en cada una de las neuronas ocultas y de salida.

Un efecto de la conexión de tendencia puede observarse en la ilustración siguiente, donde una neurona oculta que no es estimulada directa ni indirectamente por la entrada no puede considerarse inútil pues aportará una constante independiente de la entrada proveniente de la aplicación de la función de transferencia al valor de su conexión de tendencia.

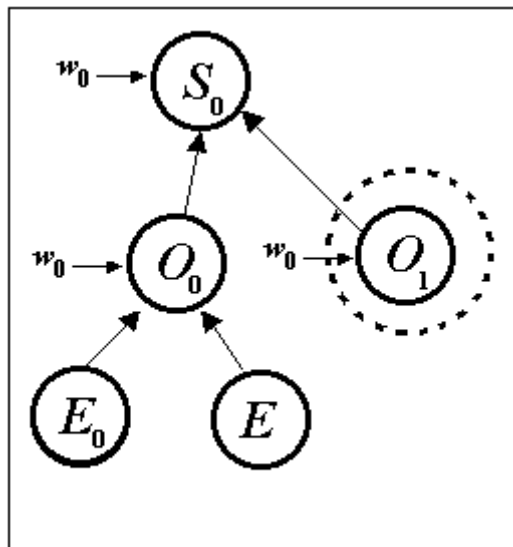


Ilustración 93 La neurona oculta encerrada por la línea punteada, aportará en cada evaluación de la red el valor de la función de transferencia aplicada al valor de su conexión de tendencia, contribuyendo a la salida.

Múltiples funciones de transferencia

En el capítulo anterior se comentó que algunos investigadores han conseguido redes neuronales artificiales evolucionando topología y función de transferencia al mismo tiempo, reportando buenos resultados. Existen estudios donde la función completa es codificada en el cromosoma y sometida a evolución. Otros parten de un conjunto de funciones predefinidas y

en el cromosoma se codifica por cada neurona, o capa, una etiqueta que identifica cual de ellas debe utilizarse.

En todos los tipos de RNA implementados en este trabajo, las neuronas ocultas y de salida utilizan la conocida función de transferencia sigmoide logística. Se ha incorporado la posibilidad de que la evolución determine para cada neurona qué función utilizar de entre cuatro variantes de sigmoide implementadas.

De esta manera el algoritmo genético podrá variar la sensibilidad de una neurona a la estimulación recibida simplemente por medio del cambio de la función de transferencia de la misma, en lugar de corregir los pesos de todas sus conexiones de entrada.

Las cuatro variantes de sigmoide implementadas son las siguientes:

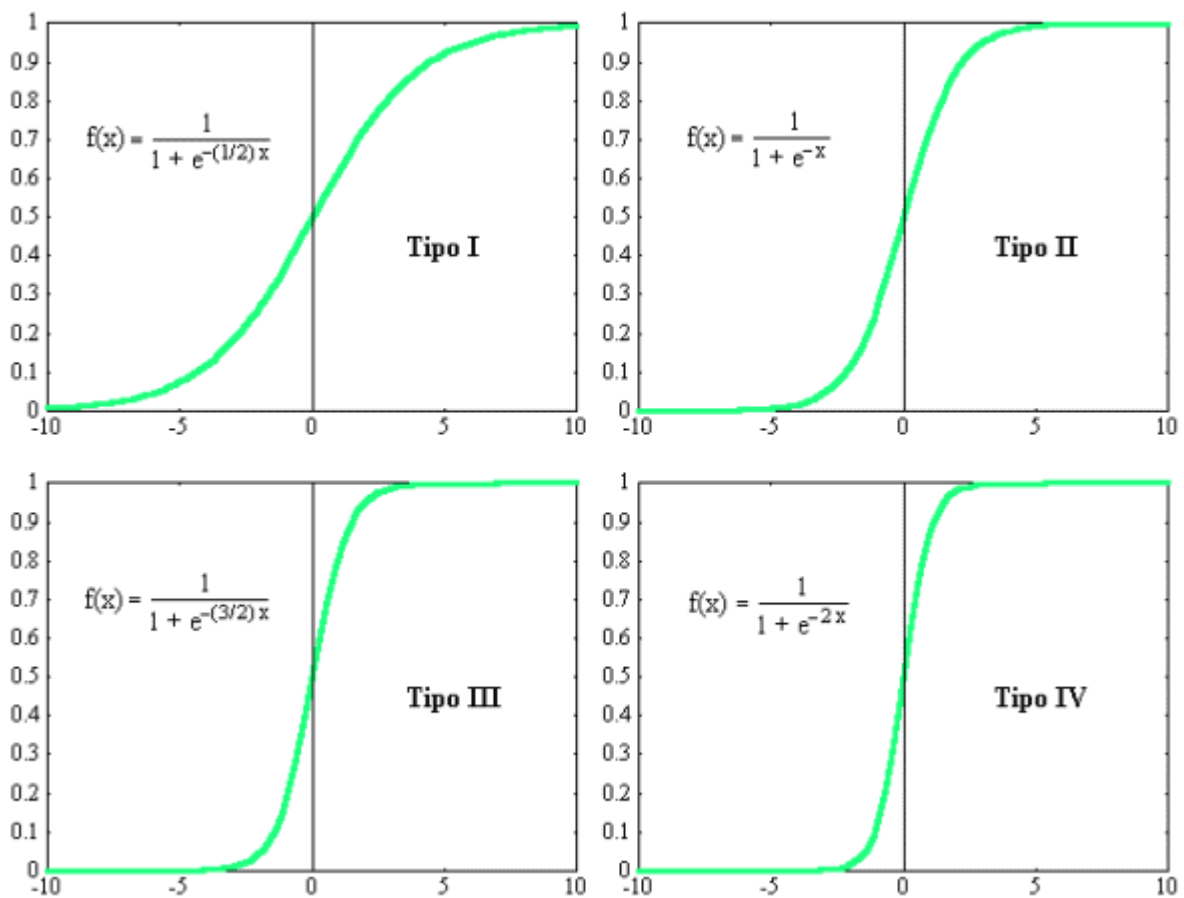


Ilustración 94 Los cuatro tipos de funciones sigmoides implementadas se consiguen cambiando el coeficiente que multiplica la variable x

Puede observarse que cualquiera que sea el tipo de sigmoide que se aplique a la activación de una neurona su salida siempre estará acotada al intervalo $[0..1]$. Por lo tanto la salida de las redes neuronales puede verse como un vector de dos dimensiones (x_1, x_2) con $x_i \in [0..1]$.

La manera de transformar esta salida en un desplazamiento válido ya se ha mencionado en el apartado “Las simulación” en la pág. 174.

A estas Redes Neuronales que admiten las cuatro variantes de función de transferencia vistas, las llamaré de aquí en más Redes Neuronales multisigmoide, denominación impuesta sólo a los efectos de nombrarlas en el presente trabajo.

Estrategias de Neuroevolución implementadas

He implementado seis distintos métodos de neuroevolución y en todos ellos se ha tomado de entre el conjunto de algoritmos evolucionarios a los genéticos para llevarlas a cabo. Esto presupone el uso del crossover como principal operador genético y la representación binaria para la codificación de los genes en el cromosoma. No obstante se han realizado algunas combinaciones con representación real en cromosomas como detallaré más adelante al exponer SANE, ESP con población de Blueprints y GNE.

Brevemente, recordemos que un Algoritmo Genético comienza con una población de cromosomas, habitualmente generados al azar, e itera hasta cumplir alguna condición de terminación pasando por las etapas de selección, reproducción y reemplazo en cada lazo. En todos los casos se ha utilizado como criterio de finalización el cumplimiento de un cierto número de iteraciones, o lo que es lo mismo decir, la evolución se prolongará por un número prefijado de generaciones.

La selección, proceso por el cuál se consigue el subconjunto de individuos de la población para la reproducción, siempre se realiza en base a la función de fitness que califica las potenciales soluciones para que, aquellas con valores más altos, tengan mayor probabilidad de reproducirse más veces.

En la realización del presente trabajo se han utilizado dos técnicas de selección distintas de acuerdo a la estrategia neuroevolutiva implementada. La más utilizada será una estrategia de selección basada en ranking que se utiliza en SANE, ESP, ESP con población de Blueprints y GNE, que explicaré en detalle más adelante. La otra forma de conseguir el conjunto de parentales será por medio del conocido método de “Rueda de ruleta” popularizado por Goldberg [Goldberg89].

La reproducción en todos los casos se llevará a cabo por la utilización de los dos operadores genéticos más importantes: El crossover –de un punto– y la mutación. La probabilidad de aplicación de estos operadores dependerá de la estrategia neuroevolutiva. El software desarrollado para el presente trabajo, permite que el usuario defina estos valores de probabilidad que utilizará el Algoritmo Genético, para el caso de las estrategias que he denominado “Neuroevolución simple” y “Neuroevolución simple con Algoritmo Genético Elitista”. Las demás pueden considerarse variantes o extensiones de SANE y por lo tanto es preferible utilizar los valores fijados por este método.

Como veremos más adelante SANE es bastante particular en muchos sentidos y la probabilidad de aplicación de los operadores genéticos no es la excepción. El crossover se aplica siempre –probabilidad 1.0– mientras que existen dos mecanismos de mutación que, contradiciendo las pautas para el diseño de un Algoritmo Genético eficiente, se aplican con muy alta probabilidad. Más adelante se darán precisiones pero debe mencionarse que ello es posible porque en realidad se trata de mutaciones controladas.

Remito al lector al capítulo “Computación Evolutiva” que comienza en la pág. 43 para más detalles sobre los algoritmos evolucionarios.

Sin más preámbulo comenzaré con la descripción de cada una de las estrategias neuroevolutivas desarrolladas.

Neuroevolución Simple

He implementado una estrategia simple de neuroevolución para obtener un parámetro de referencia en el análisis comparativo con otras más complejas.

Por “simple” debe entenderse tradicional, sin ningún tipo de innovación, donde se utiliza el Algoritmo Genético canónico para evolucionar una población de Redes Neuronales que controlen los agentes en su entorno virtual 2D. Aquella RNA con el mayor fitness obtenido en la evolución se considera la solución buscada.

Aquí se presentará la codificación genética elegida que también se utilizará en la próxima estrategia evolutiva que se analizará. Se ha intentado lograr una codificación que se acerque a la que utilizarán las restantes cuatro estrategias que utilizan más de una población – típicamente de neuronas y de redes neuronales– que coevolucionan para hallar la solución.

La intención de tener una codificación genética parecida en todos los métodos neuroevolutivos se fundamenta en la necesidad de minimizar su impacto en las diferencias de performance medidas durante el análisis comparativo. En otras palabras, queremos medir virtudes y defectos de la “esencia” de los distintos métodos y no de una codificación particular.

Algoritmo Genético utilizado

Para implementar esta estrategia de neuroevolución se ha utilizado un algoritmo genético simple basado en el ejemplo presentado en el capítulo Computación Evolutiva bajo el subtítulo de “Ejemplo de Algoritmo Genético Simple” en la pág. 51.

Veamos esquemáticamente cómo funciona el algoritmo elegido.

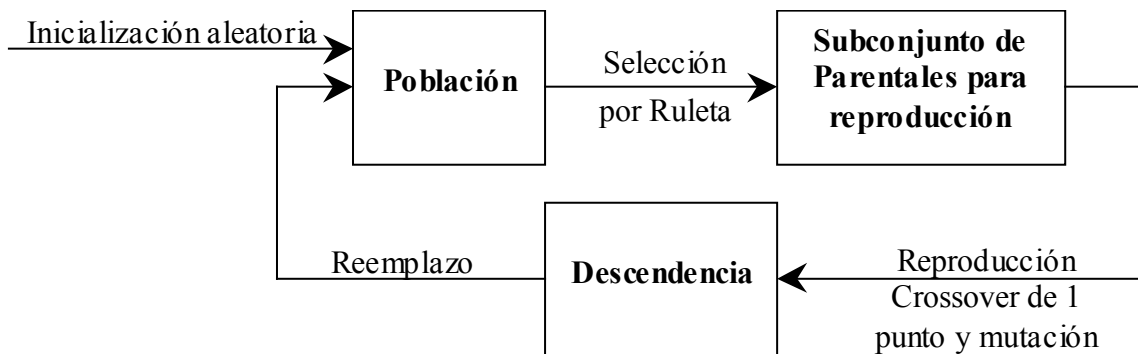


Ilustración 95 Esquema de Algoritmo Genético simple.

Básicamente se trata de un Algoritmo Genético con población de tamaño fijo, no elitista, donde la siguiente generación se consigue por reemplazo total de la población. Los reemplazantes son aquellos descendientes de un subconjunto de parentales elegidos por el método de la “rueda de ruleta”. El algoritmo se detiene cuando se ha alcanzado y evaluado el número de generaciones previamente fijado.

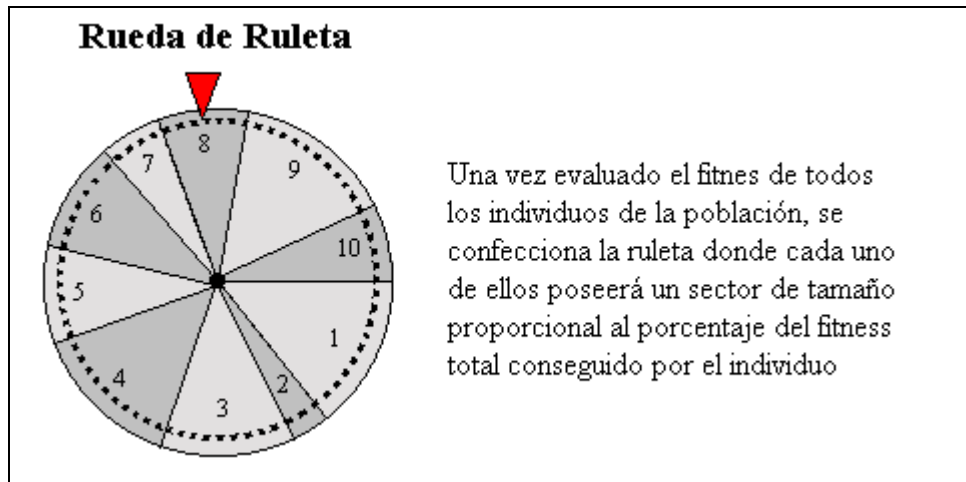


Ilustración 96 La Rueda de ruleta constituye un método de selección proporcional al valor de fitness sencillo de implementar.

Sin entrar en demasiado detalle –remítase el lector al capítulo “Computación Evolutiva” para más precisiones– recordemos que el proceso de selección y reproducción utilizando la ruleta se lleva a cabo de la siguiente manera:

1. Se acciona dos veces el mecanismo para elegir así dos individuos.
2. Se obtienen exactamente dos descendientes por aplicación de crossover y mutación entre los padres elegidos.
3. Se repiten los pasos 1-2 hasta obtener tantos descendientes como cantidad de individuos haya en la población –se asume número par de individuos–.
4. Se reemplaza completamente la población por la nueva descendencia producida.

En el punto 2 entra en juego la probabilidad de aplicación de los operadores genéticos – parámetro definido previamente por el usuario–. Debe tenerse en cuenta que es usual, y así se hizo en la implementación del software que acompaña el presente trabajo, definir la probabilidad de aplicar crossover por cromosoma completo, pero la probabilidad de aplicar mutación por bit –o alelo–.

Por ejemplo, una probabilidad de aplicación de crossover de 0.8 y de aplicación de mutación de 0.001 –valores dentro del rango normalmente utilizado– significa que de cada 10 parejas seleccionadas se cruzarán probablemente a 8 de ellas para obtener sus descendientes, los descendientes de las otras 2 serán simplemente copias de sus padres. La mutación se aplicará a todos los descendientes recorriendo sus cromosomas para alterar un bit a razón de uno cada 1000.

Codificación

He mencionado en otro capítulo que Holland y Goldberg aconsejan utilizar una codificación con alfabeto de baja cardinalidad, en lo posible binario. Holland argumentó que la codificación binaria posee un grado más elevado de “paralelismo implícito” porque permite manejar un número mayor de esquemas¹. De esta manera es preferible tener muchos genes

¹ Para más información ver “Fundamento teórico” en la pág. 59 y “Codificación” en la pág. 65, ambos apartados pertenecen al capítulo “Computación Evolutiva”

que puedan tomar unos pocos valores –pocos alelos–, que el caso contrario, es decir pocos genes pero muchos alelos por gen. A modo de comentario, la Naturaleza también utiliza un alfabeto de baja cardinalidad en el genotipo de los organismos ya que lo usual es encontrar cromosomas con muchos genes y pocos alelos para cada gen.

Siguiendo a Holland y a su discípulo se ha elegido la representación tradicionalmente usada por los Algoritmos Genéticos en la codificación de las posibles soluciones dentro de cromosomas, es decir, la representación binaria con alfabeto $\{0,1\}$.

Existen muchas maneras en que se podría codificar una Red Neuronal dentro de un cromosoma. Por ejemplo podemos utilizar una codificación posicional, donde los genes quedan determinados por la posición relativa dentro del cromosoma.

Veamos una primera aproximación a un esquema de codificación del cromosoma para una RNA recurrente, multisigmoide, con conexión de tendencia, con p neuronas de entrada q neuronas ocultas y r neuronas de salida.

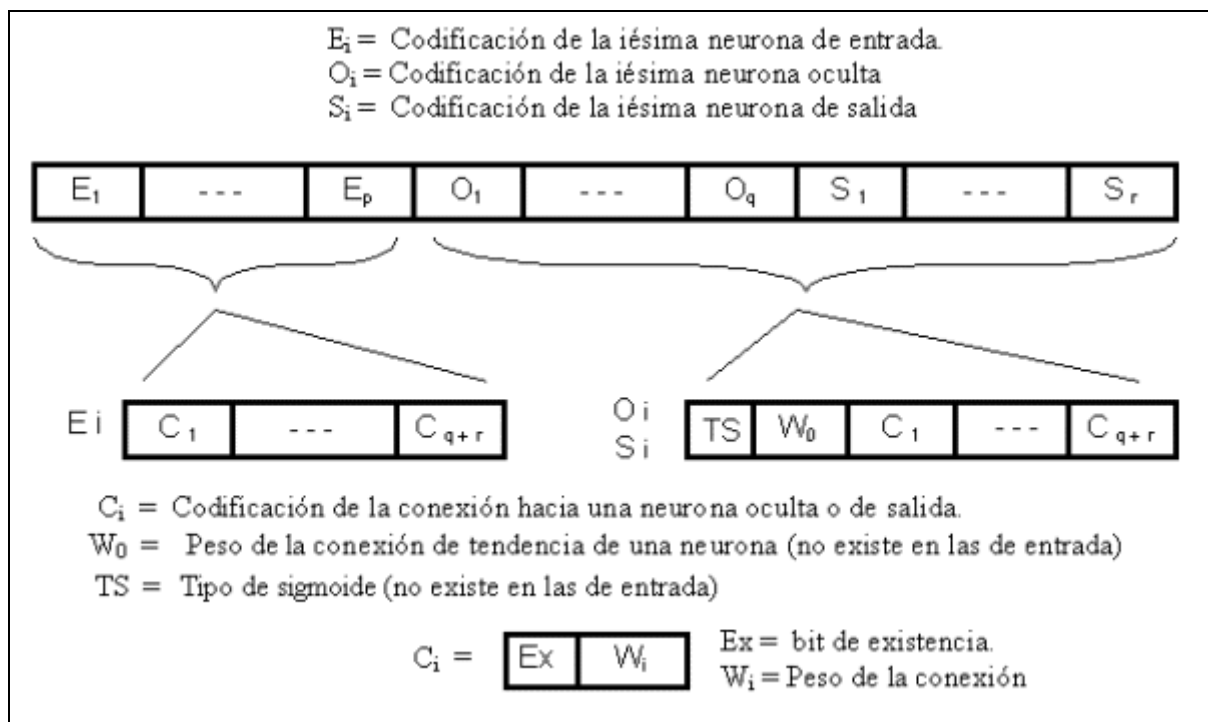


Ilustración 97 Ejemplo de codificación genética de una Red Neuronal recurrente, multisigmoide, con conexión de tendencia, con p neuronas de entrada q neuronas ocultas y r neuronas de salida.

Como puede verse en la ilustración anterior por cada neurona de la RNA se codifica una conexión a cada una de las neuronas que lo admiten, es decir neuronas ocultas y de salida. Además por cada neurona que no sea de entrada se codifica el peso de la conexión de tendencia y el tipo de función sigmoide.

La codificación es claramente posicional y cada una de las posibles conexiones existentes en una red neuronal recursiva completamente conectada, tiene su lugar reservado en el cromosoma. Aún así, las conexiones pueden no expresarse en el fenotipo si el bit de existencia de la misma posee el valor 0 (cero).

La dimensión del cromosoma depende, además de la cantidad de neuronas de la red, de la precisión –medida en cantidad de bits– que se utilice para codificar los pesos. El valor de este parámetro puede cambiarse al momento de comenzar una prueba de neuroevolución en el

software que acompaña este trabajo. Los campos¹ correspondientes al tipo de sigmoide (TS) se codifican en 2 bits, mientras que para codificar la existencia de las conexiones (Ex) sólo hace falta 1 bit.

En caso de evolucionar redes neuronales sin término de tendencia y sin la característica multisigmoide se prescinde de los campos correspondientes, así el cromosoma se forma por la concatenación de la codificación genética de cada conexión –bit de existencia y peso de la misma–. La información de las neuronas *fuelle* y *destino* se determina de manera implícita por la posición que ocupa en el cromosoma.

Este tipo de codificación es perfectamente viable, tan viable que incluso ha sido la primera forma implementada. No obstante he preferido modificarla. La razón de ello se basa principalmente en la intención, ya explicitada, que la codificación genética utilizada en las seis estrategias neuroevolutivas sea lo más parecida posible; pero también en algunas otras ventajas menores que se mencionarán luego.

Codificación genética definitiva

Basándome en la codificación genética que se acaba de presentar y en la que utiliza el método SANE en su población de neuronas [Moriarty96] [Moriarty97_1] [Moriarty97_2], he arribado a un resultado que puede considerarse una extensión de la codificación presentada por Moriarty en dos sentidos: SANE no utiliza conexión de tendencia ni distintos tipos de función de transferencia y está definido sólo para redes del tipo feedforward.

Más adelante en este capítulo también expondré de qué manera se ha extendido SANE a redes neuronales recurrentes, multisigmoide y con término de tendencia. Por ahora regresemos al tema que nos concierne. Para la nueva codificación adoptada se resolvió lo siguiente:

- Codificar sólo neuronas ocultas y de salida en las RNA recurrentes y sólo neuronas ocultas en las RNA feedforward. Cada neurona se codifica como una secuencia de n conexiones y opcionalmente los campos TS y W_0 . Donde n es un parámetro agregado al algoritmo que debe ser establecido antes de comenzar la evolución.
- Codificar cada conexión como un par (id, w) donde id identifica la neurona con quien conectarse y w representa el peso de la conexión.

Así el ejemplo de codificación de una Red neuronal recurrente, multisigmoide, con conexión de tendencia, con p neuronas de entrada q neuronas ocultas y r neuronas de salida puede esquematizarse de la siguiente manera:

¹ El lector podría preguntarse por qué llamo *campo* y no *genes* a las subdivisiones funcionales codificadas en el cromosoma. Siguiendo a Goldberg [Goldberg89], gen es una posición binaria con dos alelos posibles (0 ó 1). En realidad existen varias definiciones de gen. Cada definición hace referencia a distintas ideas. Puede considerarse el término gen como unidad básica de herencia, como unidad básica de mutación o como unidad funcional [GALA]. En el presente trabajo se considera al gen como unidad básica de mutación, de esta manera, en un cromosoma binario cada bit representa un gen y cada división funcional es un conjunto de genes posiblemente epistáticos –recordemos que epistasis es una relación entre genes no alélicos que se da cuando la expresión de un gen está influenciada por otros genes–. Más adelante cuando veamos SANE tendremos la oportunidad de trabajar con cromosomas codificados como un vector de números reales, en la población de Blueprints. Allí sí cada número real representará un alelo de su gen correspondiente.

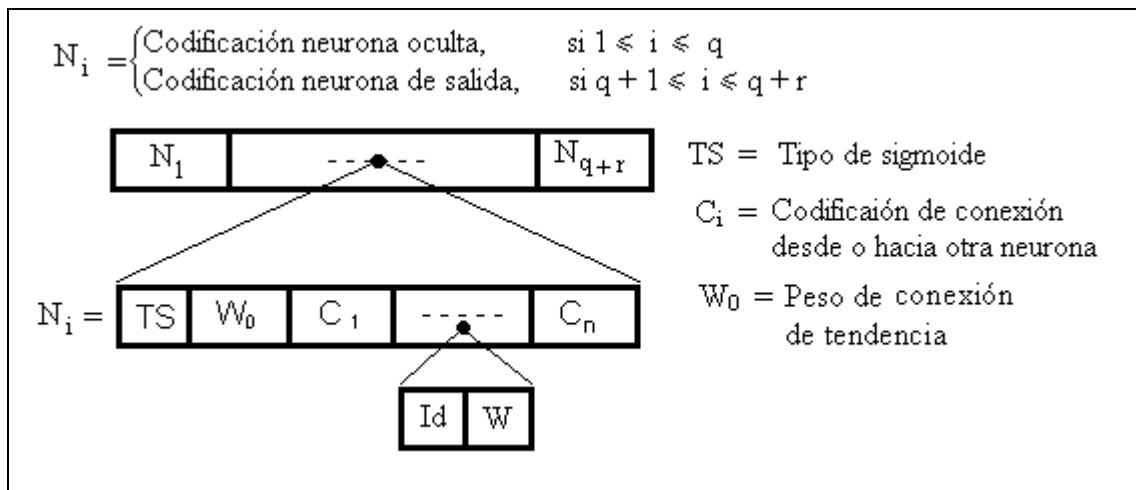


Ilustración 98 Codificación de RNA recurrente, multisigmoide, con conexión de tendencia, con p neuronas de entrada q neuronas ocultas y r neuronas de salida.

La simplificación con el esquema anterior es evidente. La clave está en cómo interpretar el campo Id para conocer la identidad de la otra neurona y el sentido de la conexión.

Se ha fijado el tamaño del campo Id en 8 bits. Así su valor decimal pertenece al intervalo $[0..255]$. Sea n el número total de neuronas de la RNA, n_E , n_O y n_S el número de neuronas de entrada, ocultas y de salida respectivamente, es decir $n = n_E + n_O + n_S$. Si se numeran todas las neuronas comenzando por cero, colocando primero las de entrada, luego las ocultas y por último las neuronas de salida, el campo Id se interpreta de la siguiente manera:

Sea D el valor decimal del campo Id :

Si $D > 127 \rightarrow$ la conexión es hacia la neurona numerada $(D \bmod (n_O + n_S)) + n_E$

Si $D \leq 127 \rightarrow$ la conexión es desde la neurona de entrada $(D \bmod n_E)$

Obsérvese que más sencillo era identificar a la neurona por medio de $(D \bmod n)$ y si ésta resultaba de entrada sería una conexión *desde* ella, por el contrario si fuese oculta o de salida se trataría de una conexión *hacia* la misma. Si bien es perfectamente factible, esta interpretación podría dar lugar a redes con escasas conexiones desde las neuronas de entrada, puesto que en cierta forma se encuentran en desventaja al no codificarse en el cromosoma. Incluso podrían generarse redes con neuronas de entradas totalmente desconectadas, situación indeseable ya que la capa de entrada generalmente queda fijada por la naturaleza del problema a resolver y no se espera que el algoritmo optimice esta cantidad.

Al igual que en la codificación anterior, se podrán generar redes con menor cantidad de neuronas del máximo especificado, como consecuencia de la presencia de circuitos neuronales que no influyan en la salida. En tal caso, las neuronas comprometidas en ellos no se consideran pertenecientes a la red.

El tamaño del cromosoma ahora depende de otro parámetro más, que el usuario debe fijar antes de comenzar la evolución, me refiero a la cantidad de conexiones a codificar por cada neurona –oculta o de salida–. Este parámetro nos brinda una nueva forma de controlar la longitud del cromosoma y la complejidad de las redes neuronales resultantes.

En caso de evolucionar redes neuronales sin término de tendencia o con función de transferencia sigmoide fija, simplemente se obvian los campos TS y W_0 .

La codificación de las redes feedforward es idéntica salvo que se codifican sólo las neuronas ocultas de la red y la interpretación del campo Id es la siguiente:

Se numeran por separado las neuronas de entrada y de salida comenzando por cero. Sea D el valor decimal del campo Id :

Si $D > 127 \rightarrow$ la conexión es hacia la neurona de salida ($D \bmod n_S$)

Si $D \leq 127 \rightarrow$ la conexión es desde la neurona de entrada ($D \bmod n_E$)

Ventajas de esta codificación

Además de la ya mencionada homogeneidad con las demás estrategias neuroevolutivas implementadas en el presente trabajo, se pueden mencionar otras cualidades:

- Esta codificación permite mayor control sobre la longitud del cromosoma debido al nuevo parámetro configurable: cantidad de conexiones por neurona. Esto brinda una mayor gama de posibilidades al momento de decidir qué pruebas evolutivas realizar.
- Elegir un valor bajo para este parámetro producirá cromosomas más pequeños –lo que mejora la performance del Algoritmo Genético, y Redes Neuronales más simples, que si logran resolver la tarea lo harán consumiendo menos recursos de cómputo.
- Se eliminan las zonas desaprovechadas dentro del cromosoma, como aquellas conexiones que ocupando lugar no se manifiestan por tener el bit de existencia igual a cero.

Finalmente una última acotación sobre la codificación de los pesos de conexión en el cromosoma. Como ya se mencionara, el espacio asignado a tal efecto se define a través de un parámetro antes de comenzar un ensayo neuroevolutivo. Este valor debe ser al menos de 2, debido a que el bit más significativo codifica el signo y el resto la magnitud del peso que se interpreta como el valor decimal de los $p-1$ bits dividido por $2^{(p-1)}$ –siendo p la precisión del campo–. De esta manera se obtiene un valor real perteneciente al intervalo $[-1..1]$.

Esta operación de escala permite que distintas pruebas evolutivas con desiguales cantidades de bits asignadas a los pesos, arrojen fenotipos –Redes Neuronales– con similares magnitudes en los pesos de conexión.

Neuroevolución Simple con Algoritmo Genético Elitista

Recordemos que el elitismo es una técnica utilizada en los Algoritmos Genéticos para mejorar la convergencia. Se utiliza en problemas donde la evolución se comporta de manera errática con excesivo grado de diversidad. Sin llegar a tal extremo, la estrategia neuroevolutiva antes presentada se mostró en varias oportunidades algo lenta en la producción de individuos que mejoren el fitness alcanzado. Así fue que se decidió agregar una pequeña cuota de elitismo asegurando que el mejor individuo de cada generación pase directamente a la siguiente.

Por lo tanto, esta estrategia constituye una variante de la anterior, siendo igual en todos sus aspectos salvo en el hecho que dos copias del mejor individuo de una generación –la elite– se incorporan directamente a la siguiente sin pasar por el proceso reproductivo, sólo aplicando el operador de mutación, con baja probabilidad.

Veamos esquemáticamente el Algoritmo Genético implementado.

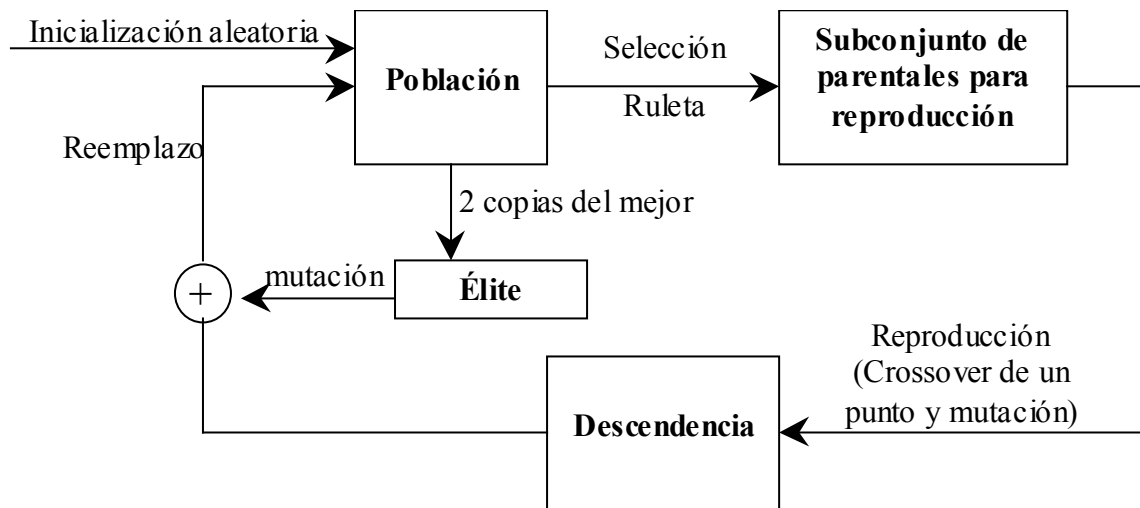


Ilustración 99 Algoritmo Genético Elitista implementado

Obviamente para mantener el tamaño de la población constante se generan sólo $n-2$ descendientes, siendo n el tamaño de la población.

La codificación genética utilizada es la misma que en el caso de la estrategia denominada Neuroevolución Simple.

Pasemos ahora a la explicación de SANE, un método neuroevolutivo importante en el contexto del presente trabajo pues el resto de las estrategias implementadas –salvo GNE– pueden considerarse variantes o extensiones de él. Dejamos de lado en esta apreciación al método GNE pues si bien se vale de una variante de SANE para sus evoluciones, ello no es lo más relevante y podría no hacerlo. GNE está centrado en otro concepto: conseguir más de una red neuronal como respuesta a un problema dado.

SANE –Symbiotic Adaptive NEuroevolution–

Moriarty y Miikkulainen presentaron este método de neuroevolución demostrando las ventajas de una coevolución cooperativa en la búsqueda de soluciones a problemas de control [Moriarty96][Moriarty97_1][Moriarty97_2].

Básicamente la idea es evolucionar una población de neuronas que cooperen entre sí para formar redes neuronales que sean capaces de resolver el problema en cuestión. De esta manera, las neuronas de la población evolucionarán para cumplir diferentes –aunque superpuestos– roles que al integrarse en una red neuronal resuelven satisfactoriamente un problema dado.

Según los autores SANE se mostró más eficiente y adaptativo manteniendo un mayor nivel de diversidad que las evoluciones basadas en redes neuronales completas. La importancia de la diversidad genética en los algoritmos evolucionarios ya ha sido tratada en este trabajo –ver “Diversidad Genética y Convergencia” pág. 64–.

SANE coevolucionar dos poblaciones, a saber, de neuronas y de redes neuronales. Pero ¿qué es exactamente la coevolución? Seguramente el lector tenga la respuesta, o al menos una aproximación a la misma. Los autores de SANE no dan mayores precisiones del concepto y por ello citaré un párrafo de un artículo de Claudio Gutiérrez [Gutierrez00_1] donde se explica el concepto con gran claridad.

“...Quiero en cambio ahondar aquí en el concepto de coevolución, un estilo particular de evolución muy extendido en la naturaleza. Un ejemplo sencillo y hermoso: la integración ecológica entre flores y abejas. La vista multicolor de las abejas se desarrolló

simultáneamente con los colores de las flores cuyos néctares las alimentan. ¿Las flores obtuvieron sus colores para que las abejas las polinizaran o las abejas obtuvieron su visión de colores para poder encontrar a las flores? Las dos cosas y ninguna de ellas. Ninguna de ellas porque "para" no existe en la naturaleza. La evolución ocurre mecánicamente, no por la intervención de propósitos. Y las dos cosas, porque la evolución seleccionó simultáneamente a las flores más visitadas por las abejas, y a las abejas que pudieron distinguir a las flores. Esto es lo que llamamos coevolución.”

Claudio Gutiérrez, La coevolución del cerebro y el lenguaje dentro de una perspectiva humanista.

Dentro del paradigma de la Computación Evolutiva los algoritmos cooperativos coevolucionarios ofrecen una alternativa atractiva a los algoritmos evolucionarios clásicos en problemas dinámicos y de difícil solución. En este modelo cada individuo representa sólo una solución parcial al problema. La solución completa debe formarse agrupando varios –o todos– los individuos que cooperativamente pueden llevar a cabo la tarea requerida. Así el objetivo de los individuos es optimizar diferentes piezas de la solución cooperando con otras soluciones parciales que optimizan otras piezas. La hipótesis –comprobada por los autores en varios casos de experimentación– es que varias búsquedas paralelas de diferentes piezas de solución es más eficiente que una simple búsqueda de una solución entera.

Además manteniendo varias piezas en una población simple, ésta no converge a un único individuo, es decir se mantiene la tan deseada diversidad y el Algoritmo Genético siempre obtendrá el combustible necesario para continuar su búsqueda a través del operador de crossover. Según los autores esto se debe a la presencia de una fuerte presión evolucionaria para hallar varios tipos diferentes de neuronas, ya que un solo tipo no puede realizar bien el trabajo. La importancia de mantener la diversidad genética en la población que se evolucionan no es menor en el contexto de la Computación Evolutiva, de hecho es un tema difícil aún abierto a la investigación. En otro capítulo ya se ha visto que existe una línea importante de investigación en este tema y se han presentado varias estrategias desarrolladas a favor de ésta. Pues bien, SANE se perfila también como una buena alternativa para conseguirla.

Otra ventaja importante radica en que al reconocer a las neuronas como componente funcionales dentro de las redes neuronales esto favorece una búsqueda más precisa y mejor evaluación de los “building blocks” genéticos. –ver “Paralelismo implícito y la Hipótesis de construcción de bloques” en la pág. 62–. Así será más difícil destruir buenos bloques constructivos en la población de blueprints –población de redes neuronales armadas con unidades seleccionadas de la población de neuronas que veremos un poco más adelante– y en consecuencia se incrementa la performance del Algoritmo Genético.

Descripción del método

El método de neuroevolución simbiótica adaptativa propuesto por Moriarty y Miikkulainen se limita a un tipo de arquitectura específica. Si bien existe evolución de arquitectura pues parte de la topología –esquema de conexión– está sometida al proceso evolutivo, con él se consiguen sólo RNA del tipo feedforward con una única capa oculta. Tampoco se evolucionan ni pesos de tendencia, pues éste no existe en el modelo presentado, ni funciones de transferencia.

En el presente trabajo se implementa fielmente el modelo original pero también se realiza una generalización del mismo que permite evolucionar redes neuronales de arquitectura arbitraria –recurrentes–. Además el software desarrollado permite realizar pruebas evolutivas con o sin pesos de tendencia y con o sin la característica multisigmoide ya comentada, es decir que también se ha implementado una forma de evolución de función de transferencia. Más

adelante se explicará esto con detalle, pero por el momento, retomemos la explicación del método original.

El método trabaja coevolucionando dos poblaciones, una de neuronas a la que se hará referencia como población de unidades y la otra de redes neuronales que recibe el nombre de población de blueprints.

Población de unidades

Cada individuo en esta población representa una neurona en la capa oculta de una red de dos capas¹. La red neuronal completa se crea por la combinación de varias neuronas elegidas de esta población. En el cromosoma cada neurona se codifica como un conjunto de conexiones, éstas pueden ser desde las neuronas de entrada o hacia las neuronas de la capa de salida. Así simplemente por la elección de un conjunto de neuronas ocultas queda determinada la topología y los pesos de la red.

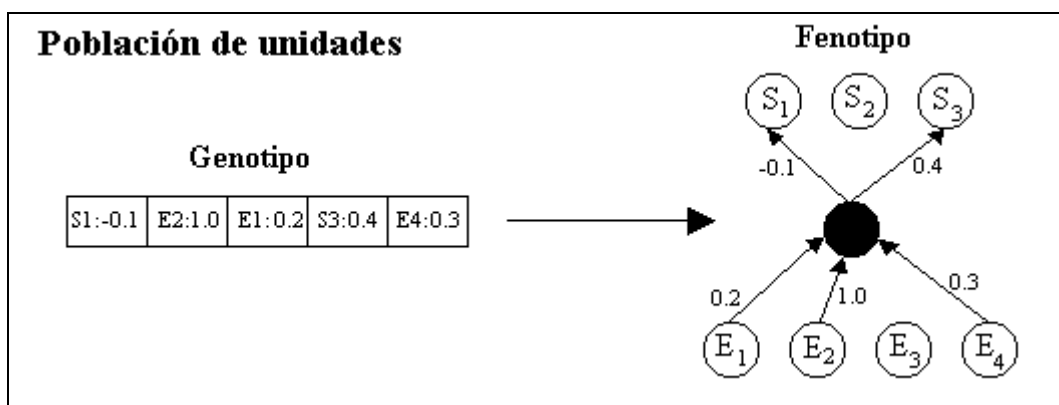


Ilustración 100 Un individuo en la población de unidades representa una neurona de la capa oculta codificada como una serie de pesos de conexión desde la entrada o hacia la salida. Por ejemplo, el primer campo especifica una conexión hacia la neurona de salida 1 con peso -0.1 y el segundo una conexión desde la neurona 2 de entrada con peso 1.0

Ya se ha mencionado que la presión evolucionaria llevará a la evolución de varios tipos diferentes de neuronas –especialización– que deberán interactuar de manera simbiótica cooperando para formar una red neuronal eficiente. Por ello este tipo de evolución que ocurre en SANE se caracteriza como evolución simbiótica². Así la evolución simbiótica es aquella donde individuos cooperan explícitamente con otros y dependen de ellos para subsistir. Esto pone en evidencia una clara diferencia entre SANE y los métodos de neuroevolución tradicionales donde los individuos compiten más que cooperan entre sí para sobrevivir.

Como la solución no depende de un individuo –neurona– sino de varios que intervienen en la formación de la RNA surge el problema de la asignación del fitness a cada uno de ellos. Al evaluar el desempeño de una solución se obtiene un valor de fitness para la RNA que

¹ Recordamos que la postura asumida en este trabajo es no contar a las neuronas de entrada como una capa de la red ya que éstas no hacen ni procesamiento ni intervienen en ningún proceso de aprendizaje, no obstante no hay unicidad en el criterio adoptado por la literatura especializada.

² Recordemos que en biología una relación simbiótica –a diferencia de una parasitaria en donde sólo una parte se beneficia– dos individuos o especies de individuos se relacionan obteniendo ambos beneficios recíprocos de la relación. Ya se ha mencionado en el apartado “Evolución biológica” –pág. 2– de un capítulo anterior un ejemplo de relación simbiótica que se supone tuvo lugar hace miles de millones de años cuando los organismos vivos más complejos eran apenas unicelulares. En aquel entonces, una bacteria habría invadido las células conformando una de las primeras simbiosis que se hayan producido. Esta bacteria primitiva se ha convertido en las mitocondrias actuales que se hayan en el citoplasma de las células recibiendo protección a cambio de suministro de energía a través de la respiración celular que llevan a cabo.

representa dicha solución. Alguna estrategia de reparto debe ser implementada para poder asignar un valor de aptitud a las neuronas intervinientes en la RNA para proseguir con la evolución. Esta estrategia será discutida luego, ahora veamos en la siguiente ilustración la diferencia fundamental entre un método neuroevolutivo estándar y SANE.

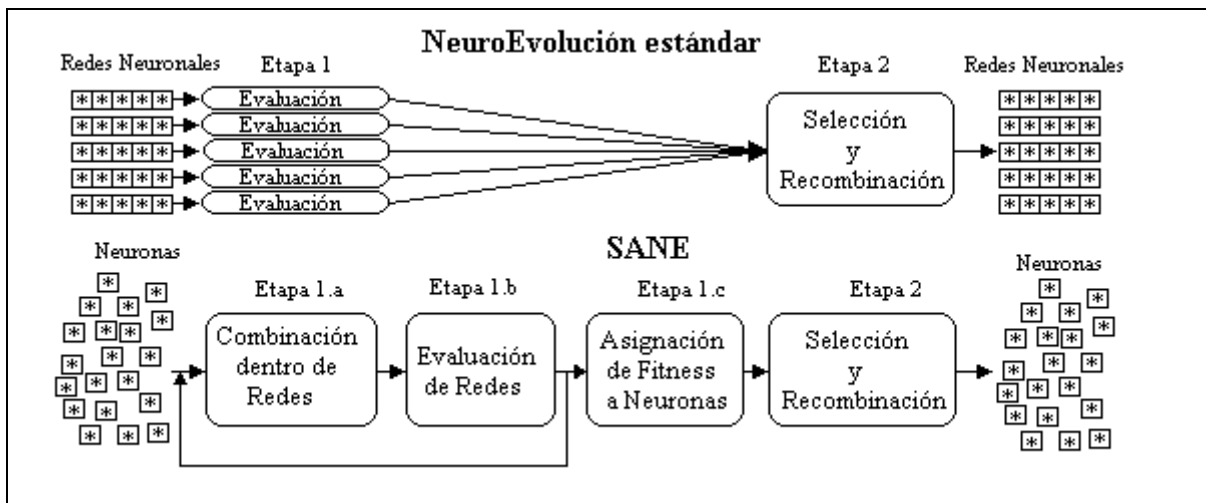


Ilustración 101 Mientras que en la neuroevolución estándar se mantiene una población de redes neuronales completas y se evalúan cada una independientemente, en SANE se mantiene una población de neuronas las que sólo pueden ser evaluadas en relación con otras neuronas. En SANE la etapa de evaluación se divide en tres subetapas. Las neuronas son continuamente combinadas con otras y la red neuronal resultante es evaluada en la realización de la tarea. Cada neurona luego recibe un valor de fitness basado en la performance de las redes en las cuales ha participado.

Población de blueprints

La evolución de sólo neuronas no alcanza para generar las redes necesarias en la resolución de tareas difíciles. El conocimiento de las buenas combinaciones de neuronas debe ser mantenido y explotado de alguna forma. La población de blueprints se encarga de ello, manteniendo un registro de las combinaciones más efectivas de neuronas encontradas en la población de unidades –neuronas– corriente. Por lo tanto cada cromosoma de esta población se decodifica como una determinada combinación de neuronas de la otra población y esta combinación está sometida a evolución. Así el método coevoluciona dos poblaciones simultáneamente, y al igual que en el ejemplo de las flores y las abejas, nunca intercambian material genético pero la evolución de una está supeditada a la de la otra.

Mientras que la población de unidades busca optimizar redes neuronales parciales –de una sola neurona– la población de blueprints busca la mejor combinación de ellas para optimizar una RNA completa.

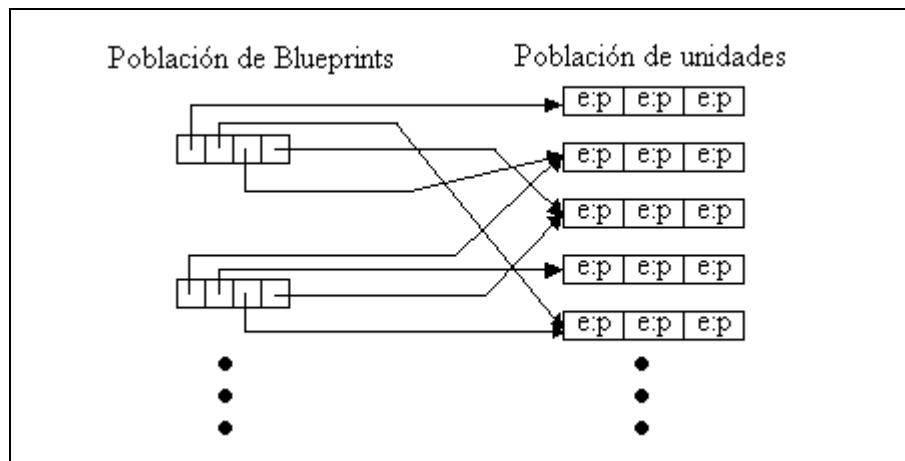


Ilustración 102 Relación entre la población de Blueprints y la población de unidades. Cada miembro de la población de unidades codifica una secuencia de conexiones (etiqueta:peso) que definen completamente una neurona de la capa oculta. Cada miembro de la población de Blueprints consiste en una serie de punteros a la otra población para determinar qué neuronas usar en la construcción de la red neuronal.

Antes de pasar a describir los detalles de implementación se debe hacer una aclaración. SANE se ha convertido en un método de neuroevolución sobre el cuál se han propuesto muchas variantes que mejoran en algún aspecto el modelo original. Por ende es frecuentemente citado pero rara vez se hace explícita la versión que se está citando.

Básicamente existen dos versiones de SANE, la primera fue presentada en 1996 [Moriarty96] introduciendo el modelo de evolución cooperativa. En esta versión no existe población de blueprints. Las redes neuronales se construyen tomando al azar la cantidad de neuronas necesarias de la población de unidades. Deben crearse suficientes redes como para asegurar la participación de cada neurona en lo posible más de una vez en alguna RNA. Aquí el fitness de las neuronas se calcula simplemente como el promedio de los fitness obtenidos por todas las redes en las cuales la neurona haya participado.

Esta primera versión mostró algunas limitaciones como consecuencia de no llevar registro de las buenas combinaciones de neuronas halladas. Tales limitaciones fueron superadas inmediatamente por la segunda versión del método presentada en 1997 [Moriarty97_1] [Moriarty97_2] que agregó la importante población de blueprints y con ella el concepto de coevolución.

Esta segunda versión es la que se describe e implementa en este trabajo, y siguiendo a sus autores se hará referencia a ella con el término SANE. Cuando se necesite nombrar a la primera versión de SANE se utilizará el nombre “*Neuron SANE*” pues así ha sido rebautizada por Moriarty y Miikkulainen en [Moriarty97_1] [Moriarty97_2]

Detalles de implementación

Ya se ha mencionado que SANE coevoluciona dos poblaciones: de unidades –neuronas– y de blueprints –redes neuronales–. En el modelo presentado por Moriarty y Miikkulainen, cada individuo de la población de unidades representa una neurona de la capa oculta de una red de dos capas feedforward.

Estas neuronas se codifican genéticamente en un cromosoma como una secuencia de conexiones utilizando un alfabeto binario $\{0,1\}$. Estas conexiones, como se ve en la ilustración anterior, se definen por un par $e:p$ en donde e es la etiqueta que identifica la otra

neurona –de entrada o de salida– que interviene en la conexión, y p representa el peso de esta conexión.

Los autores han utilizado 8 bits en el cromosoma para codificar las etiquetas y 16 bits para codificar los pesos. En la implementación del método realizada aquí, al igual que ellos, se han adoptado los 8 bits para codificar la identificación de la otra neurona a conectar, en cambio, se ha dejado como un parámetro a definir por el usuario la cantidad de bits para codificar los pesos.

Recordemos que SANE trabaja con una clase restringida de redes con propagación hacia delante y de una sola capa oculta. Así todas las conexiones de cada neurona oculta o bien provienen de alguna neurona de entrada o bien se dirigen hacia una neurona de salida no permitiéndose conexiones recurrentes ni conexiones entre las neuronas ocultas.

Los valores de las etiquetas determinan donde se hará la conexión de la siguiente manera. Si el valor decimal de la etiqueta es D y suponiendo que la cantidad de neuronas de entrada es n_E y la cantidad de neuronas de salida es n_S se tiene que:

Si $D > 127 \rightarrow$ la conexión es hacia la neurona de salida ($D \bmod n_S$)

Si $D \leq 127 \rightarrow$ la conexión es desde la neurona de entrada ($D \bmod n_E$)

El valor del peso de la conexión queda especificado por el campo p interpretado como número real.

Como puede observarse esta codificación recuerda la utilizada en las estrategias neuroevolutivas anteriores, para el caso de las redes de tipo feedforward. Ello se debe a que precisamente los métodos “Neuroevolución simple” y Neuroevolución simple con Algoritmo Genético elitista” utilizan una tipo de codificación genética definida en base a la presentada por SANE para su población de unidades.

En la ilustración siguiente vemos un ejemplo para una RNA con 8 neuronas de entrada, 5 de salida y 3 neuronas ocultas. Debe recordarse que tanto la cantidad de neuronas de entrada como de salida quedan definidas por el problema a resolver, sin embargo, la cantidad de neuronas de la capa oculta es un parámetro crítico que no lo especifica el problema y por lo general su elección queda a cargo del diseñador del algoritmo, a menos que sea sometido a proceso evolutivo y hallado automáticamente. Una RNA de las características antes señaladas podrá construirse a partir de la decodificación de tres cromosomas de la población de unidades –apuntados por un cromosoma blueprint–.

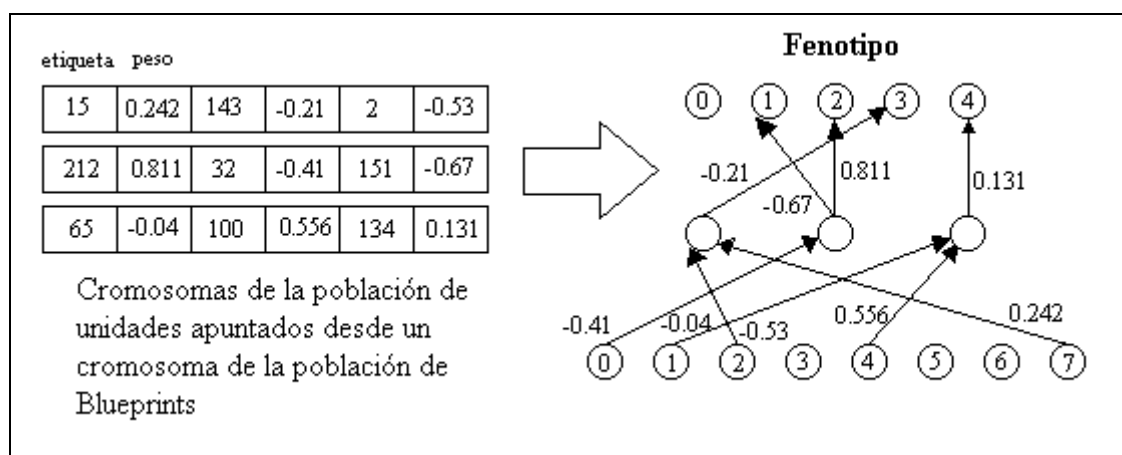


Ilustración 103 Decodificación de una red neuronal artificial a partir de tres cromosomas de la población de unidades apuntados por un individuo blueprint.

Una observación interesante puede hacerse al respecto. Una neurona de la capa oculta podría tener sólo pesos desde la entrada o sólo pesos hacia la salida, esto haría de ella una neurona inservible y equivaldría a la no existencia de la misma. Por lo tanto se puede decir que en SANE –al igual que en todas las otras estrategias neuroevolutivas implementadas en el presente trabajo– se debe determinar el número máximo de neuronas ocultas, que no es lo mismo que el número total de neuronas ocultas ya que la evolución puede llegar a conseguir RNA con menor cantidad. No obstante para ello –aunque sus autores no lo expliciten– debe haber presión selectiva, por ejemplo dándole un valor de fitness más alto a aquellas que realicen bien la tarea con menor cantidad de neuronas. También se da el mismo efecto si todos los pesos tienen el valor 0 (cero) –aunque este hecho es más difícil de conseguir–.

Un parámetro que hay que determinar y sobre el cual los autores no hacen referencia a ningún criterio para elegirlo es la cantidad de conexiones que tendrá cada neurona, creo que este también es un parámetro crítico que incide tanto en el fenotipo como en la longitud del cromosoma en la población de unidades.

Tampoco se establece ninguna restricción sobre la posibilidad de que las etiquetas de las conexiones se repitan en el cromosoma de la población de unidades, de esta manera es posible que una neurona de la capa oculta posea más de una conexión desde una neurona de entrada –o hacia una neurona de salida–.

Esto en apariencia es una novedad en lo que se ha escrito hasta aquí sobre redes neuronales artificiales, al menos no he encontrado bibliografía que hable de la posibilidad de que una neurona excite –o inhiba– más de una vez a la misma neurona, no obstante debe observarse que es equivalente a tener una sola conexión cuyo peso es la suma de aquellas conexiones que proviene de –o se dirigen hacia– la otra neurona. Este efecto también aparece en la codificación genética adoptada para las estrategias neuroevolutivas presentadas anteriormente que, como ya se explicó, constituye una adaptación de la utilizada aquí.

Los cromosomas de la población de blueprints están constituidos por un arreglo de longitud L –donde L es la cantidad de neuronas ocultas que tendrá la red resultante– de punteros a cromosomas de neuronas en la población de unidades –ver Ilustración 102–. El alfabeto utilizado para la codificación genética ya no es el binario sino por el contrario se utiliza una representación de números reales.

Como ya se ha explicado en el capítulo “Computación Evolutiva” la utilización de un alfabeto distinto al binario en los Algoritmos Genéticos, frecuentemente debe ir acompañada por la redefinición de los operadores de mutación y crossover. En el caso de SANE el crossover es el mismo que en el algoritmo genético canónico pero la mutación es redefinida. Es más, los autores han utilizado no uno sino dos mecanismos de mutación distintos que veremos más adelante.

El algoritmo de neuroevolución SANE

Inicialmente las dos poblaciones se construyen aleatoriamente. Se pueden resumir los pasos del algoritmo de la siguiente manera:

1. Borrar el valor de fitness de cada neurona y blueprint.
2. Seleccionar L neuronas de la población de unidades usando un blueprint.
3. Crear una RNA con las neuronas seleccionadas.
4. Evaluar la performance de la red en la resolución de la tarea dada.
5. Asignar al blueprint el valor de fitness de acuerdo con la performance de la RNA evaluada.

6. Repetir los pasos 2 a 5 para cada individuo de la población de blueprint.
7. Asignar a cada neurona un fitness de acuerdo a la performance de las RNA en las que ha participado.
8. Obtener la siguiente generación en la población de unidades a través de la selección, reproducción y reemplazo.
9. Obtener la siguiente generación en la población de blueprints a través de la selección, reproducción y reemplazo.
10. Si no se cumple alguna condición de terminación volver al punto 1.

Según los autores del método éste se enmarca dentro del paradigma de los Algoritmos Genéticos por utilizar crossover además de la mutación como operadores genéticos aplicados a strings binarios –al menos en la población de unidades–.

Asignación de fitness

Mientras que los individuos de la población de blueprints reciben directamente el valor de fitness de acuerdo a la performance en la realización de una tarea específica, de la RNA que codifican, no ocurre lo mismo con la población de neuronas donde cada una de ellas puede participar en más de una red neuronal durante la etapa de evaluación. Además estas redes pueden tener muy distinta performance dependiendo de las otras neuronas involucradas en la misma.

Es importante mantener las neuronas que hayan tenido un buen desempeño en alguna RNA aunque haya participado en otras de muy mala performance. Por lo tanto los autores han propuesto asignar el fitness a las neuronas calculado como la suma de los fitness de las 5 mejores RNA en las que ha intervenido. No obstante, en algunos ejemplos han sugerido que puede ser mejor utilizar sólo las 3 mejores. En la implementación del método realizada aquí, se ha experimentado no solo con estos valores sino con otros más. No detectando mejora evidente de alguno respecto de otro he fijado este valor en 5.

Obsérvese que de esta manera se premia a las neuronas que más participación posean en las redes neuronales puesto que una que sólo participe de una red no sumará un buen fitness aunque la performance de ésta sea buena. Según los autores en SANE se consigue que la distribución de las neuronas participantes en redes neuronales se dirija hacia las mejores, dejando muchas neuronas sin evaluar en cada generación. A su vez estas neuronas no evaluadas recibirán un fitness de 0 y serán seguramente reemplazadas en la próxima generación. Esto marca una gran diferencia con la versión primitiva Neuron SANE donde todas las neuronas tenían la misma oportunidad de participar.

Los autores no dicen nada respecto al tamaño de las poblaciones, pero en ejemplos presentados han utilizado una relación $n_U = n_O * n_B$ donde n_U es la cantidad de individuos en la población de unidades, n_O es la cantidad de neuronas ocultas de las redes a construir, y n_B la cantidad de individuos en la población de blueprints. De esta manera existen tantas neuronas en la población de unidades como neuronas se necesitan para construir todas las redes neuronales que definen la población de blueprints. En el software que acompaña este trabajo se deja completa libertad al usuario para definir el tamaño de las poblaciones a su gusto. No obstante se calcula la relación mencionada anteriormente para sugerir, en una leyenda, el tamaño de la población de unidades por si acaso el usuario desea considerarla.

El fitness no se calcula con los valores recogidos de todas las redes en que la unidad ha intervenido porque interesa mantener aquellas neuronas que realizan un buen trabajo en las RNA buenas aunque participen de otras RNA con mal desempeño. Esto se debe a que como

decíamos antes la población de unidades tenderá a evolucionar distintas clases de éstas que se especializarán en ciertas funciones precisas. Estas neuronas especializadas serán cruciales en la performance de las RNA exitosas y es necesario mantenerlas. Por ejemplo si estamos evolucionando redes para controlar un brazo robótico, una neurona que se especialice en pequeños movimientos cuando el dispositivo esté próximo al objetivo será efectiva en la red que sea capaz de posicionar el brazo cerca del objetivo, pero resultará ineficiente en aquellas que no lleguen a tal situación. Estas neuronas son muy importantes en la población y no deben ser penalizadas por su participación en RNA de pobre desempeño en las que no pueden ayudar.

Selección, reproducción y reemplazo

El método SANE utiliza una estrategia de selección y reemplazo fuertemente elitista –ver “Elitismo” en pág. 72– en ambas poblaciones. En pocas palabras, la mejor mitad de la población pasa intacta a la próxima generación, y el mejor cuarto es seleccionado para reproducirse y con su descendencia completar la otra mitad. Veámoslo más detalladamente.

Una vez completada la evaluación y asignados los fitness de ambas poblaciones, éstas son ranqueadas en base al valor de aptitud, es decir se obtiene una tabla de posiciones de los individuos en orden decreciente de fitness. Así los mejores individuos se encuentran arriba en la tabla y los peores en las últimas posiciones. El primer 50% de la tabla –mejor mitad ranqueada– pasa a la próxima generación. Luego para cada individuo del primer 25% se selecciona al azar otro individuo de este mismo súper selecto 25%. Estos dos ejemplares se reproducen obteniendo dos hijos que pasan a la siguiente población reemplazando al peor 50% de la población actual.

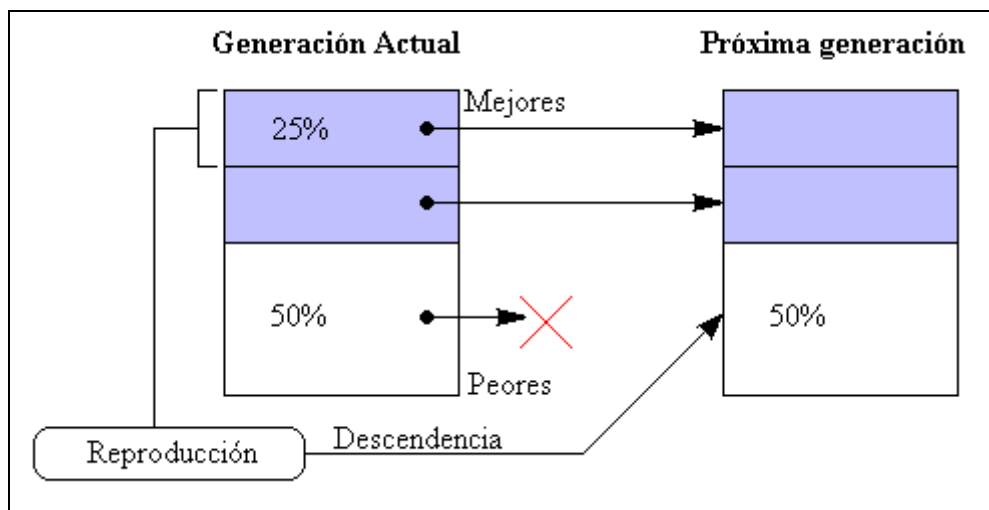


Ilustración 104 Selección y reemplazo para la población de unidades y de blueprints en SANE.

Este fuerte elitismo y selección agresiva no son normalmente usados en aplicaciones evolucionarias porque lleva a una rápida convergencia de la población, que como se advirtiera varias veces en este trabajo no es deseable pues se pierde diversidad y con ella eficiencia en la evolución. Sin embargo en SANE funciona muy bien pues a la convergencia se contrapone una fuerte presión evolucionaria por conseguir distintos tipos de neuronas especializadas en la población de unidades, y un doble mecanismo de mutación con alta probabilidad de ocurrencia para el caso de la población de blueprints.

Si bien el proceso de selección y reemplazo es el mismo en ambas poblaciones, la reproducción se realiza de manera diferente en cada una de ellas.

Reproducción en la población de unidades

La reproducción en la población de unidades se realiza por medio del operador de crossover de un punto. Esta operación, como ya sabemos, produce dos nuevos cromosomas. No obstante sólo se utiliza uno de ellos elegido aleatoriamente como primer descendiente. El segundo se consigue seleccionando al azar a alguno de los padres. El carácter marcadamente elitista del método vuelve a manifestarse aquí. Copiando uno de los padres como el segundo descendiente reduce el efecto de una mutación adversa en esta población de unidades, pero este efecto lo se explicará mejor en el próximo apartado.

Una vez completada la fase de recombinación con la que se obtiene el 50% de la próxima generación, se aplica el operador de mutación con probabilidad 0.001 por bit a la población entera, es decir tanto a la elite como a los descendientes obtenidos.

Reproducción en la población de blueprints

En la población de blueprints, debido a que los cromosomas están constituidos por una secuencia de punteros –arreglos de números reales– en lugar de bits, el crossover ocurre sólo entre punteros. Así toda neurona interviniente en alguno de los dos padres también pertenecerá a alguno de los dos hijos.

Es necesario redefinir el operador de mutación para que al aplicarse sobre estos cromosomas sigamos teniendo un genotipo válido, permitiendo sólo punteros a neuronas existentes en la población de unidades. Para evitar el problema de convergencia en la población de blueprints se han definido dos tipos de mutación distintas que se aplican con alta probabilidad sólo a la descendencia, no a la elite del 50% superior que pasa directamente a la próxima generación sin ningún tipo de alteración.

Mutación 1: Por cada puntero en el cromosoma se aplica con alta probabilidad de 0.01 una mutación consistente en el reemplazo por otro puntero válido a una neurona en la población de unidades seleccionado al azar. Este operador hace posible que participen de las redes neuronales unidades mal ranqueadas en su población. Así una neurona que no participa en ninguna red puede conseguir ser apuntada e intervenir en la próxima generación.

Mutación 2: El segundo operador de mutación intenta tomar ventaja de las nuevas neuronas conseguidas por la reproducción en la población de unidades –por ello se obtiene primero la siguiente generación de neuronas y luego la de redes neuronales–. Recordemos que la reproducción de neuronas genera dos descendientes, uno idéntico –salvo por mutación– a alguno de sus padres y otro generado por crossover de un punto entre ellos. Así se supone que los descendientes serán similares y potencialmente superiores a sus padres. Este conocimiento puede ser aprovechado en la evolución de blueprints haciendo que con una muy alta probabilidad de 0.5 cada puntero a una neurona que se haya reproducido en la población de unidades sea reemplazado por un puntero a alguno de sus descendientes.

Recordemos que en el capítulo “Computación Evolutiva” se ha expuesto que el operador de mutación debe aplicarse con probabilidad muy baja –habitualmente cercana a 0.001– para no caer en una búsqueda aleatoria. La mutación de tipo 2 presentada anteriormente viola ampliamente este precepto. No obstante debe notarse que lejos de introducir un componente totalmente aleatorio representa un mecanismo de mutación extremadamente limitado y por ende no le caben las generalidades de la ley respecto de la probabilidad de aplicación.

Este mecanismo de mutación limitada posee dos ventajas. La primera ya se ha mencionado y radica en el hecho que permite explotar las nuevas neuronas generadas por la evolución de la población de unidades. La segunda ventaja se relaciona con el tipo de reproducción que se

lleva a cabo en la población de unidades donde uno de los descendientes es la copia directa de uno de los padres.

Muchos punteros serán reasignados a una copia de la neurona que apuntaban anteriormente. Esto aparentemente no tiene ningún efecto, sin embargo hace a los blueprints más resistentes a la mutación adversa en la población de neuronas. Si estos punteros no fueran reasignados a una copia muchos blueprints apuntarían a exactamente la misma neurona y una mutación adversa en esta neurona afectaría a todo blueprint que la apunte.

Cuando los punteros son ocasionalmente reasignados a copias, esta mutación se limitará a unos pocos blueprint. Los autores del método relacionan este hecho con el teorema de los esquemas que presenté en otro capítulo –ver “Teorema fundamental –Teorema de esquemas” pág. 59– en el que se afirma que aquellos esquemas con alto fitness –en este caso las buenas neuronas– comienzan a dominar la población y la mutación de una copia del esquema no afecta otras copias en la población.

En la implementación de SANE que he realizado para este trabajo, se ha respetado fielmente cada uno de los detalles expuestos, para lograr una versión totalmente acorde a la especificación del método presentada en los artículos citados [*Moriarty97_1*] [*Moriarty97_2*].

No obstante, debe considerarse que SANE no soporta ni conexión de tendencia ni evolución de función de transferencia que se vienen manejando en este trabajo. Más restrictivo aún resulta el hecho que está definido sólo para redes feedforward de 2 capas y aquí he puesto cierta expectativa en las redes recurrentes para resolver el dominio de problemas que planteado y del cual ya ha hablado bastante al comienzo de este capítulo.

Es así que se evidencia la necesidad de extender el método de Moriarty –alcanza con extender la codificación– para incorporar conexión de tendencia, característica multisigmoide y especialmente, para soportar redes neuronales recurrentes.

Extensión al método SANE implementada

Como ya se ha mencionado, es posible extender fácilmente este método para soportar otras características simplemente haciendo cambios en la codificación genética.

Codificación con conexión de tendencia y tipo de sigmoide

Para dotar al método de estas dos características simplemente se modifica el esquema de codificación de la población de unidades anteponiendo a la secuencia de conexiones que definen una neurona los campos correspondientes al tipo de sigmoide (TS) y el peso de la conexión de tendencia (W_0).

Ningún cambio se hace necesario en la población de blueprints.

Así el nuevo esquema de codificación puede esquematizarse de la siguiente manera:

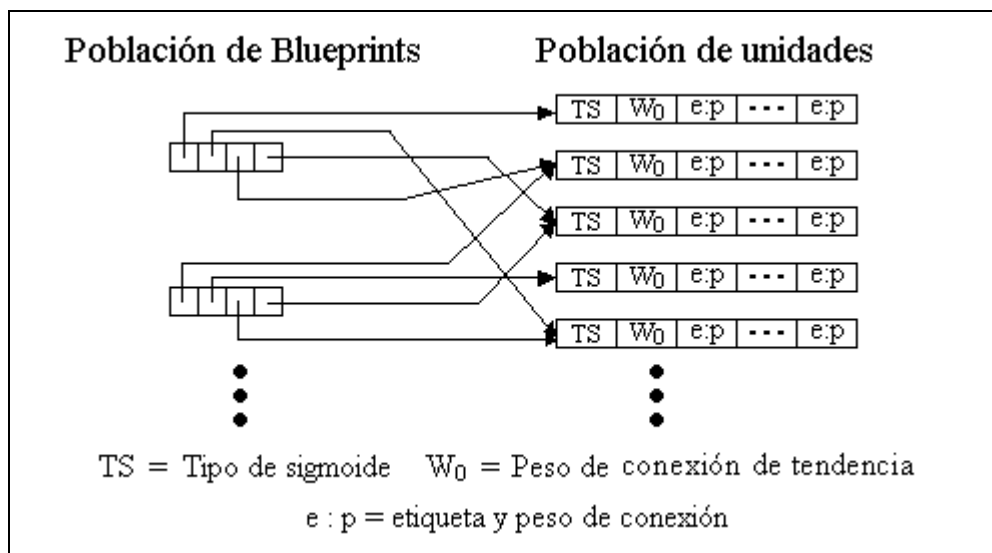


Ilustración 105 Esquema de codificación SANE extendida para soportar redes neuronales multisigmoide y conexión de tendencia

Como ejemplo de decodificación se extiende el presentado en la Ilustración 103 para una RNA feedforward con 8 neuronas de entrada, 5 de salida y 3 neuronas ocultas con término de tendencia y distintas funciones sigmoide de transferencia.

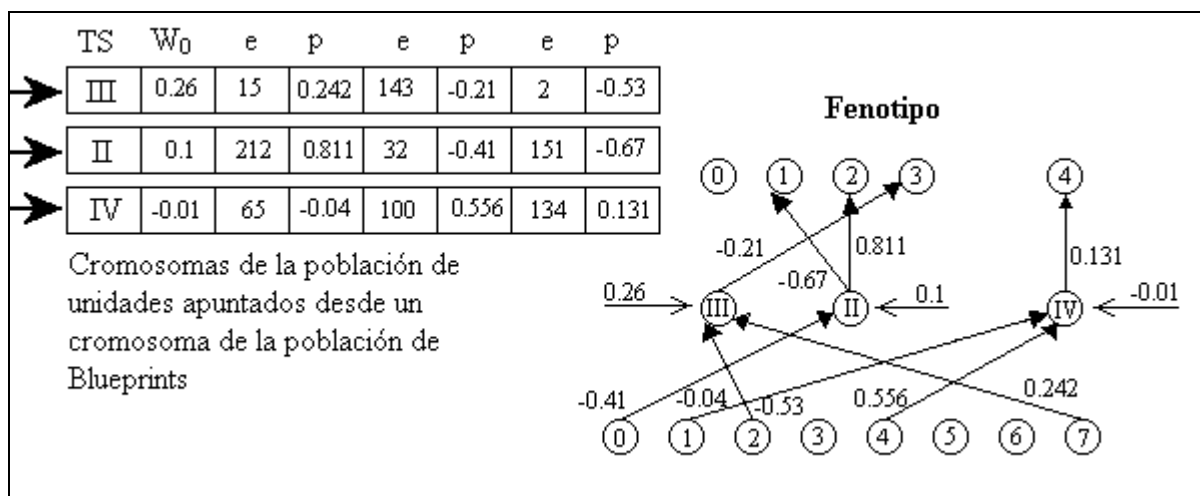


Ilustración 106 Ejemplo de decodificación de una RNA feedforward multisigmoide, con conexión de tendencia en la capa oculta, a partir de tres cromosomas de la población de unidades apuntados por un individuo blueprint.

El resto de los aspectos del método permanece inalterable. De esta sencilla forma se consiguen RNA feedforward con las características requeridas. No obstante, debe notarse que, como consecuencia de la codificación de tan sólo las neuronas de la capa oculta que realiza SANE, estos nuevos aspectos incorporados no afectan a la capa de salida, pues ninguna información se codifica respecto a estas neuronas. Para subsanar esta deficiencia se requiere un cambio más profundo en la codificación. Sin embargo, en el presente trabajo apuntamos a la evolución de RNA recurrentes y tal inconveniente, como veremos más adelante, no se presenta en ellas. Recordemos además que las RNA recurrentes que se han definido son un caso más general, un superconjunto que contiene a las de tipo feedforward.

Codificación para redes neuronales recurrentes

Para lograr redes neuronales recurrentes a través de la evolución se hace necesario cambiar la codificación en la población de blueprints, como así también la forma de interpretar las etiquetas de las conexiones en la población de unidades.

Lo que se expondrá en este momento no es completamente nuevo, puesto que guarda estrecha relación con la codificación para redes recurrentes que se utilizó en la estrategia “Neuroevolución simple” ya mencionada en este capítulo.

El cambio en la población de blueprints es muy sencillo. Debido a la posibilidad de que neuronas de salida retroalimenten a cualquier neurona de la red que no sea de entrada, se hace necesario incorporar esta información a nivel genético para someterla a evolución. He adoptado una forma muy simple de hacerlo. Se trata a las neuronas ocultas y a las de salida de igual manera para su codificación. Ni siquiera es necesario algún tratamiento especial en la decodificación fenotípica más que señalar cuáles neuronas son las de salida. Obviamente este señalamiento es importante al momento de evaluar la red para determinar a partir de cuáles neuronas se construye el vector resultado.

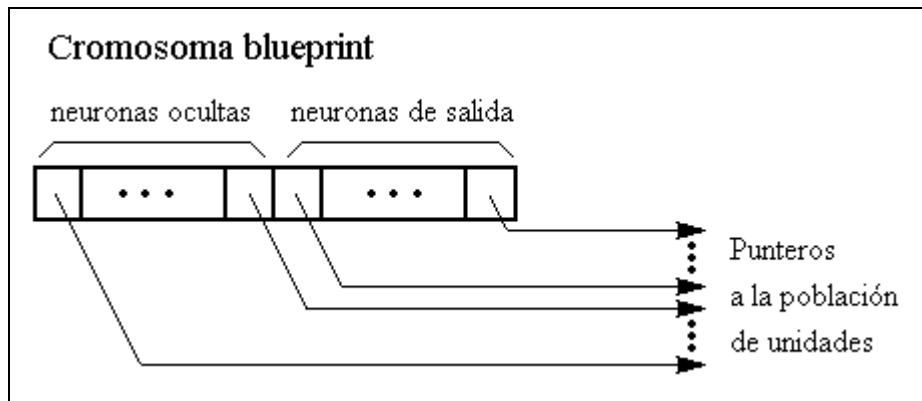


Ilustración 107 Esquema de codificación del cromosoma blueprint.

La codificación de unidades no sufre cambio alguno, sólo se modifica la interpretación de la etiqueta del par (e, p) en el momento de la construcción del fenotipo.

Al igual que como sucede en la decodificación ya presentada para redes recurrentes en la estrategia “Neuroevolución simple”, si numeramos todas las neuronas comenzando por cero, colocando primero las de entrada, luego las ocultas y por último las neuronas de salida, el campo e se interpreta de la siguiente manera:

Sean n_E , n_O y n_S el número de neuronas de entrada, ocultas y de salida respectivamente, y sea D el valor decimal del campo e :

Si $D > 127 \rightarrow$ la conexión es hacia la neurona numerada $(D \bmod (n_O + n_S)) + n_E$

Si $D \leq 127 \rightarrow$ la conexión es desde la neurona de entrada $(D \bmod n_E)$

Veamos un ejemplo para una RNA recurrente con 5 neuronas de entrada 2 ocultas y 2 de salida. Sin conexión de tendencia y no multisigmoide –por simplicidad–. Supongamos que se codifican sólo dos conexiones por unidad. Así cada blueprint se forma por cuatro punteros a individuos de la población de unidades.

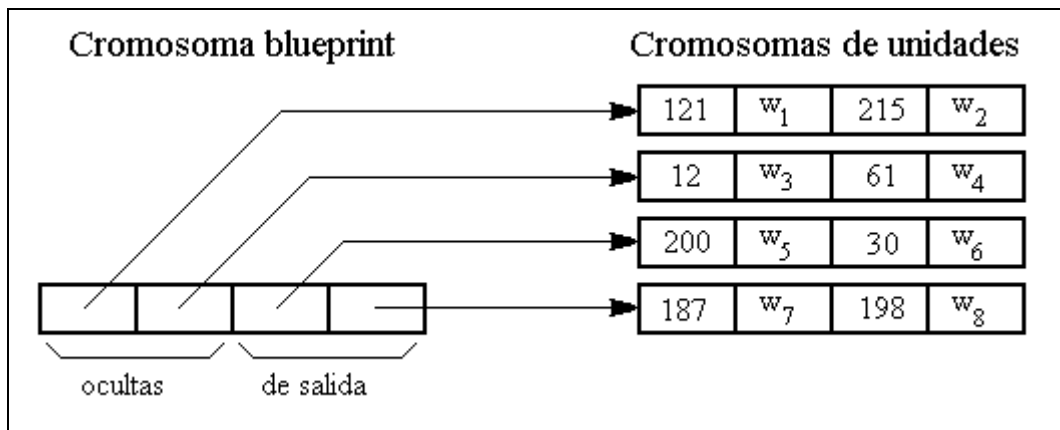


Ilustración 108 Ejemplo de codificación para RNA de 2 neuronas ocultas y 2 neuronas de salida. Por simplicidad no se colocan los valores de los pesos.

Suponiendo que se trata de una red neuronal recurrente con 5 neuronas de entrada, la decodificación del fenotipo requiere los siguientes cálculos:

Cálculo	Resultado fenotípico
$(121 \text{ mod } 5) = 1$	Conexión desde la neurona 1
$(215 \text{ mod } 4) + 5 = 8$	Conexión hacia la neurona 8
$(12 \text{ mod } 5) = 2$	Conexión desde la neurona 2
$(61 \text{ mod } 5) = 1$	Conexión desde la neurona 1
$(200 \text{ mod } 4) + 5 = 5$	Conexión hacia la neurona 5
$(30 \text{ mod } 5) = 0$	Conexión desde la neurona 0
$(187 \text{ mod } 4) + 5 = 8$	Conexión hacia la neurona 8
$(198 \text{ mod } 4) + 5 = 7$	Conexión hacia la neurona 7

Finalmente el fenotipo queda construido de la siguiente manera:

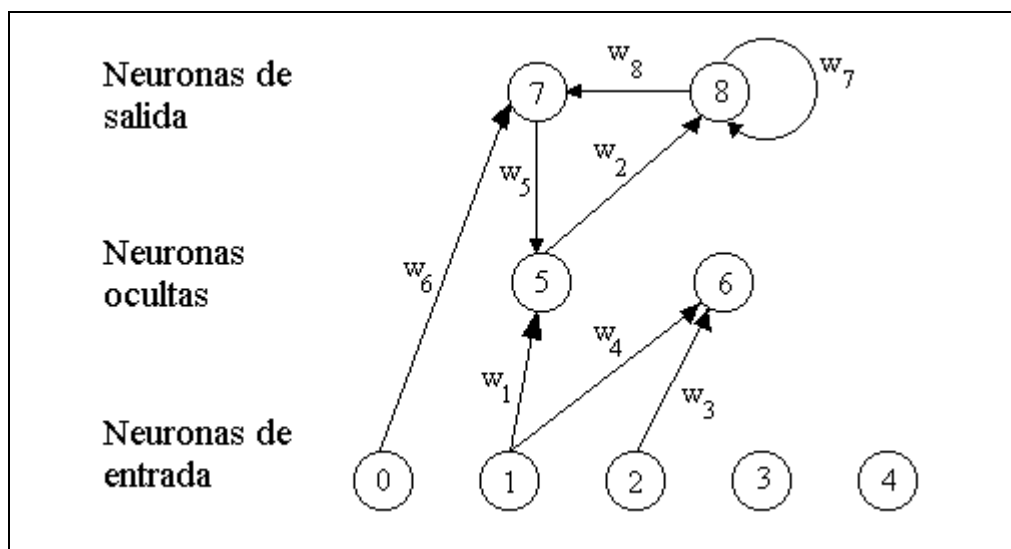


Ilustración 109 Fenotipo correspondiente a la codificación genética presentada en la Ilustración 108.

En este ejemplo se ve claramente la importancia de la cantidad de conexiones por neurona que se codifiquen en la población de unidades. Para simplificar el gráfico se ha elegido sólo 2, pero tal escasez de conexiones si bien arroja como resultados redes simples –característica deseable– aumenta el riesgo de dejar neuronas de entrada desconectadas.

Ya se ha mencionado que la cantidad de neuronas de entrada no se somete a evolución pues queda definida por el problema, al menos la mayor parte de las veces. Por lo tanto habitualmente no se espera que la evolución reduzca la complejidad de las redes a costa de sacrificar información de entrada.

Creo que un buen criterio para seleccionar la cantidad de conexiones a codificar en los cromosomas de las unidades es el número total de conexiones que una neurona podría tener. En el tipo de red recurrente que se define en el presente trabajo cada neurona codificada –las de entrada no se codifican– puede conectarse con cualquiera de las otras, incluso consigo misma. Así en el ejemplo anterior, una buena elección sería codificar las unidades como una secuencia de 9 conexiones.

Como ya se ha explicado antes, la evolución podrá generar neuronas con menor cantidad de conexiones que el máximo especificado, ya que más de una puede conectar al mismo par de neuronas en el mismo sentido. En tal caso, la suma de sus pesos representa el peso de la única conexión que resulta de esa situación.

Así finaliza la exposición sobre este método. Obsérvese que SANE ha sido, y aún lo será, referenciado varias veces en el presente trabajo, incluso desde los primeros capítulos. Su importancia radica también en el hecho que existe mucha documentación que confirma sus ventajas respecto a los métodos que evolucionan poblaciones de redes neuronales completas. Además es frecuentemente citado por otros investigadores. Todo ello lo convierte en un método atractivo para tomarlo como referente comparativo en el desarrollo de otras estrategias.

A continuación se presenta otro método de neuroevolución conocida con el nombre de ESP –Enforced subpopulation– que, al igual que SANE, fue desarrollada por investigadores del Departamento de Ciencias de la Computación de la Universidad de Texas y que puede considerarse una variante de este último. También se expondrán los detalles propios de la implementación de ESP realizada en el contexto del presente trabajo.

ESP –Enforced subpopulations–

Faustino Gomez y Risto Miikkulainen en un artículo titulado “Incremental Evolution of Complex General Behavior” [Gomez97] presentan un método de neuroevolución en el que el objetivo de una tarea compleja es transformado en una secuencia ordenada de objetivos parciales de complejidad creciente. El último elemento de esta secuencia corresponde al objetivo global planteado.

Si bien no se realiza en el presente trabajo el tipo de evolución incremental presentada por estos autores, sí se implementa ESP –variante de SANE– que introdujeron en su artículo. No obstante, antes de describir el método ESP, se expondrá brevemente el concepto de evolución incremental por dos razones: 1) Para dejar claro el contexto en el que fue desarrollado ESP; y 2) Porque el concepto en sí mismo es relevante pues más adelante, al exponer GNE, se presentará una forma de evolución incremental, aunque lograda de manera distinta a la que estos autores utilizan aquí.

Más precisamente si O es el objetivo a alcanzar, el método propone encontrar una serie de objetivos más simples de complejidad creciente $O_1, O_2, \dots, O_n = O$ proponiendo hacer que la

evolución primero alcance el objetivo O_i y a partir de allí, se prosiga la evolución hacia el objetivo O_{i+1} . Aquí tiene validez aquella vieja frase muchas veces escuchada “*primero se debe aprender a caminar antes de aprender a correr*”.

Para llevar a cabo esta tarea utilizan como método de neuroevolución la variante de SANE que nos interesa –ESP, que explicaremos enseguida– y un sistema de codificación denominado Delta-coding –desarrollado por Whitley y colaboradores en 1991–.

Whitley propuso este sistema de codificación como medio para seguir impulsando la evolución una vez que la población de un Algoritmo Genético haya convergido. En realidad no es sólo un sistema de codificación sino una estrategia evolucionaria completa.

Cuando la población converge, la evolución deja de avanzar debido a la falta de diversidad genética de la que se nutre el algoritmo para funcionar –ver “Diversidad Genética y Convergencia” en la pág. 64–. Llegado este punto de la evolución Whitley propone salvar el mejor individuo de la población, quien representa la mejor solución hallada por un algoritmo evolucionario clásico, e inicializar una nueva población de individuos ligeramente distintos llamados Δ -cromosomas. Estos Δ -cromosomas poseen la misma longitud –número de genes– que el individuo salvado, pero están constituido por Δ -valores que representan diferencias respecto de aquel mejor individuo salvado. Comienza entonces la evolución de la nueva población así creada que puede utilizar el mismo algoritmo aplicado hasta el momento con la siguiente salvedad: en la etapa de evaluación los fenotipos son creados a partir de la adición de los Δ -valores al mejor individuo salvado.

Obsérvese que esta estrategia constituye una manera de incluir diversidad genética en el momento que la población adolece por falta de ella. Es por eso que en la inicialización de los Δ -cromosomas, los Δ -valores deben constituir en su mayoría diferencias muy pequeñas –0 en caso de una codificación binaria–, para no apartarse demasiado de la solución hallada previamente que se intenta mejorar. Por ello, generalmente se usa una distribución de probabilidades de Cauchy.

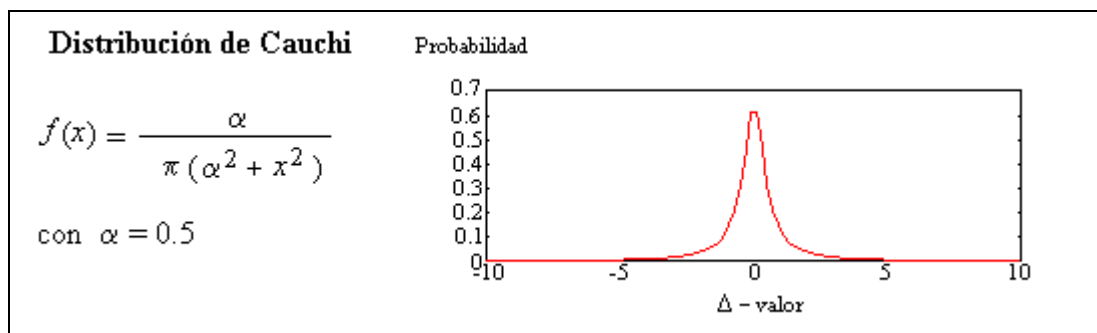


Ilustración 110 Distribución de Cauchy.

Puede considerarse como una técnica exploratoria donde se introducen pequeñas mutaciones aleatorias para impulsar la evolución detenida por la falta de diversidad en un posible subóptimo. Así Delta-Coding explora el hiperespacio en las vecindades de la mejor solución hallada.

Debe observarse que este método puede aplicarse más de una vez cuando la población de Δ -cromosomas haya convergido. Esta posibilidad ha sido aprovechada por Gómez y Miikkulainen, y en su trabajo una vez que se ha alcanzado un objetivo parcial, se prosigue la evolución hacia el próximo utilizando esta codificación.

$$\text{Estado Inicial} \longrightarrow o_1 \xrightarrow{\Delta} o_2 \xrightarrow{\Delta} o_3 \xrightarrow{\Delta} \dots \xrightarrow{\Delta} o_n = o$$

En este método no se plantea ninguna forma automática de precisar los subobjetivos y por ende necesitan ser definidos manualmente para cada problema a resolver. Como sus mismos autores lo comentan en su artículo, la división en forma automática de las etapas del método está abierta a investigación.

Luego de esta pequeña descripción del método de evolución incremental de Gómez y Miikkulainen se desplegará el método ESP, una variante de SANE que se utiliza para la evolución en cada una de las etapas descriptas, y que también se ha implementado en el presente trabajo.

En Enforced Sub-Populations, como ocurre en SANE, se evolucionan individuos que representan neuronas individuales, en lugar de redes completas. De este grupo de neuronas se seleccionan subconjuntos para formar las redes neuronales que resuelvan el problema planteado.

ESP se aparta del método de Moriarty aplicando una serie de restricciones en la forma de seleccionar las neuronas de la población de unidades para construir las redes neuronales – potenciales soluciones al problema planteado–.

ESP fue desarrollado para conseguir redes recurrentes¹ totalmente conectadas con capacidad de memoria a corto plazo. A diferencia de SANE se mantienen subpoblaciones separadas en la población de unidades sin cruzamiento entre ellas². Se crean tantas poblaciones como neuronas que no sean de entrada, tenga la red neuronal que se desea hallar. Alguna de estas neuronas codificadas se considerarán de salida y otras ocultas.

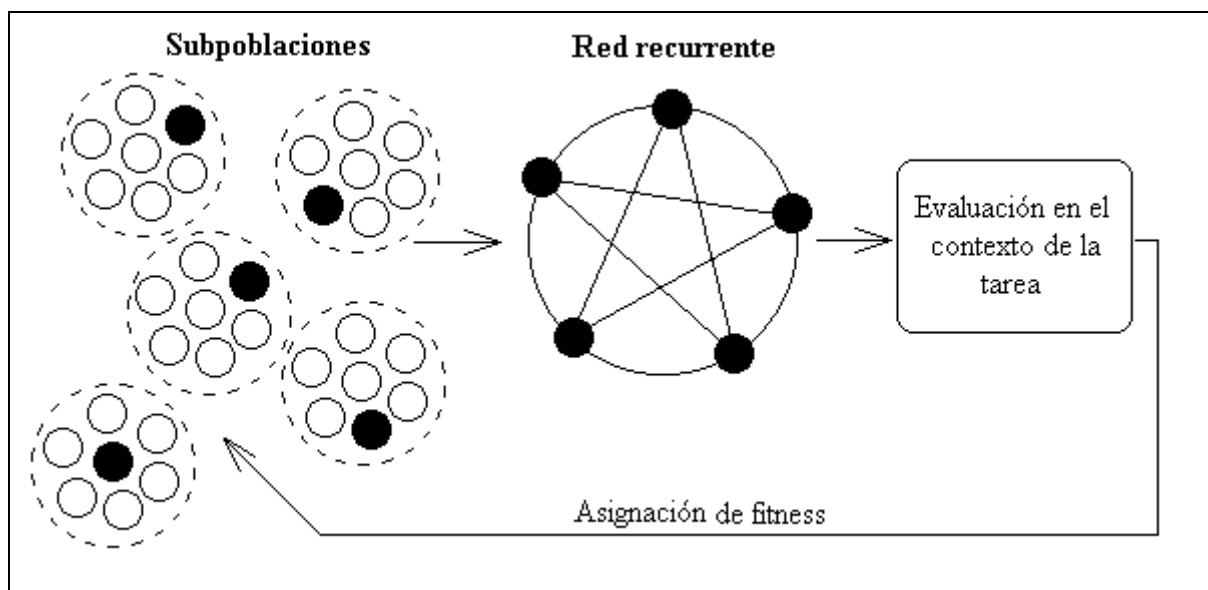


Ilustración 111 Modelo de neuroevolución ESP.

¹ Recordemos que las redes neuronales recurrentes –también denominadas FeedBack– son en general más potentes que las feedforward pues en el sistema fluye información relacionada con estímulos recibidos anteriormente, así se puede afirmar que son capaces de conseguir una memoria a corto plazo, precisamente esto es lo que se testea en el artículo de Gómez y Miikkulainen [Gomez97]. Si desea ver información sobre clasificación de RNA vea “Arquitectura neuronal” en la pág. 151.

² No conozco la razón por la que los autores han preferido hablar de subpoblaciones cuando en realidad se trata de poblaciones independientes que evolucionan separadamente sin ningún tipo de cruzamiento ni migraciones.

En la ilustración anterior se evidencia la característica típica de los algoritmos evolucionarios que trabajan con el concepto de aislamiento geográfico –ver “Aislamiento geográfico” en la pág. 73– que ESP importa para conseguir la especialización de las neuronas que requiere la resolución de tareas complejas.

Debo mencionar que, aunque no se especifica en el artículo que introduce este método [Gomez97], ESP es en realidad una variante de Neuron SANE más que de SANE ya que en él no existe población de blueprints. Existen publicados otros artículos más recientes del Departamento de Ciencias de la Computación de la Universidad de Texas que utilizan ESP [Gomez99] [HanYong01] [PerezBergquist01] sin hacer ninguna modificación explícita al método salvo por su utilización para crear redes feedforward. Así puede decirse que este método no ha sido extendido, al menos por el grupo de investigación donde fue concebido.

En cada una de las subpoblaciones de unidades ESP lleva a cabo un proceso evolutivo independiente igual al que utiliza SANE en la población de neuronas, marcado por un fuerte elitismo y selección agresiva –ver “Reproducción en la población de unidades” en la pág. 210–. Sin embargo, a diferencia de SANE¹, el fitness de las unidades se calcula como en Neuron SANE, promediando el valor de aptitud obtenido por cada red neuronal en la que haya participado.

Debe quedar claro que en este método no existe registro de las buenas combinaciones de neuronas halladas como lo hace SANE a través de los blueprints. Las unidades son elegidas en forma aleatoria de cada subpoblación para construir las redes neuronales que serán evaluadas. Esto presupone una estrategia distinta que la presentada en SANE en relación al tamaño de las poblaciones. Aquí se hace necesario una relación entre la cantidad de RNA a construir y el tamaño de las subpoblaciones que asegure que todas las unidades tendrán chance de ser elegidas para formar parte de al menos una red.

Por ello en el software que acompaña el presente trabajo, cuando el usuario define el valor de los parámetros para iniciar una prueba evolutiva, en caso de seleccionar la estrategia ESP, se sugiere en una leyenda que el tamaño de cada una de las subpoblaciones no sea mayor que $\frac{1}{4}$ del número de redes neuronales que se testearán –población de agentes–. No obstante, muchas veces esta relación no alcanza. Si bien no se hace mención explícita de cuál debería ser esta relación en el artículo que introduce el método [Gomez97] en las pruebas presentadas, el autor trabaja con subpoblaciones de 40 unidades y se construyen 400 redes para su evaluación. La misma relación (1 a 10) aparece en el artículo de Chen Han Yong [HanYong01] quien evoluciona subpoblaciones de 100 unidades construyendo 1000 redes neuronales para poder evaluarlas.

Según sus autores ESP acelera la evolución respecto a SANE por dos razones:

1. Las subpoblaciones que gradualmente se forman en ambas versiones de SANE – recordemos que hay presión evolucionaria para conseguir distintas especializaciones en una única población– son circunscriptas explícitamente por el diseño en ESP. Así, las especies no deben autoorganizarse a partir de una única población a medida que avanza la evolución y con ella el proceso de especialización. Tampoco se ven perjudicadas por el cruzamiento de individuos pertenecientes a distintas especializaciones con roles muy diferentes en las RNA que sí puede ocurrir en Neuron SANE o SANE.

¹ Recordemos que SANE calcula el fitness de los individuos en la población de unidades como la suma de los cinco mejores fitness obtenidos por las RNA donde dicha unidad haya intervenido.

2. La red se forma siempre con un representante de cada especialización, mientras que en ambas versiones de SANE pueden construirse redes con varios miembros de una determinada clase y sufrir la ausencia de otros. Esto también posibilita una asignación de fitness más justa a las neuronas, que siempre son evaluadas respecto al mismo rol que le toca representar dentro de la red neuronal.

Sin embargo, según los autores la mayor contribución de ESP respecto a Neuron SANE es permitir la evolución de redes recurrentes. Recordemos que en la arquitectura feedforward de dos capas que usa Neuron SANE las conexiones de las neuronas ocultas se realizan siempre con neuronas fijas que no son alcanzadas por la evolución. Sin embargo al permitir conexiones recurrentes éstas podrán unir dos neuronas generadas por evolución. El comportamiento de una neurona dentro de una red neuronal puede cambiar dramáticamente según cuáles sean sus congéneres conectados a ella.

Debido a que en Neuron SANE las redes neuronales se forman por la elección aleatoria de neuronas de una única población de unidades, la asignación del fitness a estas unidades en redes neuronales recurrentes no será siempre precisa pues su desempeño dependerá de los compañeros de turno –aleatoriamente elegidos– conectados a ella en la red neuronal. Por el contrario las subpoblaciones de ESP realizan una evaluación de fitness más consistente. Una conexión recurrente siempre se asociará a dos subpoblaciones determinadas, y como éstas evolucionan especializaciones a medida que la evolución avanza habrá una creciente certeza sobre el tipo de neuronas que serán conectadas.

Sin pretender contradecir a estos investigadores supongo que la dificultad de Neuron SANE para evolucionar redes recurrentes debe subsanarse con la población de blueprint que introdujo SANE pues la combinación de neuronas en las redes ya no es en forma aleatoria sino que se realiza a través del registro mantenido por los blueprints. Por ello he implementado SANE para redes recurrentes. De hecho, confirmando tal suposición, el desempeño evidenciado por SANE sobre redes recurrentes ha sido marcadamente superior al logrado sobre redes feedforward, al menos en el dominio de problemas aquí elegido –ver Ilustración 138, en la pág. 258 del capítulo “Evaluación y Análisis”–.

Como debilidad del método ESP puede citarse que no existe ningún mecanismo adicional para mantener la diversidad genética en cada una de las subpoblaciones, así a medida que la evolución avanza cada subpoblación perderá diversidad hasta converger. Claro está que en el artículo de Gómez que introduce ESP en el contexto de una evolución incremental, llegado este momento se utiliza la codificación Delta-coding ya comentada.

Extensión al método ESP implementada

Antes de describir la extensión al método que he implementado, debo señalar con precisión los aspectos sobre los cuales me he apartado ligeramente de la versión original. Aunque creo que no forma parte de la esencia de ESP, Gómez y Miikkulainen han utilizado la representación real en los cromosomas mientras que en este trabajo, siguiendo la línea de desarrollo hasta el momento presentada, se utilizó la binaria.

Además, a diferencia de la versión aquí implementada, en ESP original no se codifica información de la segunda neurona que interviene en la conexión, pues está implícita en la posición que ocupa la conexión dentro del cromosoma. Esto puede ser de esta forma porque ESP original sólo evoluciona redes neuronales recurrentes totalmente conectadas lo que exige codificar una conexión por cada neurona de la red. Así un cromosoma de una unidad simplemente es un vector de números reales que representan los pesos de cada conexión.

Veamos ahora qué detalles se han agregado, que no existen en la versión original. Al igual que como se hiciera con el método SANE, se ha adecuado ESP para conseguir con él la evolución de redes neuronales con y sin término de tendencia, como así también con y sin característica multisigmoide. Para ello se utiliza en cada una de las subpoblaciones de unidades la codificación usada en la versión extendida de SANE.

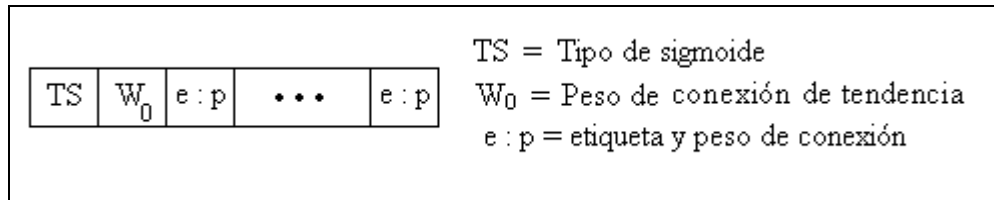


Ilustración 112 Cromosoma de las subpoblaciones de unidades.

También se ha dotado al método de libertad para evolucionar la conectividad de las redes resultantes. Aquí se utiliza ESP para evolucionar tanto redes feedforward como recurrentes con esquema de conexión libre. Para la construcción del fenotipo se utiliza el mismo proceso que usa SANE para interpretar las etiquetas de las conexiones codificadas en las unidades, según sean redes feedforward o recurrentes. Pero a diferencia de SANE que utiliza blueprints, por cada neurona oculta y de salida que se necesiten codificar, se asocia una subpoblación de unidades, de las que se elegirá una al azar para construir las redes neuronales.

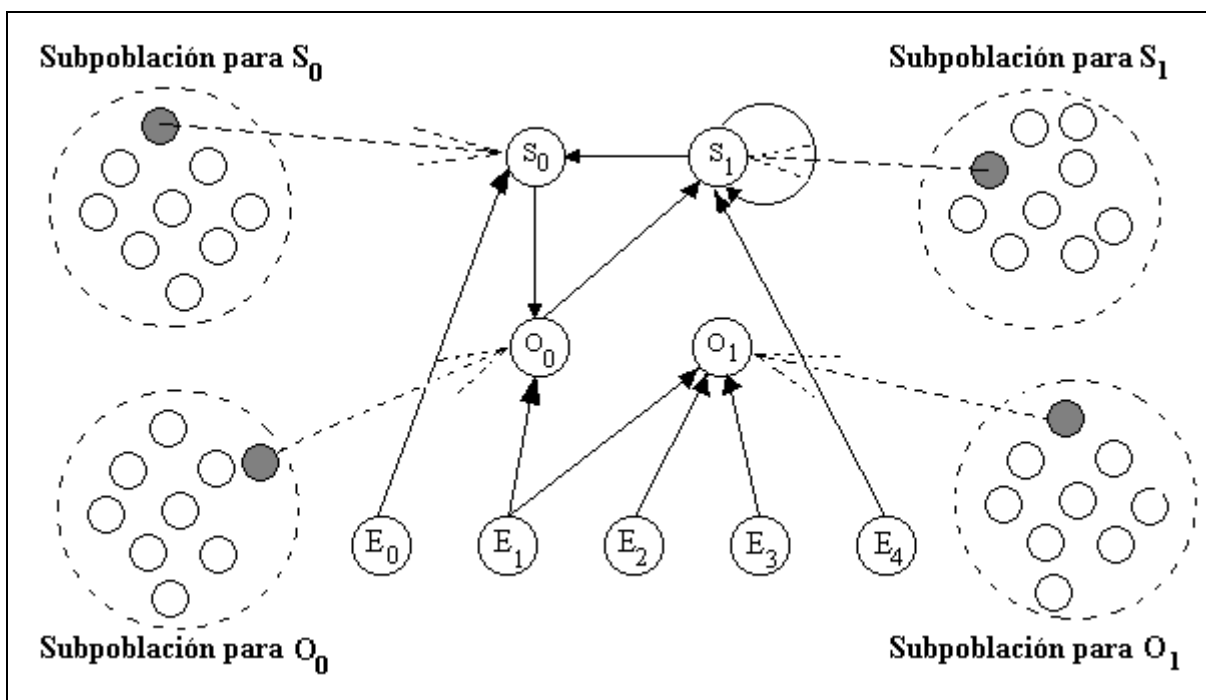


Ilustración 113 Ejemplo de construcción del fenotipo de una RNA recurrente. Las conexiones y los pesos vienen codificados en la neurona elegida al azar de la subpoblación de unidades correspondiente.

Debilidad del método

Ya se ha mencionado que ESP necesita construir un gran número de redes neuronales para asegurar que cada unidad de las subpoblaciones tenga chance de participar en varias redes. Veamos por qué es tan importante que esto suceda.

Recordemos que el fitness para cada unidad se calcula promediando el valor de aptitud obtenido por todas las redes donde haya participado. Debido a que las redes neuronales se construyen tomando al azar un individuo de cada subpoblación existe el riesgo de que no todas las unidades sean seleccionadas. Una unidad no seleccionada tendrá valor de fitness 0, aunque sea la mejor unidad existente en la subpoblación.

Por lo tanto, para una asignación precisa del valor de fitness –crucial en cualquier algoritmo evolucionario– debe maximizarse la probabilidad de que toda unidad participe en alguna –en lo posible más de una– red neuronal. Esto se consigue simplemente construyendo un número mayor de redes que individuos haya en cada subpoblación de unidades.

Este requisito en la cantidad de redes neuronales que deben construirse para su evaluación, lo consideramos un elemento negativo. Quizá en otros dominios de problemas no se evidencie pues se necesite naturalmente trabajar con grandes poblaciones.

Sin embargo en las pruebas realizadas aquí para el conjunto de problemas planteado –ver “Dominio de problemas a resolver” al comienzo de este capítulo– se han obtenido buenos resultados, utilizando otros métodos, con poblaciones relativamente pequeñas de 80 redes neuronales. Para mantener la misma relación utilizada por Gómez en su artículo [Gomez97] (40 contra 400) o la utilizada por Chen Han Yong [HanYong01] (100 contra 1000) deberíamos utilizar subpoblaciones de tan sólo 8 unidades. He probado esta relación con no buenos resultados. Claro está que 8 es un número muy bajo de individuos para pretender que un Algoritmo Genético trabaje correctamente.

La solución evidente es trabajar con una población de agentes –redes neuronales– mucho mayor, permitiendo así subpoblaciones de tamaño más apropiado. Esta solución consume mucho más recursos de cómputo pues precisamente en la evaluación de las redes neuronales es donde está concentrada la mayor carga de procesamiento.

Intentando resolver esta dificultad se plantea la siguiente estrategia que puede considerarse una hibridación entre SANE y ESP.

ESP con población de blueprints

Como se mencionara párrafos arriba el requerimiento de evaluación de una mayor cantidad de agentes proviene de la necesidad de promover la participación de las unidades en varias redes neuronales para su evaluación. A su vez, esta necesidad de participación se origina en la forma de construcción de las redes a través de la selección aleatoria de una unidad por cada subpoblación.

La misma situación se daba en Neuron SANE que, sin embargo, ya no se manifiesta en SANE gracias a la población de blueprints que guarda el registro de las neuronas y combinaciones de ellas exitosas. De esta manera deja de ser necesaria la construcción de gran cantidad de redes que garantice la participación de todas las neuronas, puesto que sólo participarán aquellas unidades apuntadas desde algún blueprint.

Según Moriarty [Moriarty97_1] SANE tuerce el índice de participación hacia las mejores neuronas de la población de unidades y deja muchas otras sin evaluar –fitness cero– en cada generación. Una neurona no evaluada seguramente será rankeada por debajo del primer 50% y seguramente será reemplazada. Obsérvese que para SANE es importante la cantidad de participaciones pues su fitness se calcula sumando –no promediando– los fitness de las cinco mejores redes en que ha participado.

En SANE la relación entre número de redes neuronales a evaluar –tamaño de la población de blueprints– y el número de neuronas en la población de unidades es inversa a la requerida en

ESP. Como no todas las neuronas participarán en la contracción de redes, sino que paulatinamente al ir avanzando la evolución irán participando las mejores, conviene partir con una población de unidades bastante mayor que la de blueprint. Esta situación no es molesta puesto que una población grande de unidades presupone mayor procesamiento en la fase de reproducción y reemplazo –fracciones de segundo–, en cambio una población de blueprints –redes neuronales– grande, requiere mayor procesamiento en la fase de evaluación de las RNA –varios minutos de procesamiento–.

Las ventajas de ESP en relación a la primera versión de SANE –Neuron SANE– ya se han explicado en este capítulo, básicamente se refieren a los beneficios del aislamiento geográfico para construir neuronas especializadas y a la evaluación más precisa del fitness pues las neuronas son evaluadas –indirectamente a través de las redes donde participan– siempre respecto al mismo rol que le toca interpretar dentro de la red.

Juntando las ventajas de ambos métodos he implementado con buenos resultados la estrategia neuroevolutiva que llamaré “ESP con población de blueprints”. Puede analizarse desde dos puntos de vista: concibiéndola como una extensión de SANE con subpoblaciones de unidades, o bien como una extensión de ESP con población de blueprints. Obviamente, no cabe ninguna duda que se trata de una hibridación de estas dos estrategias.

La idea es simple. Se tiene una población de blueprints que representan a las redes neuronales. Cada cromosoma blueprint está conformado por un arreglo de punteros a unidades. Para cada posición de este arreglo se asocia una subpoblación de unidades distinta. Sólo se puede apuntar a una subpoblación desde la misma posición en cada uno de los blueprints.

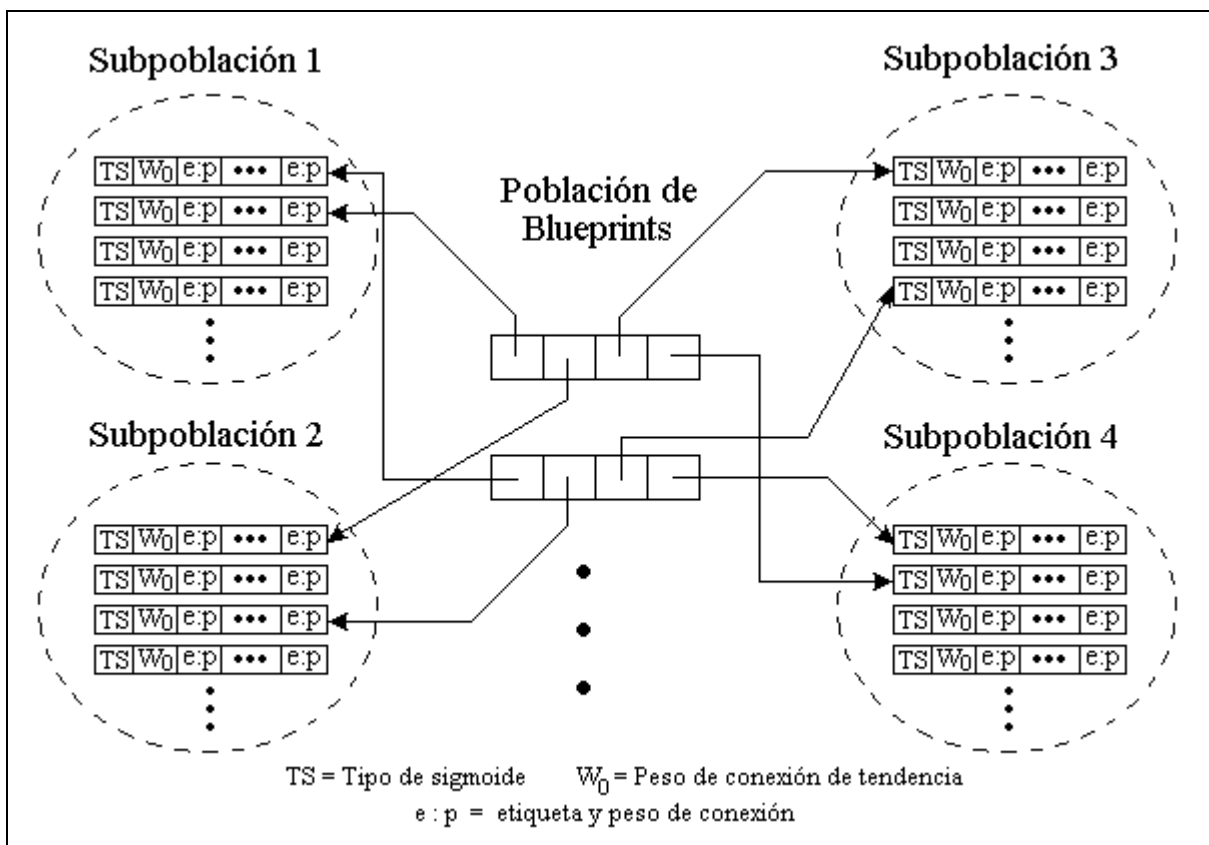


Ilustración 114 Esquema de codificación para la estrategia ESP con población de blueprints.

La decodificación del fenotipo es idéntica a la expuesta cuando se analizó la extensión al método SANE implementada. Es decir, a la secuencia de conexiones que definen una neurona

en cada una de las subpoblaciones, se antepone los campos correspondientes al tipo de sigmoide (TS) y el peso de la conexión de tendencia (W_0) para soportar estas dos características implementadas en todos los métodos aquí desarrollados.

Si se trata de redes feedforward sólo se codifican en blueprints las neuronas ocultas, por el contrario, en las redes recurrentes se codifican neuronas ocultas y neuronas de salida. La interpretación del valor de la etiqueta e en el par que identifica la conexión sigue interpretándose de la forma ya expuesta:

Tipo de Red	D	Conexión
Feedforward	≤ 127	desde la neurona de entrada ($D \bmod n_E$)
	> 127	hacia la neurona de salida ($D \bmod n_S$)
Recurrente	≤ 127	desde la neurona de entrada ($D \bmod n_E$)
	> 127	hacia la neurona ($D \bmod (n_O + n_S) + n_E$)

Siendo D el valor decimal de la etiqueta e , n_E , n_O y n_S la cantidad de neuronas de entrada, ocultas y de salida respectivamente que conforman la red a construir.

El algoritmo evolutivo es una adaptación de SANE, comienza con una inicialización aleatoria de cada una de las subpoblaciones de unidades y de la población de blueprints. Luego realiza los siguientes pasos:

1. Borrar el valor de fitness de cada neurona y blueprint.
2. Seleccionar una neurona de cada subpoblación utilizando un blueprint.
3. Crear una RNA con las neuronas seleccionadas.
4. Evaluar la performance de la red en la resolución de la tarea dada.
5. Asignar al blueprint el valor de fitness de acuerdo con la performance de la RNA evaluada.
6. Repetir los pasos 2 a 5 para cada individuo de la población de blueprint.
7. Asignar a cada neurona un fitness calculado como la suma de los 5 mejores fitness conseguido por las RNA en las que ha participado.
8. Obtener la siguiente generación en cada una de las subpoblaciones de unidades a través de la selección, reproducción y reemplazo.
9. Obtener la siguiente generación en la población de blueprints a través de la selección, reproducción y reemplazo.
10. Si no se cumple alguna condición de terminación volver al punto 1.

Al igual que SANE se utiliza la misma selección agresiva y reemplazo elitista en cada una de las subpoblaciones y en la población de blueprints. Recordemos brevemente que los parentales a reproducirse son elegidos del primer 25% ranqueado y su descendencia reemplazará la peor mitad de la población.

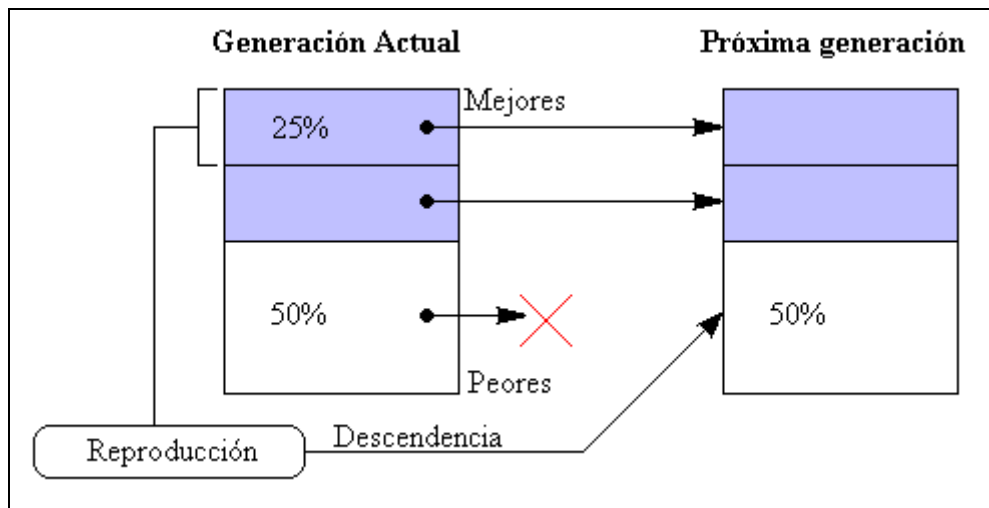


Ilustración 115 ESP con población de blueprints utiliza el mismo esquema de selección y reemplazo utilizado por SANE en cada una de las subpoblaciones de unidades y en la población de blueprints.

También se utilizan los mismos mecanismos de reproducción de SANE, es decir, en todas las poblaciones se aplica crossover de un punto para conseguir el primer descendiente, el segundo se obtiene por copia de uno de los padres. Mutación binaria con probabilidad 0.001 por bits en cada una de las subpoblaciones de unidades y el doble mecanismo de mutación en la población de blueprints –apuntar a otra unidad de la subpoblación correspondiente elegida al azar con probabilidad 0.01 y apuntar a un descendiente de la unidad apuntada con probabilidad 0.5–. Para más detalles consultar el apartado “Selección, reproducción y reemplazo” en la pág. 209 que forma parte de nuestra exposición sobre el método SANE.

Como puede observarse en las pruebas realizadas presentadas en el capítulo “Evaluación y Análisis”, este método supera tanto a SANE como a ESP en escenarios complejos, pues es capaz de combinar las virtudes de ambos métodos –ver a partir de la Ilustración 153 en la página 270–.

A continuación presentaré la nueva estrategia neuroevolutiva desarrollada con la intención de demostrar que más de una red neuronal, cooperando en la realización de una tarea puede ser más efectiva que si todo el trabajo recae sobre una sola, tal cual ocurre en todos los métodos presentados hasta el momento. Además pretendo demostrar que no solo el resultado puede ser mejor, sino que el proceso de evolución para conseguirlo puede verse acelerado.

Grupos Neuronales Evolutivos –GNE–

Como ya se ha comentado en el capítulo anterior el estudio de las redes neuronales artificiales ha surgido para dar respuesta a problemas de difícil o impracticable solución algorítmica, intentando emular en la medida en que sea posible, el funcionamiento del cerebro –animal o humano– en base al limitado conocimiento que se tiene de este órgano, el más evolucionado que existe.

Sin embargo, a menos que concibamos al cerebro como una única gran red neuronal, puede decirse que la mayor parte de los desarrollos en neuroevolución tienden a emular sólo pequeñas áreas del cerebro puesto que se focalizan en la obtención de una red neuronal que realice una determinada tarea.

En el capítulo “Redes Neuronales. Fundamentos biológicos” se han mencionado los estudios realizados en humanos por Jackson y Penfield que revelaron la existencia de una notable

localización funcional en la corteza cerebral –Ver “Zonas del cerebro. Distintas funciones” en la pág. 104–. También se hizo mención a otros centros neuronales en el interior del cerebro como el sistema límbico relacionado con las emociones y la actividad intelectual. Se ha comentado incluso un artículo de La Nación informando sobre los estudios publicados en la revista Nature, realizados por un científico argentino quien identificó neuronas individuales que intervienen en la imaginación.

Todo esto hace difícil concebir al cerebro como una única gran red neuronal, sino más bien se presenta como un conjunto complejo de centros –circuitos o redes– neuronales asociados a distintas actividades, aunque posiblemente interrelacionados y con capacidad de cubrir otras funciones en caso de necesidad¹.

Creo entonces que un intento más cercano de emular el cerebro humano debe ser por medio de un sistema de varias redes neuronales artificiales que de alguna manera se “repartan” la tarea o funciones a resolver. De acuerdo a la información de entrada recibida por el sistema, alguna estrategia de selección debe implementarse para determinar cuál o cuáles redes neuronales deben activarse.

Es sabido que, por un problema de escala –100.000 millones de neuronas en la especie humana– no es posible, al menos con la tecnología del momento, intentar un sistema que se acerque a la arquitectura cerebral –este ha sido y aún es uno de los puntos más débiles del Conexionismo–, pero podríamos aproximarnos, aunque sea conceptualmente, utilizando en lugar de una red neuronal, un grupo de redes que se autoorganicen con alguna división funcional para llevar a cabo una tarea compleja.

Ya en terreno de la Neuroevolución, debo admitir que cuando comencé a diseñar este método GNE, supuse que resultaría más fácil para un algoritmo genético evolucionar varias redes neuronales que resuelvan tareas sencillas supervisadas por algún mecanismo de activación selectiva, que evolucionar redes neuronales complejas que resuelvan por sí solas toda la tarea. Como se verá en lo que resta del capítulo, he podido comprobar que esta hipótesis es cierta si se imponen algunas restricciones que están presentes en GNE y que le confieren su calidad de método de neuroevolución incremental.

Pretendo aquí plantear no sólo una estrategia que resuelva mejor que otras algún determinado problema, sino una perspectiva poco explorada, un enfoque de trabajo, una línea de investigación a seguir, que ya fuera comentada hace casi diez años por Freeman y Skapura [Freeman93] y que creo aún no ha sido explotada convenientemente.

“Nos parece plausible (y también a Minsky y a Papert) que el cerebro esté compuesto por muchos sistemas diferentes paralelos, distribuidos, que llevan a cabo funciones bien definidas, pero bajo el control de un sistema secuencial de procesamiento con uno o más niveles. Para resolver el problema del escalado, quizá necesitemos aprender a combinar redes pequeñas, poniéndolas bajo el control de otras redes”

James Freeman. David Skapura.

Me resulta extraño no haber encontrado mayor cantidad de material bibliográfico que aborde directamente esta cuestión. Sin embargo, algunos artículos que comentaré manejan el concepto, aunque no de la manera en que pretendo plantearlo.

¹ Se ha observado que ciertas funciones cerebrales llevadas a cabo en una región del cerebro pueden “moverse” hacia otras localizaciones por ejemplo ante una lesión en dicha zona del cerebro.

Artículos relacionados con el tema

En un artículo de 1996 Xin Yao y Yong Liu [*XinYao96*] reconocen que existe muchos estudios sobre cómo evolucionar redes neuronales artificiales eficientemente, pero escasean – según los autores no existen– reportes sobre cómo generar la solución final a partir del proceso evolutivo.

En la neuroevolución estándar el objetivo es evolucionar una red neuronal para resolver satisfactoriamente una tarea. Por lo tanto se elige aquella que haya alcanzado el mejor fitness en la evolución. La práctica habitual es tomar el mejor individuo de la última generación. De esta manera son desechadas todas aquellas que, aunque con buen desempeño, posean un valor de aptitud inferior al logrado por la red neuronal campeona.

Según los autores esta práctica es aceptable cuando se trata de optimización evolucionaria. En el capítulo “Computación Evolutiva” vimos que los algoritmos evolucionarios son útiles como métodos de búsqueda y optimización. Pero en conjunción con las redes neuronales surge un nuevo concepto que ya vimos en el presente capítulo: el aprendizaje evolucionario. Distinto a lo que ocurre en otras clases de problemas, aquí no queda del todo claro qué significa un aprendizaje óptimo.

Por ejemplo, una red neuronal entrenada con el algoritmo Backpropagation hará mínima, idealmente, la función de error cuadrático medio para los datos de entrenamiento utilizados. Sin embargo esto no significa que dicha red haya logrado un aprendizaje óptimo puesto que la mejor RNA debería también presentar el mejor nivel de generalización. Como se comentó en el capítulo anterior, aún optimizando la función de error, la red puede sobreajustarse a los datos de entrenamiento “prestando atención” al ruido o a los detalles particulares de los mismos, haciéndola incapaz de reconocer similitudes, y fallando en la clasificación de datos nuevos.

Pero en la práctica no existe una buena forma de medir la generalización y por lo tanto usar el algoritmo Backpropagation es totalmente aceptable para conseguir aprendizaje supervisado. Sin embargo, en caso de aprendizaje evolucionario la situación es diferente debido a la existencia de una población completa de redes neuronales que podría aprovecharse.

Según los autores tomar el mejor individuo de la última generación y arrojar el resto de la población equivale a tratar al proceso de evolución como un proceso de optimización, en donde sólo interesa el mejor valor careciendo el segundo de cualquier utilidad. Pero tratándose de aprendizaje, es muy probable que el resto de la población posea información valiosa que mejore el nivel de generalización de la mejor RNA evolucionada.

No entraré en detalles sobre este artículo, simplemente mencionaré que en él se argumentan los beneficios de utilizar la información contenida en la población completa de la última generación de un proceso evolutivo. Se presentan varias estrategias. La más simple consiste en considerar a la población completa como el resultado de la evolución utilizando un mecanismo de votación para obtener la respuesta a cada entrada de datos. Así simplemente la salida de la población será aquella con mayor número de coincidencias entre las redes neuronales de la misma. Las otras estrategias utilizan combinaciones lineales de las salidas de un subconjunto de redes o de la población entera.

Más recientemente –2001–, Joseph Bruce y Risto Miikkulainen en su artículo titulado “Evolving Populations of Expert neural Networks” [*Bruce01*] trabajando sobre el problema de reconocimiento de caracteres escritos a mano alzada, demostraron que todas las redes neuronales de una población, producto de un proceso evolucionario, combinado con una técnica efectiva de especialización, pueden responder colectivamente mejor que cualquiera de ellas individualmente, aún mejor que la de mayor fitness.

En su análisis observaron que en un determinado momento de la evolución la red neuronal campeona era capaz de identificar correctamente el 64% de los caracteres, sin embargo, el 98% de ellos eran bien identificados por al menos un individuo de la población. Para que el método funcione es necesario que se mantenga un alto grado de diversidad para obtener distintas salidas. El problema a resolver es cómo determinar la red neuronal que está brindando la mejor respuesta en cada momento.

El reconocimiento óptico de caracteres pertenece a una clase distinta de problemas con un tipo también distinto de aprendizaje al utilizado en el presente trabajo. Es posible utilizar un algoritmo Backpropagation con aprendizaje supervisado por corrección de error. También es posible una solución evolutiva con aprendizaje evolucionario por refuerzo, pero es necesario contar con la señal de refuerzo en cada momento de evaluación. Así no cae en la categoría de “sequential decision task” en la que se circunscribe este trabajo. Salvando esta distancia, el artículo de Bruce y Miikkulainen representa un nuevo caso en el que la solución a un determinado problema es presentada como un conjunto de redes neuronales artificiales en lugar de una sola.

Es interesante la estrategia formulada por estos investigadores para determinar la respuesta de la población completa ante una entrada de datos. Sólo una red neuronal se activará para dar la respuesta en cada caso. La cuestión aquí es cómo identificarla.

La solución presentada por los autores fue evolucionar individuos que no solo produzcan la respuesta sino también un nivel de confianza acerca de la correctitud de la misma. Así la respuesta de la población a una determinada entrada puede definirse como la salida de la red neuronal que ante dicha entrada genera el mayor grado de confianza.

El nivel de confianza puede ser representado agregando una unidad de salida más a la red neuronal. Al presentar una entrada a la misma el valor de esta neurona de salida representará la confianza de la respuesta.

Estos autores definieron el rango de confianza como un número real entre 0 y 1. Para conseguir que las redes además de aprender la tarea específica aprendan a autocalificarse con el nivel de confianza correcto esta información debe intervenir en el cálculo del fitness. Éste se consigue en base a una serie de evaluaciones de la red sobre un conjunto de datos de entrada. En cada caso se sabe si la red reconoció correctamente el carácter presentado a la misma. El fitness se calcula de la siguiente manera.

$$f = \sum_i s(v_i)c(v_i)$$

Siendo v_i la entrada presentada, $c(v_i)$ el grado de confianza que arroja la red para la entrada v_i , $s(v_i) = 1$ si la respuesta es correcta y $s(v_i) = -1$ si es incorrecta. De esta manera una red es penalizada con $s(v_i)c(v_i)$ para cada respuesta equivocada y premiada con $s(v_i)c(v_i)$ en caso de acierto. Esto promueve a que las redes tengan un valor de confianza alto sólo para las decisiones que son probablemente correctas.

Luego de presentar resultados que confirman que la población entera puede resolver un problema supervisado mejor que cualquier red neuronal individualmente, los autores sugieren que dicha estrategia podría funcionar también en problemas de la clase “sequential decision task” como por ejemplo un controlador robótico. Obviamente, en este tipo de tareas donde no sólo se desconoce la salida esperada sino que se necesitan varias acciones para poder evaluar una secuencia completa de decisiones, no es posible utilizar una función de fitness como la presentada más arriba.

No obstante, en el artículo también se esboza una posible forma de atacar el problema de la asignación del valor de aptitud. Según Bruce y Miikkulainen un controlador robótico podría implementarse de la siguiente manera. Se presentaría a todas las redes de la población la entrada proveniente de los sensores del robot. La acción recomendada por su mayor nivel de confianza sería la seleccionada. Después de varias decisiones, una vez evaluado el fitness correspondiente a la secuencia completa, éste debería distribuirse a las redes de acuerdo al nivel de confianza esgrimido para cada salida y a la frecuencia con que hayan sido seleccionadas.

Aunque en primera instancia el cálculo del fitness propuesto parece dar pocas garantías de ser preciso, la situación es similar a lo que ocurre con la asignación de este valor a la población de unidades en SANE y ESP –ambas estrategias desarrolladas por el mismo equipo de investigadores al que pertenece el artículo que estoy comentando–. Y es precisamente con base en esta similitud y al éxito de estas estrategias que se sugiere como alternativa válida de asignación de fitness. De todas maneras, hasta el momento no han presentado ningún nuevo artículo relacionado con este tema.

Ambos artículos comentados, en especial este último, han servido como punto de partida en la búsqueda de un método que permita evolucionar grupos de redes neuronales artificiales para resolver eficientemente tareas complejas.

El planteo

Más arriba se ha comentado que los artículos encontrados donde se utilizan más de una red en la resolución de una tarea dada, no presentan la solución de la manera en que se pretende plantear aquí.

La idea que dio origen a GNE, no es evolucionar una única población de redes neuronales para utilizar más de un individuo de esa misma población en la realización de alguna tarea, como se hace en los dos artículos comentados anteriormente. Lo que se pretende aquí es evolucionar una población donde cada individuo de la misma represente un grupo de redes neuronales que resolverá el problema con la participación colectiva de sus miembros quienes deberán implementar alguna división funcional para aumentar su eficiencia.

La idea es utilizar los algoritmos genéticos asumiendo que cada individuo de la población representa por sí solo una solución potencial al problema –interpretación tradicional–. Pero como se pretende que la solución esté a cargo de un grupo de redes neuronales y no de una sola de ellas, es necesario definir una población de grupos y no de redes neuronales.

Los algoritmos genéticos, al trabajar con una población, en lugar de un solo punto en el espacio de búsqueda encuentran seguridad en la cantidad. Un conjunto de puntos bien adaptados reduce la probabilidad de alcanzar un óptimo local. Dicho de otra forma, operan simultáneamente con varias soluciones a la vez, en lugar de trabajar en forma secuencial. Por ello no se pretende perder esta ventaja aquí, y en consecuencia se plantea una población de grupos de redes neuronales.

Primeros intentos

He realizado varios intentos diferentes para obtener lo que acabo de plantear. Mi primera intención fue desarrollar una estrategia neuroevolutiva genérica, robusta, donde no fuera necesaria la intervención humana para adecuarla a la tarea particular a realizar. Como comentaré en breve, los resultados no fueron los esperados. En todos los casos la performance no superó la del método SANE implementado.

No daré aquí demasiados detalles sobre estos intentos porque, en definitiva, no obtuve los resultados esperados. Sin embargo, esbozaré la idea general para que se comprenda cuál ha sido el camino seguido en el desarrollo de GNE y porque aún no descarto completamente la iniciativa considerando que con ciertos ajustes todavía podría reflatarse¹.

Basado en la organización jerárquica de SANE decidí agregar un nuevo nivel codificando en cromosomas grupos de redes neuronales de igual forma que los blueprints codifican grupos interconectados de neuronas.

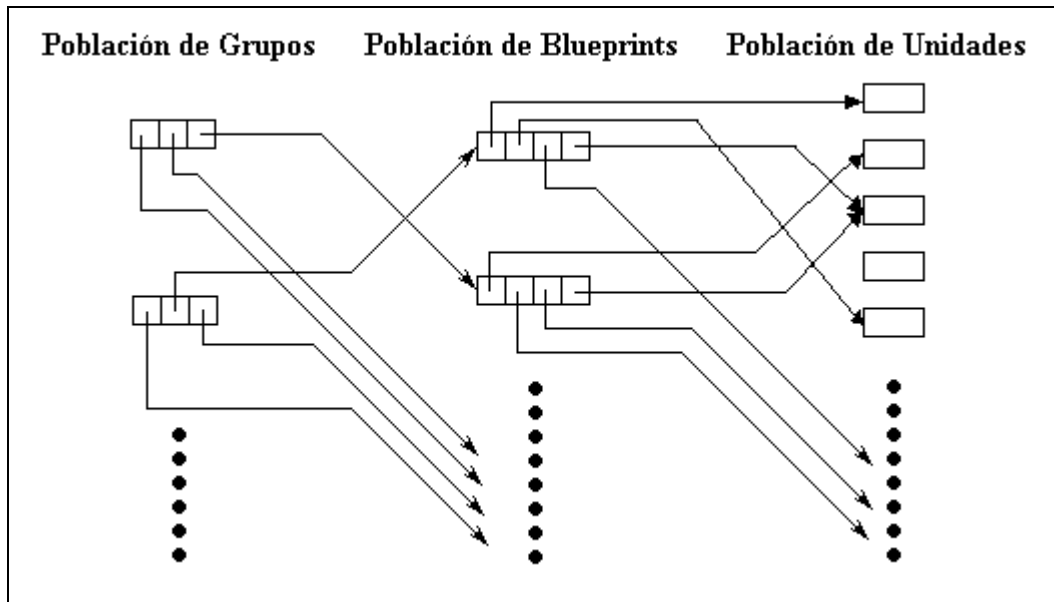


Ilustración 116 Primer intento para lograr la evolución de grupos neuronales.

Para la evaluación de cada grupo se adopta la estrategia de respuesta autocalicada con nivel de confianza ya comentada. Así, las redes neuronales utilizan una neurona de salida extra para estimar este nivel de confianza sobre la respuesta producida ante una entrada particular.

La asignación de fitness a los grupos neuronales se obtiene como resultado de la evaluación de los agentes en el entorno virtual. Cada agente es comandado por un grupo, así que el grado de aptitud conseguido en la simulación se corresponde directamente con el grado de adecuación del grupo neuronal.

Ahora bien, cada individuo de la población de blueprints puede participar en más de un grupo neuronal. Su fitness se calcula como la suma de los fitness obtenidos por los 5 mejores grupos en los que ha participado en la etapa de evaluación. Una vez asignado el fitness de los blueprints se puede repetir el mismo mecanismo para la población de unidades, calculando el valor de aptitud de cada neurona como la suma de los fitness obtenidos por las 5 mejores redes –blueprints– en las que hayan participado.

El algoritmo evolutivo se corresponde a una doble aplicación de SANE en dos niveles. Luego de la inicialización aleatoria de las tres poblaciones prosigue de la siguiente manera:

1. Borrar el valor de fitness de cada neurona, blueprint y grupo neuronal.

¹ Para ello es necesario un estudio más profundo y detallado de las particularidades del método e incluso del dominio de problemas a resolver, no adecuándose el tiempo requerido a los plazos fijados para este trabajo de grado, queda planteado como trabajo futuro.

2. Por cada individuo de la población de blueprints crear la RNA con las unidades apuntadas por el cromosoma blueprint.
3. Por cada individuo de la población de grupos neuronales crear el grupo con las redes apuntadas por el cromosoma grupo.
4. Evaluar la performance de cada grupo como controlador de los movimientos de un agente en la simulación.
5. Asignar a cada grupo el valor de fitness alcanzado por el agente comandado.
6. Asignar a cada blueprint el fitness calculado como la suma de los 5 mejores fitness conseguidos por los grupos en los que ha participado.
7. Asignar a cada neurona el fitness calculado como la suma de los 5 mejores fitness conseguido por las RNA en las que ha participado.
8. Obtener la siguiente generación en la población de unidades.
9. Obtener la siguiente generación en la población de blueprints
10. Obtener la siguiente generación en la población de grupos.
11. Si no se cumple alguna condición de terminación volver al punto 1.

La selección reproducción y reemplazo en las poblaciones de unidades y blueprints se realizan de igual forma que en SANE.

En la población de grupos se utiliza la misma estrategia de selección reproducción y reemplazo que se usa en la población de blueprints, incluido el doble mecanismo de mutación con las mismas probabilidades.

En esta iniciativa se ha utilizado SANE en dos niveles. He realizado unas cuantas combinaciones de métodos empleados en ambos niveles para conseguir la evolución de grupos: SANE y ESP, ESP y SANE, ESP y ESP además de otras donde utilicé SANE con población de blueprints.

No obstante, pese a todas las combinaciones testeadas, y manteniendo una carga computacional aceptable no se han obtenido resultados superiores a los arrojados por SANE. Debe considerarse que todos estos intentos presuponen un mayor requerimiento de procesamiento que las estrategias presentadas anteriormente. Sin embargo el concepto “carga computacional aceptable” puede tener distintos significados en diferentes situaciones o problemas. Dejamos entonces planteado como posible trabajo futuro la realización de estas mismas pruebas sobre una clase distinta de problemas.

Así es que finalmente, resignando robustez a favor de eficiencia, siendo necesario definir distintas etapas o subobjetivos en el proceso, he implementado un método de neuroevolución de grupos neuronales, con mejor performance que las estrategias presentadas anteriormente y que puede considerarse como un caso de neuroevolución incremental.

Descripción del método GNE

Como se indicó en el apartado anterior, los primeros intentos de conseguir grupos de redes neuronales que se autoorganicen para responder selectivamente a los estímulos recibidos en la realización de una determinada tarea, no han arrojado los resultados esperados. En todos los casos la evolución ha presentado el síndrome de convergencia lenta. Estudiando la manera en que se comportaron estas primeras aproximaciones se han obtenido las siguientes conclusiones:

1. La evolución simultánea de redes neuronales especializadas que se autoorganicen para determinar cuál de todas debe asumir la responsabilidad de proporcionar la respuesta, no representa una tarea sencilla de resolver para un algoritmo genético –AG difícil¹–.
2. En el dominio de problemas que se ha elegido para resolver, la estrategia de autocalificación de confianza es proclive a producir situaciones indeseables de comportamiento cíclico.

El segundo punto merece una explicación a través de un ejemplo. Puede parecer, en principio, que un grupo formado por 2 redes neuronales una especializada en el acercamiento rápido al objetivo y la otra en la evasión de obstáculos resolvería eficientemente el problema que se plantea en la ilustración siguiente.

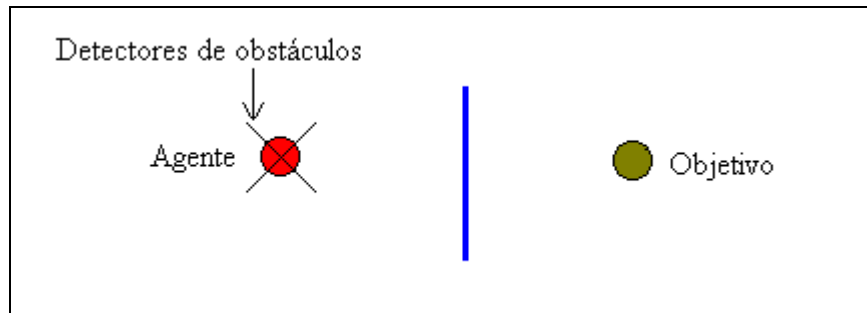


Ilustración 117 La trayectoria más corta entre el agente y el objetivo se encuentra interrumpida por la presencia de un obstáculo.

Aparentemente bastaría con elegir la primera red para procesar la entrada mientras no se detecte ningún obstáculo, en caso contrario debería seleccionarse la segunda red. Esta estrategia ha sido frecuentemente hallada por las primeras implementaciones de evolución de grupos neuronales con un comportamiento generalmente indeseable.

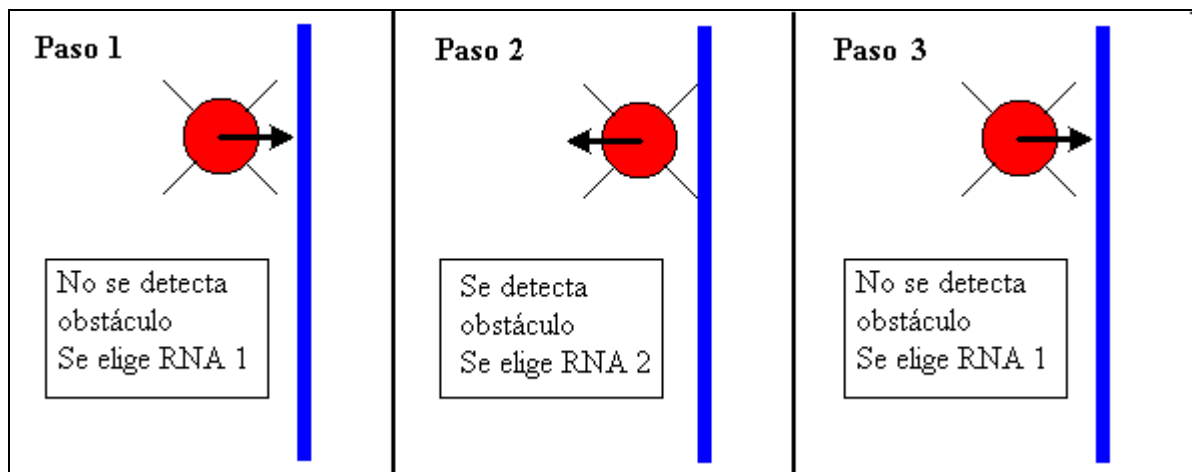


Ilustración 118 Comportamiento cíclico indeseable producto del intercambio repetitivo de control.

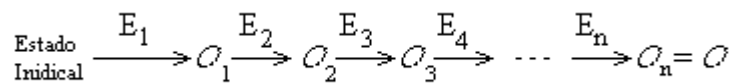
El método GNE finalmente implementado ha surgido como una respuesta satisfactoria a las dos situaciones mencionadas.

¹ Ver “Desarrollo de los Algoritmos Genéticos” en la pág. 75.

Respecto a la primera, se ha cambiado la evolución simultánea de las redes especializadas de los grupos neuronales por una evolución secuencial e incremental dividida en etapas. En cada una de estas etapas una nueva red ingresará al grupo y sólo ella será sometida a evolución, para conformar junto a las antiguas integrantes un nuevo grupo que será capaz de resolver una tarea más compleja que la resuelta por la conformación antigua del grupo.

Respecto a la segunda, se ha evitado el comportamiento cíclico indeseable imponiendo restricciones temporales al intercambio repetitivo del control entre las redes del grupo. Más adelante se darán precisiones.

Más formalmente si O es el objetivo final que se desea alcanzar, se debe hallar una serie de objetivos más simples de complejidad creciente $O_1, O_2, \dots, O_n = O$. Llamaremos E_i a la etapa del proceso evolutivo donde la población de grupos neuronales evoluciona hacia el objetivo O_i .



La etapa E_i no sólo se caracteriza por corresponder a la fase del proceso en que la población evoluciona hacia el objetivo O_i sino también por la conformación de los grupos con exactamente i redes neuronales. Así la evolución comenzará con una población de grupos unitarios.

Cada etapa estará conformada por un número de generaciones que podrá ser fijo o variable. Es decir, la etapa E_i podrá concluir una vez alcanzado el objetivo O_i o bien luego de un número predefinido de generaciones. En el software que acompaña el presente trabajo este parámetro debe ser definido antes de comenzar un proceso evolutivo.

Si bien se evoluciona una población de grupos neuronales, éstos nunca serán codificados completamente en los cromosomas. En todo momento el algoritmo genético trabajará como si se tratasen sólo de redes neuronales individuales. La diferencia sustancial subyace en la fase de evaluación. Al decodificar el fenotipo, la red resultante no se utiliza individualmente, sino por el contrario, es insertada en un grupo de redes previamente evolucionado para colaborar con el resto de los integrantes en la realización de alguna tarea. De esta manera, el fenotipo no es una red neuronal sino un grupo de ellas codificado parcialmente.

Veamos como se implementa la etapa genérica E_i . Se utiliza SANE extendido –ya comentado en el apartado “Extensión al método SANE implementada” en la pág. 211– coevolucionando de la manera conocida una población de blueprints y una de unidades. En la fase de evaluación, por cada cromosoma blueprint se construye un grupo neuronal por la incorporación de la red codificada en este cromosoma al mejor grupo salvado en la etapa E_{i-1} –o al grupo neuronal vacío si $i=1$ –. Durante la evaluación, los datos de entrada se presentan a la nueva integrante quien decide responder por sí misma o delegar la tarea a la conformación antigua del grupo. El blueprint recibirá el fitness de acuerdo al rendimiento del grupo neuronal en función del objetivo O_i . La asignación del fitness a las unidades se realiza de la forma conocida. Una vez concluida la etapa E_i se salva el mejor grupo conseguido en ella¹. Se generan aleatoriamente una nueva población de blueprints y una nueva población de unidades comenzando así la etapa E_{i+1} .

¹ Obsérvese que no se limita a seleccionar el mejor grupo de la última generación ya que no es siempre en ésta donde se manifiesta el mejor desempeño.

Obsérvese que el método es similar a la codificación Delta-coding de Whitley que se presentó al exponer ESP. Sólo que aquí las nuevas poblaciones que corresponderían a los Δ -cromosomas se generan de manera totalmente aleatoria sin restricciones y en lugar de representar diferencias numéricas respecto del mejor cromosoma salvado, representan un nuevo integrante para conformar un nuevo grupo a partir del mejor grupo neuronal salvado.

Veamos ahora en detalle cómo se autoorganiza un grupo neuronal para dar respuesta a una entrada de datos. Al igual que en el artículo de Bruce y Miikkulainen comentado, se utiliza una neurona de salida extra en las redes neuronales para determinar el integrante del grupo que responderá al estímulo. Pero en este caso, la salida añadida no representa el nivel de confianza de la red sino más bien una decisión sobre su incumbencia para brindar una respuesta a los datos de entrada presentados. Así, por medio de ella, la red opta por responder o delegar esta tarea al resto del grupo.

Obsérvese que de esta manera, en la etapa E_i del proceso evolutivo, la red sometida a evolución que se agrega a un grupo que cumple con los requerimientos necesarios para alcanzar el objetivo O_{i-1} , debe aprender a delegar la funcionalidad necesaria para alcanzar este objetivo y a resolver por sí misma las circunstancias para superarlo hasta alcanzar O_i .

Aquí no se hace necesario la evaluación de todas las RNA del grupo, sino que se van evaluando secuencialmente hasta que alguna se hace cargo de la respuesta. Pero el orden de evaluación no es arbitrario, por el contrario está completamente definido por la historia del proceso evolutivo. Se utiliza el orden inverso en que las redes se incorporaron al grupo

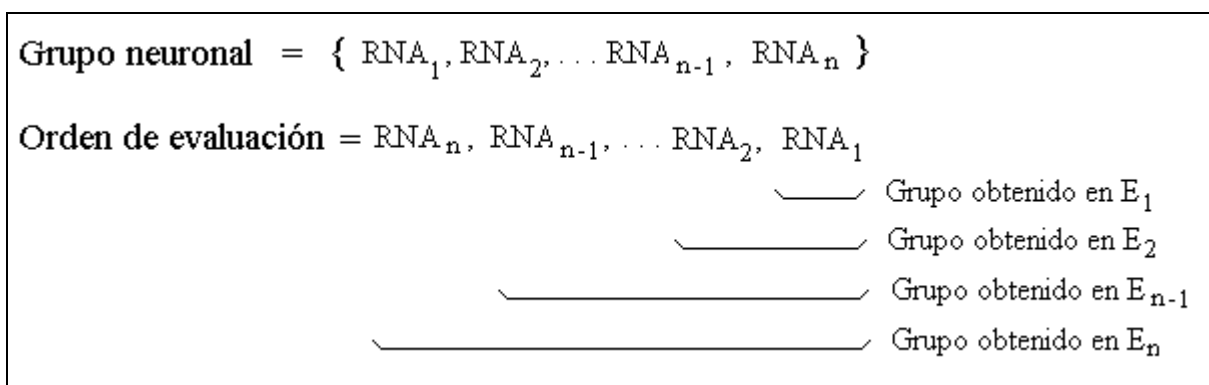


Ilustración 119 El orden de evaluación de las redes neuronales de un grupo es el inverso al orden en que dichas redes fueron incorporadas al mismo.

La red neuronal RNA₁ correspondiente a la primera etapa evolutiva es la única que no necesita la neurona de salida extra. En caso de ser seleccionada para su evaluación debe responder siempre pues significa que todas las demás han delegado esta tarea.

La neurona extra de incumbencia, podría tener una salida binaria, o bien una salida real junto con un umbral. He elegido esta segunda opción –con valores en el intervalo [0..1]– porque resulta más flexible permitiendo implementar múltiples esquemas para la delegación de la respuesta, simplemente agregando más de un umbral con distintos significados. Como ejemplo citaré una estrategia para evitar el comportamiento cíclico producido por el intercambio repetitivo de control que ya se ha comentado.

He mencionado que imponiendo restricciones temporales a las transiciones del control entre las redes neuronales del grupo podemos evitar este indeseable efecto de movimientos cíclicos. Por ejemplo, podemos hacer que cada vez que una red neuronal acepta responder a una entrada deba responder también las siguiente k entradas –siendo k un número definido previamente–. Pero el intercambio rápido de control no siempre produce movimientos

indeseables. Puede implementarse una estrategia más flexible utilizando más de un umbral, como en el siguiente ejemplo:

- (1) Si $s(v) > 0.5 \rightarrow$ “dar respuesta a la entrada de datos”
- (2) Si $0.2 < s(v) \leq 0.5 \rightarrow$ “delegar la respuesta al resto del grupo”
- (3) Si $0.1 < s(v) \leq 0.2 \rightarrow$ “delegar esta y las siguientes 20 respuestas al resto del grupo”
- (4) Si $s(v) \leq 0.1 \rightarrow$ “delegar esta y todas las siguientes respuestas al resto del grupo”

Donde v es la entrada de datos y $s(v)$ el valor de la neurona de incumbencia para dicha entrada. Así una red neuronal podría decidir no dar respuesta a la entrada corriente pero seguir participando activamente –caso (2)– o en el otro extremo, decidir excluirse completamente de cualquier decisión a partir de ese momento –caso (4)–. Más adelante se comentará la utilidad de una estrategia como esta.

Finalizando este punto debo comentar que he implementado GNE como una estrategia de neuroevolución incremental con buenos resultados, mejorando con ella no sólo el fitness sino también la velocidad de evolución en varios casos –ver a partir de la Ilustración 150, pág. 268 en el capítulo “Evaluación y Análisis”–. No obstante, existen otros beneficios potenciales que seguramente se manifiesten en problemas con entornos dinámicos que cambian con el tiempo.

A modo de ejemplo considérese la siguiente situación. Supóngase un problema ya resuelto con la utilización de alguna red neuronal. Esta red pudo haber sido conseguida por neuroevolución o entrenada con cualquier algoritmo de aprendizaje tradicional. Supóngase ahora que un cambio repentino en las condiciones del ambiente ocasiona el funcionamiento defectuoso de la red neuronal que hasta el momento se desempeñaba satisfactoriamente. Ante esta situación, lo habitual es correr un nuevo entrenamiento o proceso evolutivo desde cero para conseguir una nueva red neuronal que lleve a cabo la tarea encomendada bajo las nuevas condiciones del entorno. Es decir, se trata a la variante del problema como uno totalmente distinto al anterior.

La hipótesis a verificar es que, ante este cambio repentino en las condiciones del ambiente, GNE se desempeñará con mayor eficiencia que cualquier alternativa que implique atacar el cambio como si fuese un nuevo problema. Tales alternativas desaprovechan lo que aún la RNA a ser reemplazada puede resolver con eficiencia. GNE, por el contrario, utilizará esta red junto a una nueva sometida a evolución para conformar un grupo que idealmente aprenderá a autoorganizarse haciendo que la nueva red se haga cargo de las implicancias del cambio, mientras que la antigua siga respondiendo los aspectos del entorno que permanecen estáticos.

Claro que una argumentación similar puede hacerse a favor del método de neuroevolución incremental propuesto por Gómez y Miikkulainen [*Gomez97*] que se comentó al exponer ESP. Pero, aunque deseable, no siempre el espacio de representación es coherente con el espacio de soluciones¹ lo que degrada la eficiencia de Delta-coding. Recordemos que esta estrategia busca optimizar la mejor solución hallada hasta el momento en las proximidades de su genotipo. GNE, en cambio, no prestará atención a la codificación genética de la RNA

¹ Recordemos que al elegir la codificación genética en cualquier algoritmo evolucionario es conveniente que exista una relación coherente entre el espacio de representaciones y el de soluciones. Es decir se pretende que representaciones genéticas parecidas produzcan fenotipos también parecidos. Claro que esto no es siempre fácil de conseguir. Los códigos de Gray para representar números enteros se utilizan precisamente para conseguir que una variación pequeña en la codificación genética implique una variación también pequeña en el fenotipo. La codificación binaria no cumple con esta condición deseable. Estos conceptos ya han sido mencionados en el capítulo titulado “Computación Evolutiva” –pág. 43–.

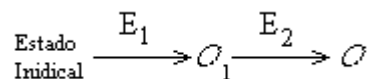
obsoleta por el cambio en el entorno –incluso podría no estar disponible–, sino que intentará hallar la mejor combinación funcional de ella con una nueva evolucionada.

Dejaré planteada la comprobación empírica de esta hipótesis para un trabajo futuro. No obstante, los argumentos a su favor son convincentes. Un cambio en las condiciones del ambiente rara vez transformará al problema en uno diametralmente opuesto al original. Como ejemplo considérese el desgaste o rotura de una pieza en un problema de control o la incorporación de nuevas clases de patrones en un sistema de clasificación. Las “reglas de juego” no cambiarán radicalmente sino que agregarán una complejidad extra para alcanzar la nueva meta. Esto encaja en los supuestos de una evolución incremental. Así una nueva etapa evolutiva de GNE podrá resolverlo más eficientemente.

Detalles de Implementación

Hasta aquí se ha descrito el método de neuroevolución propuesto de manera genérica. Ahora comentaré algunos detalles de la implementación en relación al dominio de problemas planteado en el presente trabajo.

He dividido el proceso evolutivo en dos etapas, así se evolucionarán grupos de dos redes neuronales, esto ha sido suficiente para demostrar que el método supera en varios casos las otras estrategias implementadas.



La duración de cada etapa, expresada en pasos de simulación, será fija, determinada antes de comenzar el proceso evolutivo.

Como ya se ha explicado, el objetivo O de los problemas aquí planteados para cada escenario en particular es controlar los agentes para que partiendo desde un origen determinado alcancen su destino, en la menor cantidad de pasos de simulación posible. Siendo origen el punto de partida definido en el campo de prueba, y destino algún lugar geográfico fijo o el ocupado en ese instante por alguna presa en movimiento.

Una forma de conseguir un subobjetivo O_1 es simplemente reduciendo la complejidad de la travesía de los agentes hacia su destino. Existen muchas maneras de lograrlo, aquí se ha elegido una de las más simples, acortando la longitud del recorrido que deban realizar. Así el subobjetivo O_1 será controlar los agentes para que partiendo desde algún lugar intermedio del camino entre origen y destino, alcancen este último en el menor tiempo posible.

Para no incorporar un elemento más a los escenarios, se aprovecharán las submetas, definidas para el cálculo del fitness, para determinar el punto de partida de los agentes en la etapa 1. En el software que acompaña el presente trabajo, debe indicarse cuál de ambas se utilizará con este propósito, antes de comenzar un proceso evolutivo GNE.

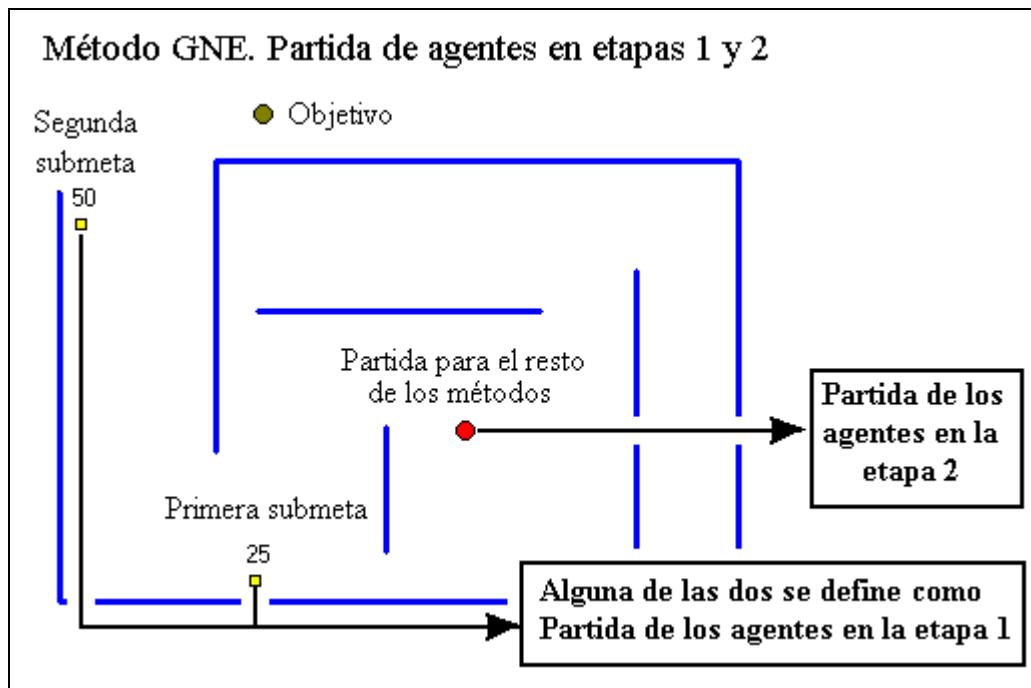


Ilustración 120 La Partida de los agentes en la etapa 1 del método GNE se realiza desde alguna de las submetas del escenario.

Así se espera que en la segunda etapa los grupos neuronales sean capaces de autoorganizarse para que la RNA₂ conduzca al agente a las cercanías del punto de partida de la etapa anterior, desde donde podría tomar el control la RNA₁ especializada en el recorrido desde allí hasta el destino final. Debe quedar claro que esta organización en la delegación del control dentro del grupo es hallada por evolución no existiendo ninguna presión del entorno para que así ocurra. Tampoco los agentes poseen información alguna sobre el sitio desde donde partieron en la etapa 1. La organización en la delegación de control queda determinada en función del valor de la tercera neurona de salida de RNA₂ que depende sólo de los valores de entrada a la red neuronal.

Incluso, en las pruebas realizadas en ocasiones los grupos han hallado otras formas de organizarse con mejores resultados. Por ejemplo, varias veces se ha dado el caso que aún no habiendo alcanzado el destino en la etapa 1 –RNA₁ de baja performance–, éste ha sido fácilmente alcanzado en la etapa 2 utilizando varios intercambios de control –ver Ilustración 156 en la pág. 271–. Una organización como la descrita en el párrafo anterior, en este caso, posiblemente haría al destino inalcanzable.

Para impedir los movimientos cíclicos producidos por el cambio reiterado de control entre las redes neuronales se ha impuesto una restricción temporal fuerte. Debe destacarse que esto es una decisión en función del dominio de problemas particular que estamos resolviendo y no debe considerarse como parte del método. Ya se ha mencionado que pueden existir distintas estrategias para restringir el cambio de control, incluso ha quedado pendiente explicar cuál sería la utilidad de que una red neuronal de un grupo, en algún momento decidiera delegar la tarea de responder todas las entradas de allí en adelante. Precisamente este es un ejemplo en el que tal estrategia es útil.

En la presente implementación del método GNE, se ha impuesto la restricción siguiente: una vez que la segunda red neuronal –incorporada en la etapa 2– decide delegar la tarea a la primera red, esta delegación será hasta que haya alguna colisión o finalice la simulación. Si hay alguna colisión, la segunda red vuelve a tomar intervención en la decisión de control, lo

que no implica que responderá a la entrada de datos comandando el movimiento, sino que nuevamente optará por alguna de las dos opciones, responder por sí misma o transferir la tarea a la otra red.

Como ya se ha mencionado, esta restricción está totalmente influenciada por el tipo de problema que estamos resolviendo, ya que hay una gran probabilidad de que el agente llegue a destino si se delega el control en las cercanías de la partida en la primera etapa.

Herramienta de software desarrollada. ENE.

Introducción

ENE o Entorno Neuroevolucionario es un programa desarrollado con Delphi 5.0 que forma parte del presente trabajo de grado y ha sido utilizado como la herramienta fundamental que ha permitido analizar y evaluar el rendimiento de distintas arquitecturas neuronales y de las 6 estrategias neuroevolutivas implementadas.

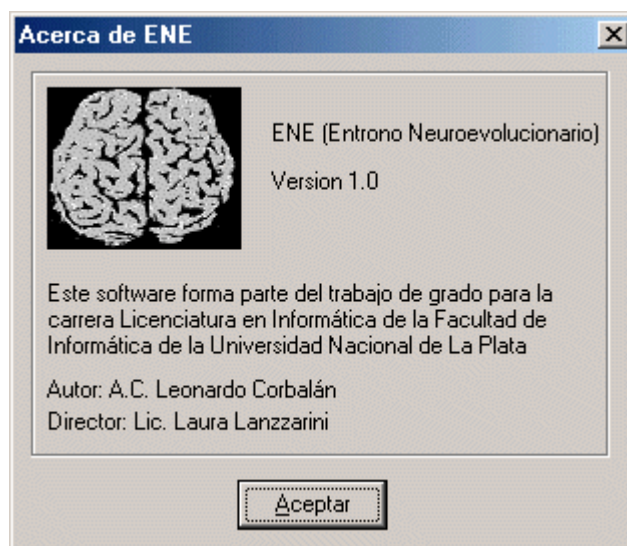


Ilustración 121 Ventana “Acerca de ENE”

Este brevísimo capítulo no pretende ser un manual de usuario sino una exposición de las características implementadas con algunas nociones del procedimiento necesario para llevar a cabo ciertas funciones del programa. Sin más preámbulo comencemos con la descripción.

Descripción del programa

A “grosso modo” se puede dividir en tres grupos las acciones que pueden realizarse con este software:

- Llevar a cabo los ensayos neuroevolutivos, salvando datos de performance relevantes y mejores controladores neuronales¹ generados. En cualquier instante de la simulación podrá inspeccionarse el estado interno de estos controladores –topología de las redes, valores de las neuronas y conexiones–.
- Analizar estadísticamente en forma de gráficos el desempeño de las evoluciones.
- Analizar visualmente en simulaciones el accionar de los controladores neuronales salvados en cada ensayo neuroevolutivo, pudiendo, en cualquier momento, inspeccionar el estado interno de estos controladores.

¹ Por “controladores neuronales” debe entenderse “grupos neuronales” o bien “redes neuronales” según se utilice o no GNE respectivamente.

Primero daré precisiones del concepto de “ensayo neuroevolutivo”. En este contexto, se denominará así a la ejecución de n evoluciones para una determinada configuración de parámetros elegida que caracteriza un caso de prueba que se desea analizar. Un ensayo nos permitirá la recopilación de datos necesaria para el análisis comparativo con los obtenidos por medio de otros ensayos o entre las distintas evoluciones que lo componen.

La repetición de las evoluciones con la misma configuración de parámetros se debe a la necesidad de promediar valores, y esta necesidad surge de la sensibilidad a las particularidades de las poblaciones iniciales de los algoritmos evolucionarios, que como se recordará, se generan aleatoriamente.

De los tres puntos mencionados arriba, el primero de ellos es quien merece mayor explicación.

Ensayos neuroevolutivos

Para comenzar un ensayo neuroevolutivo son varios los parámetros que deben fijarse. Así la selección de estos valores determinará el caso de prueba que nos interesa evaluar. El programa salva siempre la última configuración elegida.

Se han organizado los parámetros iniciales en grupos como puede verse en la siguiente ilustración

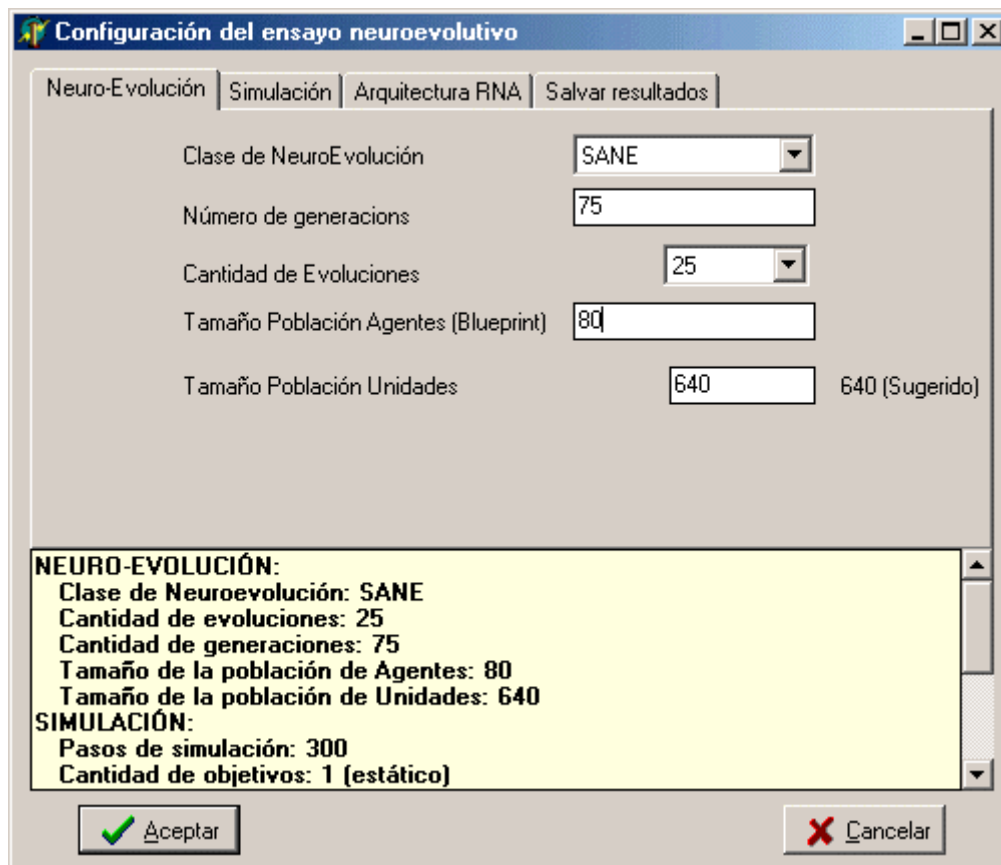


Ilustración 122 Ventana para establecer la configuración inicial de un nuevo ensayo neuroevolutivo. A medida que se van fijando los distintos parámetros, un resumen de la configuración elegida se va actualizando en el panel inferior. Este resumen es salvado junto a otros datos del ensayo para contar con una descripción compacta al momento de analizarlos.

Esta ventana se abrirá cada vez que decidamos realizar un nuevo ensayo neuroevolutivo. Esto se realiza a través del menú principal de la aplicación.

Como puede observarse en la ilustración anterior, en la ficha “Neuro-Evolución” debe elegirse la estrategia que se utilizará, la cantidad de generaciones que comprende una evolución, la cantidad de evoluciones que conforman el ensayo neuroevolutivo, el tamaño de la población de agentes y unidades si las estrategias son SANE, ESP, ESP con población de Blueprints o GNE, pero sólo de agentes si éstas son NE Simple o NE Simple Elitista. Además, en caso de elegir alguna de estas dos últimas debe especificarse la probabilidad de aplicación de los operadores de crossover y mutación –recordemos que en las otras estrategias estas probabilidades han quedado fijadas por el método–. Si la estrategia elegida es GNE deben especificarse también la cantidad de generaciones, pasos de simulación y punto de partida de los agentes para la primera etapa neuroevolutiva de este método.

Otros parámetros que deben fijarse antes de comenzar un ensayo son aquellos concernientes a la simulación. Así será necesario establecer los pasos de la simulación, el tamaño de los agentes y objetivos, el escenario sobre la que se desarrollará, y la semilla inicial del proceso de generación de números aleatorios. Si el valor de este último se deja en cero, el programa utilizará la instrucción “randomize” de Delphi que inicia este valor de acuerdo a cálculos internos sin que el usuario tenga forma de preverlo y así se trabajará bajo condiciones totalmente aleatorias.

Para definir el escenario sobre el que correrá la simulación, se ha implementado un precario, aunque útil, editor de escenarios.

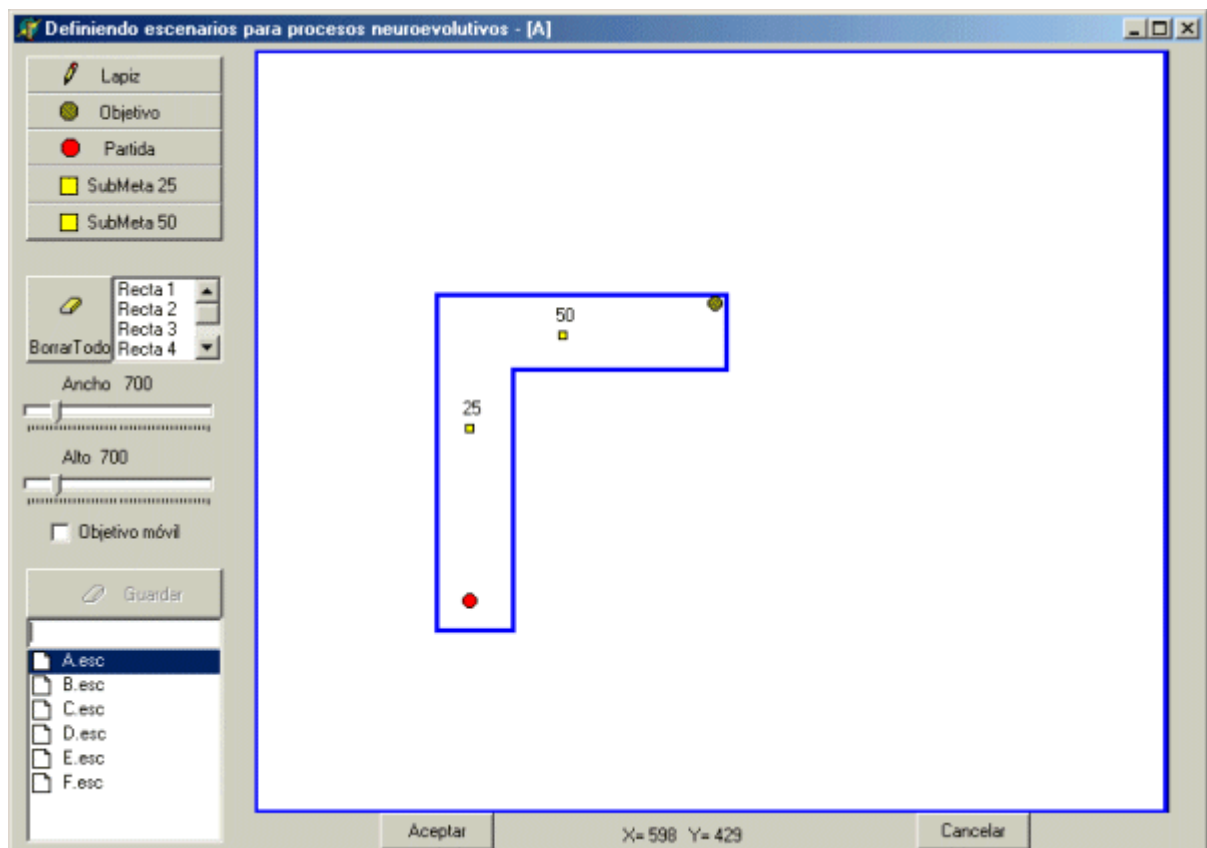


Ilustración 123 Editor de escenarios.

Con este sencillo editor se puede definir la posición de cada elemento del escenario. Es posible agregar o quitar tantos obstáculos como se requieran –rectas azules– y tampoco existe

límite para la cantidad de objetivos. Además se pueden alterar las dimensiones –ancho y alto– del campo de prueba, elegir objetivos móviles o estáticos y salvar o cargar un escenario del disco rígido.

En la ficha “Arquitectura RNA” de la ventana mostrada en la Ilustración 122 se agrupan los parámetros relativos a la arquitectura neuronal –para detalles al respecto de ésta vea “Tipos de Redes Neuronales Artificiales implementadas” en la pág. 184–. Así debe elegirse el tipo de red –feedforward o recurrente, multisigmoide o con sigmoide fija, con conexión de tendencia o sin ella–, la cantidad de neuronas de entrada –desde 10 en adelante–, la cantidad de neuronas ocultas, la cantidad de conexiones que se codificarán genéticamente por cada neurona, y la precisión en bits –locus en el cromosoma– para los pesos de las conexiones.

Finalmente, si no se especifica un nombre de ensayo, todos los datos relativos a él se guardarán bajo el nombre genérico “_default” sobrescribiendo los datos que pudieran existir de cualquier otro ensayo anterior al que no se la haya dado un nombre adecuado.

En la Ilustración 122 puede observarse a la derecha del parámetro “Tamaño Población Unidades” una leyenda sugiriendo un valor adecuado. Este valor que se actualiza automáticamente según el usuario va modificando otros, no tiene por qué ser respetado pero puede servir de referencia si tenemos presente cómo se calcula. Ya se ha comentado en el capítulo anterior que existen ciertos parámetros de los que muy poco se dice en los artículos que introdujeron o utilizan los métodos SANE y ESP [Moriarty96] [Moriarty97_1] [Moriarty97_2] [Gomez97] [HanYong01] [PerezBergquist01]. Uno de esos parámetros es el tamaño de la población de unidades, otro es el número de conexiones a codificar por cada neurona en esta población. Todos los métodos aquí implementados utilizan este último parámetro, pues como se recordará la codificación genética utilizada incluso en las estrategias de neuroevolución simple están inspiradas en SANE. En cuanto al parámetro tamaño de población de unidades es utilizado por SANE, ESP, ESP con población de Blueprints y GNE.

Observando la forma en que construyeron los casos de pruebas, y basándome en algunas justificaciones poco formales de estos investigadores –algo he comentado bajo los títulos “Debilidad del método” en la pág. 220 y “ESP con población de blueprints” en la pág. 221– el software aquí desarrollado sugiere para el tamaño de la población de unidades en SANE, ESP con población de Blueprints y en GNE que sea igual a la demanda de neuronas que se producirá al intentar construir las redes neuronales que hagan falta –tamaño de la población de agentes–. En cambio para ESP se sugiere que sea 4 veces menor que la población de agentes –debido a los requerimientos ya explicados de este método en “Debilidad del método”–.

Por otro lado, la cantidad sugerida de conexiones a codificar por unidad se calcula de igual forma para todas las estrategias. Sobre el valor adecuado de este parámetro no he encontrado ninguna mención en los artículos citados. A través de la experimentación y con el criterio que sea factible la generación de redes totalmente conectadas se sugiere aquí tantas conexiones como sea el número total de neuronas con que la unidad pueda conectarse, esto es la cantidad de neuronas de entrada más la cantidad de neuronas de salida si la red es feedforward, o la cantidad total de neuronas de la red si ésta es recurrente.

Una vez configurado adecuadamente el conjunto de parámetros iniciales, al aceptarlos comienza el ensayo neuroevolutivo.

Como observadores, el ensayo se nos presenta casi exclusivamente como una serie de simulaciones que se corresponden a cada generación evaluada. La razón es que la etapa de selección reproducción y reemplazo en las poblaciones de cualquier estrategia neuroevolutiva

son muchísimo más rápidas que la etapa de evaluación. Así el retardo entre la simulación de una generación y la próxima generalmente es imperceptible a nuestros sentidos.

Se puede optar por no visualizar la animación, en tal caso, la simulación será mucho más rápida y tendremos oportunidad de ver como generación tras generación se va dibujando la curva del fitness con el valor más alto conseguido por el mejor agente de cada una de ellas.

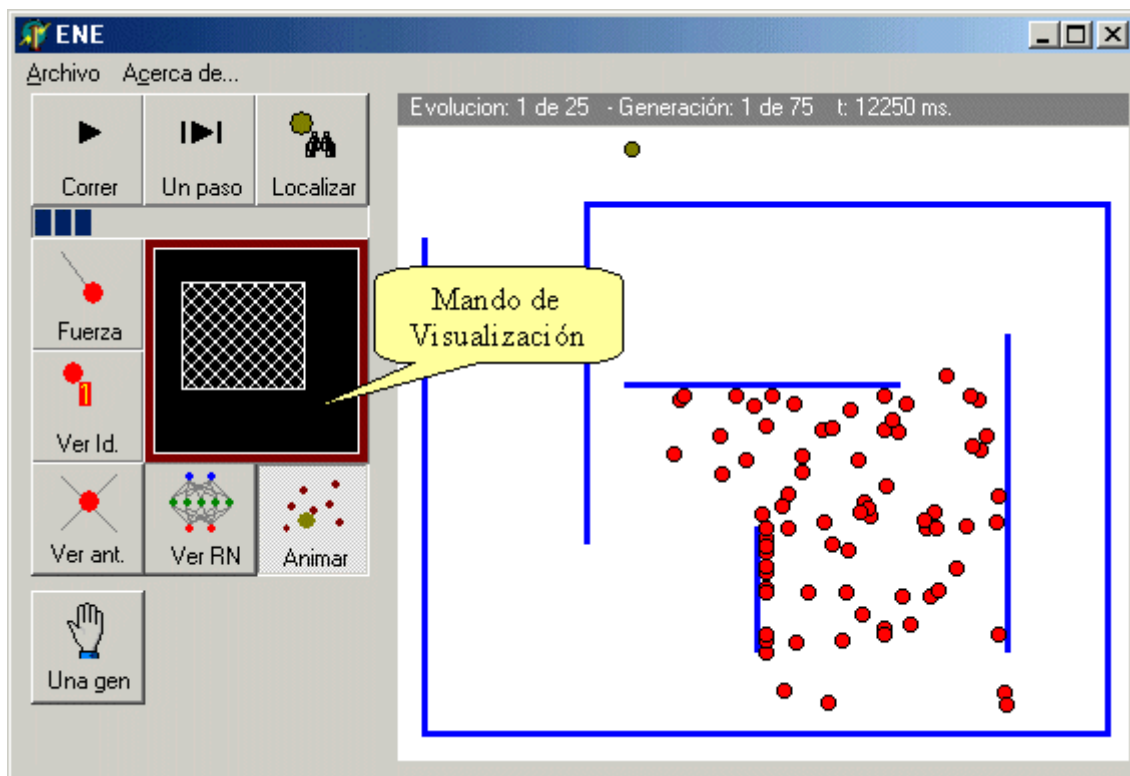


Ilustración 124 Ventana principal de ENE visualizando una simulación.

Si bien el panel destinado a la visualización de la simulación se ajusta al tamaño de la ventana principal y por lo tanto será mayor si ésta se maximiza, muchas veces trabajaremos con entornos más grandes que el panel de visualización. Entonces, como en la ilustración anterior, sólo veremos una porción del campo de prueba. Para decidir qué zona visualizar se utiliza el mando de visualización. La zona de color negro corresponde al campo de prueba, mientras que la cuadrículada representa la región que se está visualizando. Simplemente, arrastrando esta cuadrícula a otras posiciones con el ratón, movemos el campo de visión para poder observar cualquier lugar del campo de prueba.

Mientras se está corriendo un ensayo neuroevolutivo, el programa permite ejecutar paso a paso la simulación –botón “Un paso”– o reanudar la ejecución continua –botón “Correr”–. También se puede localizar los objetivos uno a uno utilizando el botón “Localizar” quien acomodará el mando de visión para que el objetivo buscado quede posicionado en el centro del panel de visualización.

En cualquier momento puede activarse o desactivarse la visualización de las fuerzas que están realizando los agentes –botón “Fuerza”–, de las etiquetas con su número de identificación – botón “Ver Id.”– y de las antenas detectoras de obstáculos –botón “Ver ant.”–.

Presionando el botón “Una gen” la simulación se coloca automáticamente en estado “Paso a paso” una vez realizado el último paso de simulación. Esto es útil porque nos da la posibilidad de detener la simulación para inspeccionar las redes neuronales justo antes de disparar el paso

evolutivo que realiza la selección reproducción y reemplazo consiguiendo la próxima generación.

El botón “Animar” es un control que posee dos estados. Si se encuentra presionado se podrá visualizar la animación, tal como lo muestra la Ilustración 124. Si por el contrario no se encuentra presionado, se oculta la animación y en su lugar se observa un gráfico donde se va construyendo la curva de fitness para la evolución corriente. Esto es sumamente útil pues acelera muchísimo la simulación al quitar la carga de procesamiento necesaria para visualizar la animación.

Finalmente, una de las funcionalidades más importantes descansa en el accionar del botón “Ver RN”. Presionándolo se abrirá una ventana para inspeccionar en línea los controladores neuronales .

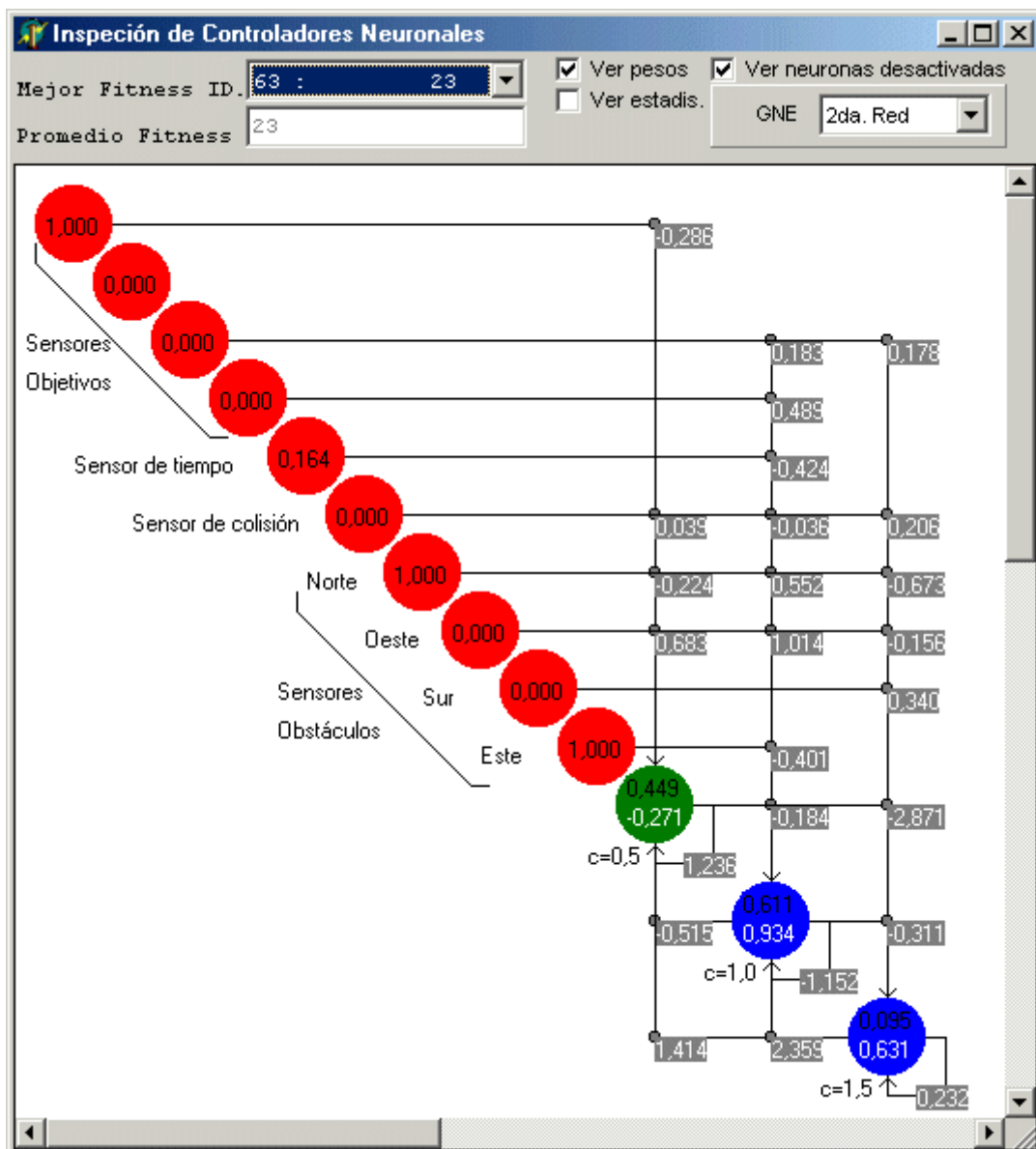


Ilustración 125 Ventana “Inspección de Controladores Neuronales”. En ella puede observarse la gráfica de la segunda red del grupo neuronal Id=63 –estrategia GNE–. En este caso se trata de una red recurrente, con término de tendencia multisigmoide, con 10 neuronas de entrada –rojas– y sólo una neurona oculta –verde–. Obsérvese la particularidad de esta red que posee una neurona de entrada desconectada.

Esta ventana se abre de manera no modal, pero permanecerá en primer plano hasta que se cierre. Así es posible acomodarla de tal modo que se pueda ver simultáneamente cómo avanza la simulación en la ventana principal en segundo plano al mismo tiempo que se observan cómo cambian los valores de cada una de las neuronas de la red. También suele ser útil identificar un agente en la simulación, abrir esta ventana de inspección, visualizar el controlador neuronal de dicho agente –seleccionándolo desde el cuadro combinado superior– y analizar paso a paso cómo se comporta la red estimulada presionando el botón “Un paso” de la ventana principal en segundo plano.

Veamos brevemente cómo se debe interpretar el esquema de las redes. Como ejemplo utilizamos la red neuronal recurrente, con tendencia y multisigmoide de la Ilustración 126.

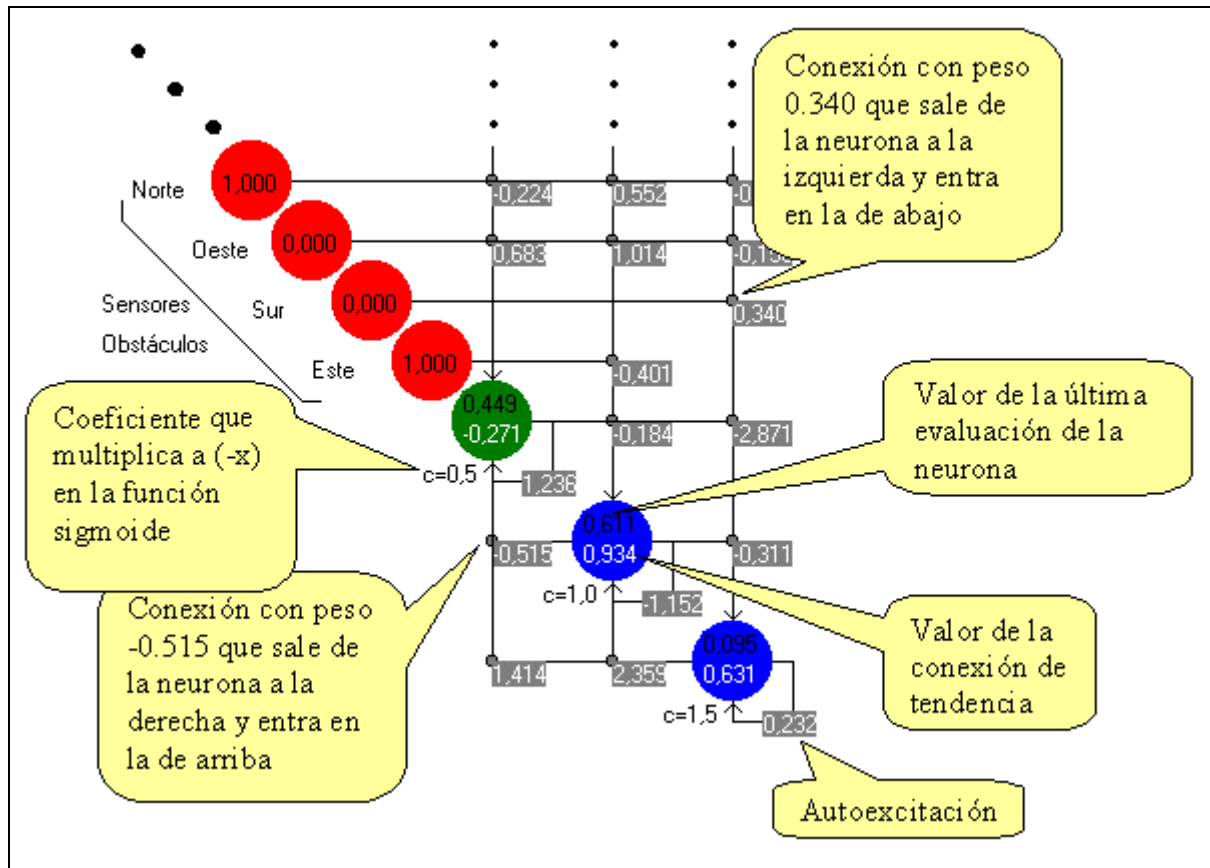


Ilustración 126 Interpretación del esquema de una red neuronal.

Las neuronas se disponen en una recta diagonal. Primero aparecerán las neuronas de entrada –rojas–, luego las ocultas –verdes– y finalmente las de salida –azules–. Las conexiones graficadas por encima de la diagonal, son conexiones hacia delante. Por el contrario, las que aparecen por debajo de las neuronas, son conexiones recurrentes, hacia atrás. Dentro del círculo que representan las neuronas se visualiza el valor de la misma que comunica por sus conexiones de salida. Si la red posee conexión de tendencia, el valor de su peso también aparece dentro de la neurona. Finalmente, en caso de ser una red multisigmoide, el coeficiente c de la función de transferencia se grafica abajo a la izquierda de cada neurona.

Desde esta ventana no sólo se puede inspeccionar uno a uno cada controlador neuronal de todos los agentes que están corriendo la simulación –generación corriente–, sino que también se puede observar un gráfico visualizando el fitness parcial calculado al presionar el botón “Ver RN” de todos ellos. Esto se consigue marcando la casilla de verificación “Ver estadis.”

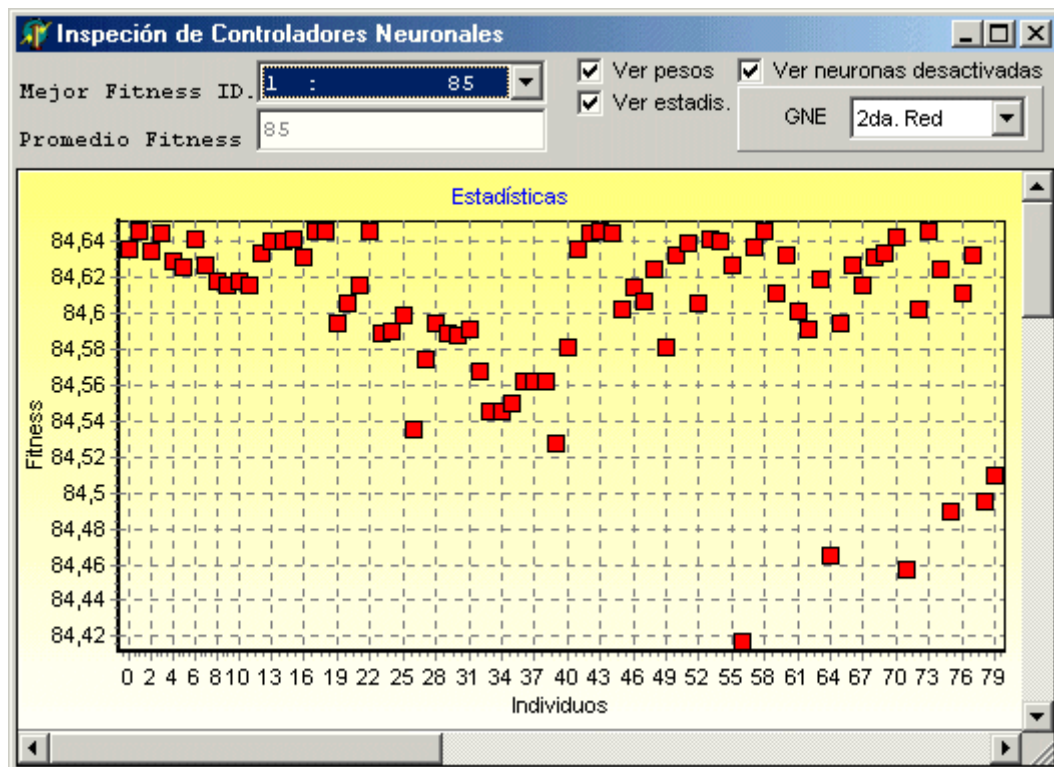


Ilustración 127 Visualización del fitness logrado por los agentes de la generación corriente calculado en el instante en que se abrió esta ventana de inspección.

Por cada generación de cada evolución de un ensayo neuroevolutivo se salvan: el controlador neuronal de mejor rendimiento de la generación, su valor de fitness, y la cantidad de pasos de simulación que realizó hasta alcanzar el objetivo, si es que lo logró, ó 0 (cero) en caso contrario. Así tendremos un interesante cúmulo de datos para realizar el análisis estadístico pretendido.

Análisis estadístico de datos

Para visualizar los datos generados en los ensayos neuroevolutivos se debe seleccionar la opción “Estadísticas” del menú “Archivo” de la aplicación. Se abrirá entonces la ventana de la Ilustración 128. Se ha utilizado multithreading, así es posible realizar el análisis de los datos aún si estamos corriendo algún ensayo neuroevolutivo sin que sea necesario detenerlo.

Con los datos salvados se puede calcular el fitness promedio y el mejor fitness para cada generación entre las n evoluciones que comprenden el ensayo. También la velocidad promedio y mejor velocidad para alcanzar el objetivo –medido en pasos de simulación– para cada generación. Asimismo, en función de estos últimos datos se puede calcular la cantidad de éxitos obtenidos –objetivo alcanzado– en las n evoluciones por cada generación. –para más detalles sobre cómo trabajamos con estos datos vea “Obtención de datos” en la pág. 251–

El análisis de los éxitos es importante porque brinda una idea de la probabilidad que se tendrá de alcanzar el objetivo en cada una de las generaciones de una evolución. Por ejemplo, si se lleva a cabo un ensayo neuroevolutivo de 30 evoluciones y hacia la generación 20 el número de éxitos obtenidos es de 15, y a partir de la generación 60 el número de éxitos se mantiene en 30, puede inferirse que, si se realiza una única evolución con la configuración de parámetros ensayada, se tendrá una probabilidad cercana a 0.5 de conseguir un controlador neuronal que resuelva en problema en sólo 20 generaciones, y si

se prolonga la evolución hasta la generación número 60, la probabilidad será altísima, muy cercana a 1.

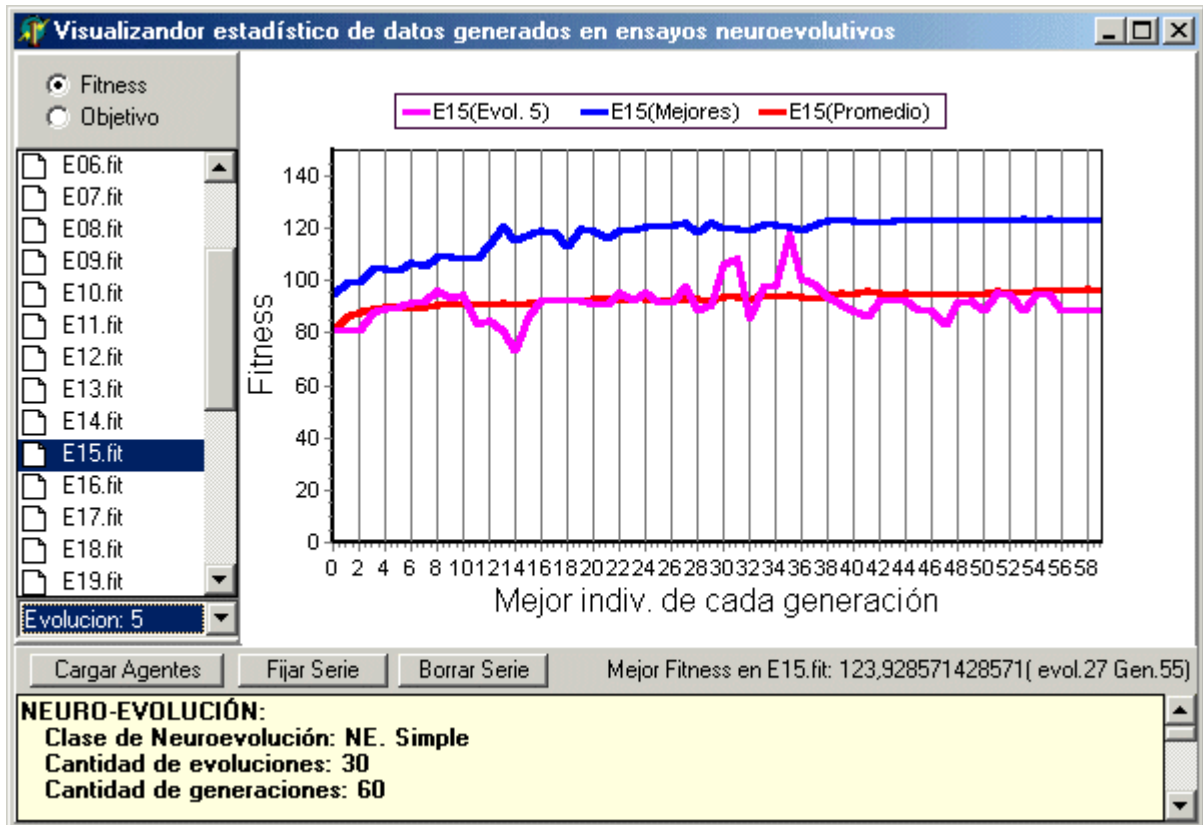


Ilustración 128 Ventana para la visualización de los datos generados en los distintos ensayos neuroevolutivos mostrando una gráfica comparativa de valores de fitness.

Como se observa en la Ilustración 128, en la lista de la izquierda pueden seleccionarse los archivos con los datos obtenidos en cada ensayo neuroevolutivo –extensión “.fit”–. En el cuadro combinado que se encuentra por debajo de esta lista, podrá seleccionarse la evolución que deseamos graficar, el promedio o los mejores fitness conseguidos por generación calculado entre todas las evoluciones que conforman el ensayo. Por ejemplo, en la ilustración mencionada puede observarse un gráfico comparativo donde se han dibujado las curvas correspondientes a la evolución número 5 del ensayo “E15”, a los mejores fitness y al promedio de fitness logrado por cada generación.

Para permitir el agregado o borrado de las series de datos a graficar deben utilizarse los botones “Fijar Serie” y “Borrar Serie”.

Pueden agregarse al gráfico la cantidad de curvas que se estimen conveniente. Claro está, que el excesivo número de series graficadas tiende a hacer difícil la interpretación de los datos.

No existe ninguna limitación en cuanto a qué serie de datos graficar. Puede combinarse la gráfica de cualquiera perteneciente a cualquier ensayo con otras pertenecientes a otros ensayos, aún si fueron generadas utilizando distinta cantidad de generaciones.

Cada vez que se selecciona algún ensayo en la lista de la izquierda, se visualiza en el cuadro de texto inferior la descripción compacta del conjunto de parámetros elegido al realizar dicho ensayo neuroevolutivo.

Obsérvese que por encima de la lista de archivos se encuentra seleccionada la opción “Fitness”, pero si se selecciona la opción “Objetivo” se tendrá la oportunidad de hacer uso de

la información guardada respecto a los pasos necesarios de simulación efectuados para alcanzar el objetivo, como lo muestra la siguiente ilustración.

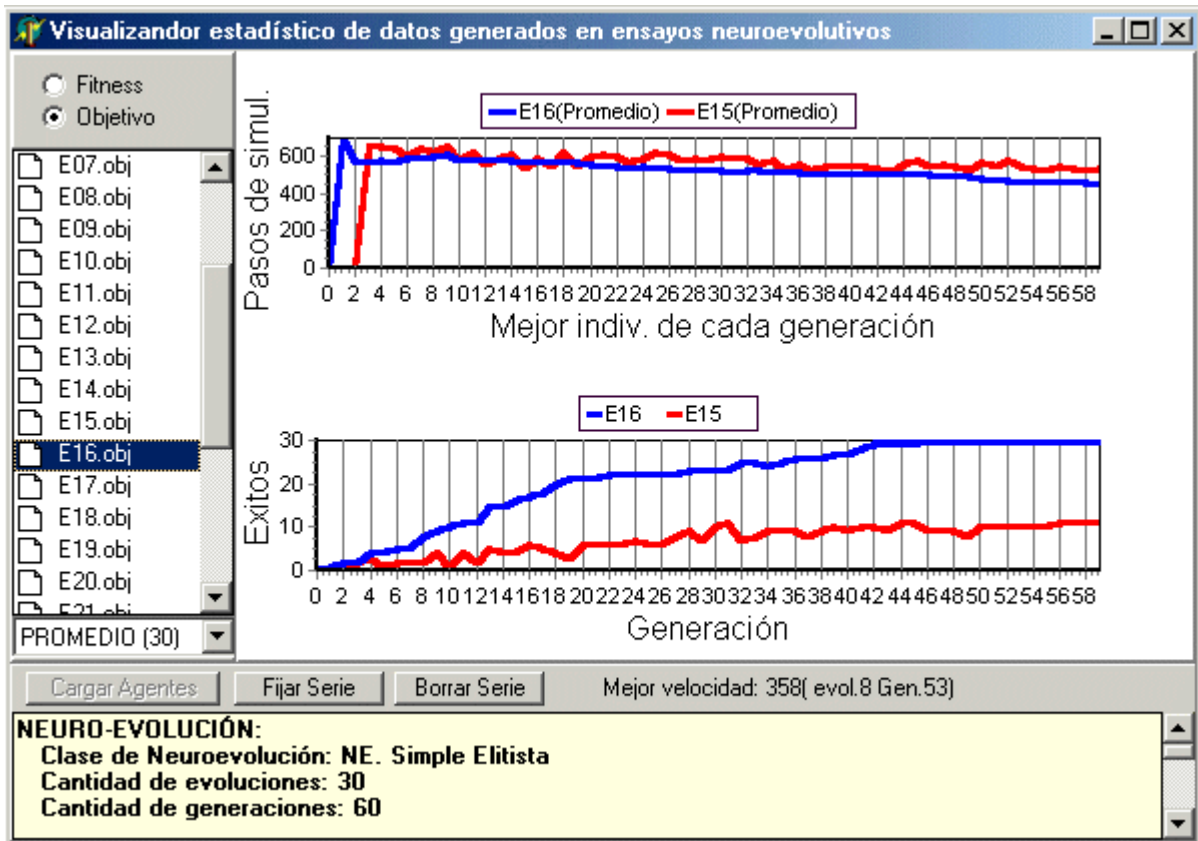


Ilustración 129 Ventana para la visualización de los datos generados en los distintos ensayos neuroevolutivos mostrando dos gráficas comparativas, una sobre la velocidad promedio medida en pasos de simulación para alcanzar el objetivo, y la otra con la cantidad de veces que el objetivo ha sido alcanzado.

Los botones “Fijar Serie” y “Borrar Serie” funcionan de igual forma que antes, sólo que ahora su efecto se aplica sobre dos gráficos en lugar de uno.

La gráfica superior muestra la cantidad de pasos necesarios para alcanzar el objetivo. En este caso la curva más baja será la de mejor rendimiento. Al igual que como ocurre con los gráficos de los fitness aquí es posible dibujar las curvas de cada evolución, de los mejores, o los promedios obtenidos sobre el conjunto de evoluciones que conforman un ensayo neuroevolutivo.

La gráfica inferior muestra la cantidad de éxitos por cada generación a lo largo de todo el ensayo neuroevolutivo.

Por ejemplo, la ilustración anterior está poniendo en evidencia la superioridad de rendimiento del ensayo “E16” respecto de “E15”. No sólo el promedio de pasos necesarios para alcanzar el objetivo –en caso de lograrlo– es inferior para todas las generaciones, sino que la cantidad de éxitos obtenidos en las 30 evoluciones que conforman dichos ensayos, es ampliamente superior.

El botón “Cargar Agentes” se activa al seleccionar alguna evolución en el cuadro combinado que se encuentra por debajo de la lista de archivos. Al oprimirlo se cerrará esta ventana, se cargarán los controladores neuronales allí guardados y comenzará la simulación en el campo de prueba correspondiente de igual manera a cuando realizamos un ensayo neuroevolutivo, con la salvedad de que, al no tratarse de una evolución, la simulación se detiene luego de

realizar los pasos establecidos. Presionando el botón “Correr” de la ventana principal la simulación se reiniciará.

Análisis de controladores neuronales salvados

Ya se ha mencionado que por cada generación de cada evolución que conforman los ensayos neuroevolutivos, se salvan los datos de rendimiento del mejor agente, pero también, y esto es lo que nos interesa en este punto, se guarda el mismísimo controlador neuronal responsable de comandar cada una de sus acciones.

Así se dispondrá de la posibilidad de cargar los controladores neuronales representativos de cada generación, para cualquier evolución de cualquier ensayo neuroevolutivo que se haya realizado. Esto puede realizarse por medio de la ventana de visualización estadística utilizando el botón “Cargar Agentes”, que ya comentamos, o bien a través del menú principal de la aplicación seleccionando la opción de menú de igual nombre, dentro de la opción “Archivos”. Si se realiza de esta segunda forma se abrirá la siguiente ventana.

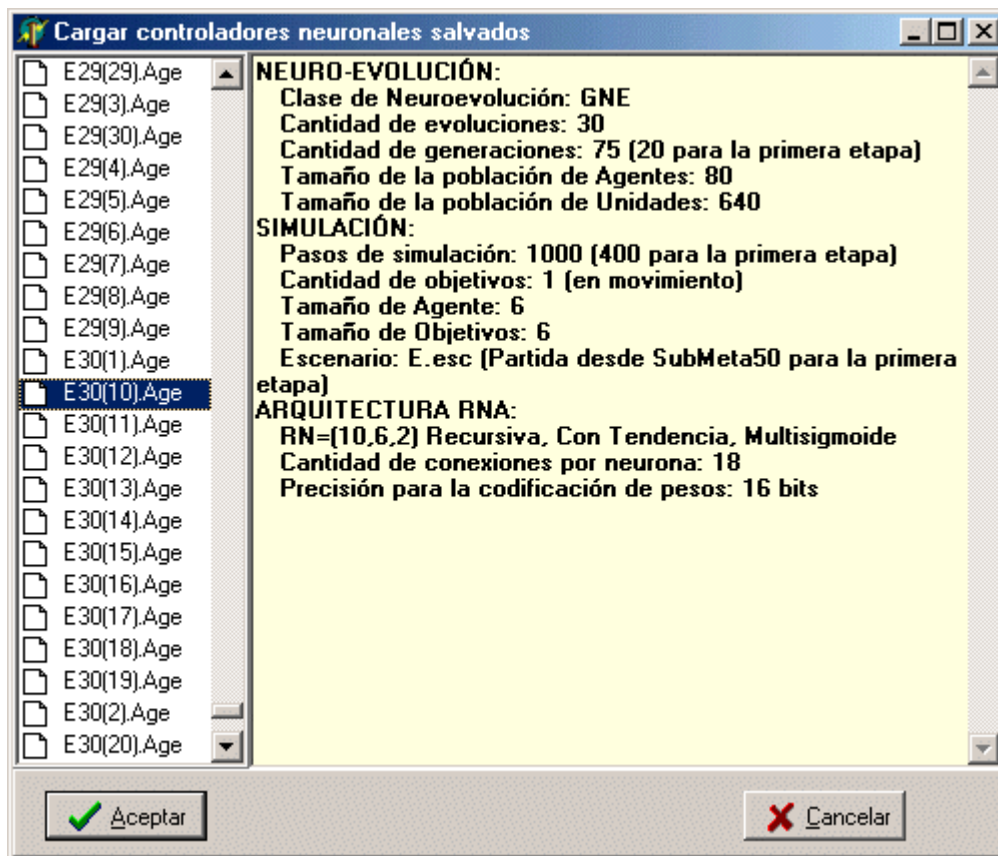


Ilustración 130 Ventana “Cargar controladores neuronales salvados”

En la lista de la izquierda puede observarse el nombre de los archivos en que se han salvado los controladores neuronales. Existe uno por cada evolución de cada ensayo neuroevolutivo. El cuadro de texto de la derecha muestra una descripción compacta del conjunto de valores de parámetros elegidos para el ensayo correspondiente a la selección en la lista de archivos. A modo de ejemplo, obsérvese que en la Ilustración 130 se ha seleccionado para cargar los controladores neuronales de la evolución número 10 del ensayo neuroevolutivo denominado “E30”, y es precisamente la configuración elegida para realizar este ensayo la que se muestra en el cuadro de texto de la derecha.

Al presionar el botón “Aceptar”, se cierra la ventana, se cargan los controladores correspondientes, se crea una población de tantos agentes como controladores se hallan cargados y se corre una simulación en el campo de prueba correspondiente. A diferencia de lo que ocurre cuando se está realizando un ensayo neuroevolutivo, aquí no hay evolución, concluida la cantidad de pasos de la simulación, ésta se detiene. No obstante, presionando el botón “Correr” vuelve a iniciarse.

Salvo el avance de la evolución, todas las demás funcionalidades están activas, así es posible ejecutar paso a paso la simulación o reanudar la ejecución continua, localizar los objetivos uno a uno, activar o desactivar la visualización de las fuerzas, de las etiquetas con su número de identificación y de las antenas detectoras de obstáculos. Pero lo más importante, en cualquier momento se puede utilizar el botón “Ver RN” para inspeccionar en línea los controladores neuronales.

Así finaliza esta breve descripción del software desarrollado con el que se ha llevado a cabo toda la experimentación que se expondrá a continuación. Así es que, retomando la línea del capítulo anterior, se volverá sobre los temas más importantes en este trabajo de grado: las distintas arquitecturas neuronales y estrategias neuroevolutivas implementadas. Esta vez para describir los ensayos efectuados y realizar el aún pendiente análisis de rendimiento de cada una de ellas.

Evaluación y Análisis

Introducción

Se pretende en este capítulo realizar un estudio comparativo sobre el rendimiento de las arquitecturas neuronales y estrategias neuroevolutivas presentadas, basado en los datos obtenidos por la experimentación que más adelante se detallará.

Todas las pruebas aquí presentadas han sido realizadas con el programa de computación ENE –Entorno Neuroevolucionario– que acompaña el presente trabajo. Con él se han realizado cada uno de los ensayos neuroevolutivos, salvado los datos relevantes y presentados en forma de gráfico para su posterior análisis.

Se han definido varios casos de pruebas tendientes a evaluar el rendimiento de las redes de tipo feedforward y recurrentes con distinta cantidad de neuronas ocultas, con conexión de tendencia y sin ella, con evolución de función de transferencia y con sigmoide fija.

Respecto a las estrategias neuroevolutivas se han testeado el rendimiento de las seis implementadas, enfatizando el análisis de SANE, ESP con población de Blueprints y GNE sobre diversos escenarios de distintas complejidades. La especial atención prestada a ellas responde a dos razones:

1. De acuerdo con los autores de SANE [Moriarty96] [Moriarty97_1] [Moriarty97_2], esta estrategia ha demostrado las ventajas de una coevolución cooperativa en la búsqueda de soluciones a problemas de control, evidenciándose superior a las estrategias tradicionales que codifican en un solo cromosoma una red neuronal completa. Además es frecuentemente citada por otros investigadores, se han propuesto algunas variantes –por ejemplo ESP– y se ha convertido en un referente comparativo interesante para evaluar el desempeño de otros métodos.
2. ESP con población de Blueprints y, especialmente GNE aportan aspectos novedosos en el área que se desean evaluar.

El presente capítulo está organizado de la siguiente manera. A continuación se presentará el conjunto de pruebas realizadas. Para ello se expondrá con precisión la forma en que se han recolectado los datos y la manera en que serán analizados.

Luego se realizará una breve descripción de los cinco escenarios definidos para realizar la experimentación junto a alguna apreciación sobre la complejidad que representa su resolución.

Así se dará paso a la exposición de los experimentos ensayados tendientes a evaluar el rendimiento de las distintas arquitecturas neuronales en primer lugar, y luego los necesarios para analizar el desempeño de las estrategias neuroevolutivas. Estos ensayos se presentarán en grupos a ser comparados argumentando su motivación y acompañados por los gráficos de curvas necesarios desde los que se elaborarán los resultados.

Y por último, para terminar el capítulo, se presentarán de forma ordenada las conclusiones que se desprenden directamente de la experimentación.

Las pruebas

Debido a que las estrategias implementadas comienzan siempre por una población generada aleatoriamente, y esto ocurre con cualquier algoritmo evolucionario tradicional, su desempeño

suele verse bastante afectado por las particularidades de esta primera población. Por lo tanto, para obtener datos representativos de cualquier caso de prueba planteado, no basta con la ejecución de un solo proceso evolutivo sino por el contrario, son necesarias varias ejecuciones para poder promediar los valores.

Recordemos el concepto de ensayo neuroevolutivo asumido en este trabajo. Se nombra así a la ejecución de n evoluciones para una determinada configuración de parámetros elegida que caracteriza un caso de prueba que se desea analizar. Un ensayo permitirá la recopilación de datos necesaria para realizar el análisis comparativo con los obtenidos por medio de otros ensayos.

Son varios los parámetros a fijar para cada prueba y en este capítulo se comentarán los más relevantes para la descripción de cada una de ellas. Para más detalle ver “Apéndice A. Ensayos realizados” en la pág. 289, donde se expone el conjunto completo de valores elegidos para cada uno de los ensayos neuroevolutivos realizados.

Respecto a estos valores de parámetros sólo se desea agregar que el tamaño de la población de unidades y la cantidad de conexiones por neurona se han elegido, salvo excepciones debidamente indicadas, de acuerdo a los valores sugeridos por el software que acompaña el presente trabajo. Recordemos que el tamaño de la población de unidades sugerido es aquel que asegura tantas neuronas –distribuidas en una o varias poblaciones de unidades, dependiendo de la estrategia neuroevolutiva– como demanda de ellas haya para la construcción de las redes neuronales que se necesiten. Por otro lado, la cantidad sugerida de conexiones a codificar por unidad es igual al número total de neuronas con que dicha unidad puede conectarse, esto es la cantidad de neuronas de entrada más la cantidad de neuronas de salida si la red es feedforward, o la cantidad total de neuronas de la red si ésta es recurrente.

Antes de comenzar con la presentación de los casos de prueba y el análisis de los datos obtenidos se darán algunas precisiones acerca de cómo se recolectan estos datos y sobre las particularidades de los escenarios definidos para las simulaciones.

Obtención de datos

Cada caso de prueba planteado se corresponde con un ensayo neuroevolutivo, obteniéndose los valores a ser analizados de la siguiente manera: Se realizan n procesos evolutivos de g generaciones cada uno. Por cada generación de cada evolución se salvan el mejor fitness obtenido en la misma y, si el objetivo fue alcanzado, la cantidad de pasos efectuados por el mejor agente para lograrlo.

Con estos datos así obtenidos se puede calcular el fitness promedio y el mejor fitness para cada generación. También la velocidad promedio y mejor velocidad para alcanzar el objetivo –medido en pasos de simulación– para cada generación.

Se trabaja entonces con dos matrices, una matriz \mathbf{F} con los valores de fitness, y una matriz \mathbf{V} con los valores relativos a la velocidad para alcanzar el objetivo.

Matriz F

		<i>Generaciones</i>		
<i>Evoluciones</i>	$f_{1,1}$		$f_{1,g}$	
	$f_{n,1}$		$f_{n,g}$	
<i>Vector P</i>	p_1		p_g	Promedio de cada columna
<i>Vector M</i>	m_1		m_g	Máximo de cada columna

Así cada elemento f_{ij} de la matriz **F** representa el fitness del mejor individuo de la generación j obtenido en el proceso de evolución número i .

Los elemento p_j conforman el fitness promedio representativo de la generación número j , calculado entre las n evoluciones.

Los elemento m_j representan al mejor fitness alcanzado en la generación j , en cualquiera de las n evoluciones.

El tratamiento de la matriz **V** es exactamente igual al de la matriz **F**, salvo que v_{ij} será igual a la cantidad de pasos de simulación necesarios para alcanzar el objetivo por el mejor individuo de la generación j en el proceso de evolución número i , ó 0 (cero) en caso de que ningún agente haya logrado alcanzarlo. Pero además de un vector de promedios y de mejores, se utiliza un vector de éxitos $E = (e_1, e_2, \dots, e_g)$ donde e_j es igual a la cantidad de elementos distintos de cero de la columna j de la matriz **V**.

Salvar la cantidad de pasos necesarios para alcanzar el objetivo, en caso de hacerlo, resulta interesante al momento del análisis de los datos. No sólo se puede comparar la velocidad de los agentes para alcanzar el destino correspondiente, sino también la cantidad de veces de las n evoluciones realizadas que se ha alcanzado el objetivo en cada una de las g generaciones. Esto resulta especialmente útil debido a que a veces el fitness –promedio o mejores– puede no arrojar diferencias significativas y sin embargo la cantidad de éxitos puede ser considerablemente mayor en un ensayo que en otro. También brinda una idea aproximada de la probabilidad de alcanzar el objetivo en una evolución para cada una de las generaciones.

Para las pruebas aquí realizadas, se ha elegido conformar cada ensayo por la realización de 30 procesos evolutivos en todo los casos. Cada evolución consta de 75 generaciones la mayoría de las veces y en algunos casos, donde la naturaleza del problema lo ha permitido, se utilizaron sólo 60 generaciones para agilizar el desarrollo de las pruebas.

Se han realizado 30 ensayos diferentes para evaluar el desempeño de distintas arquitecturas neuronales y estrategias neuroevolutivas. Cada uno de ellos será identificado por un número precedido por la letra “E”. Los detalles sobre el valor de los parámetros que determina cada uno de los ensayos {E01, E02, ... E30} puede consultarse en “Apéndice A. Ensayos realizados” en la pág. 289.

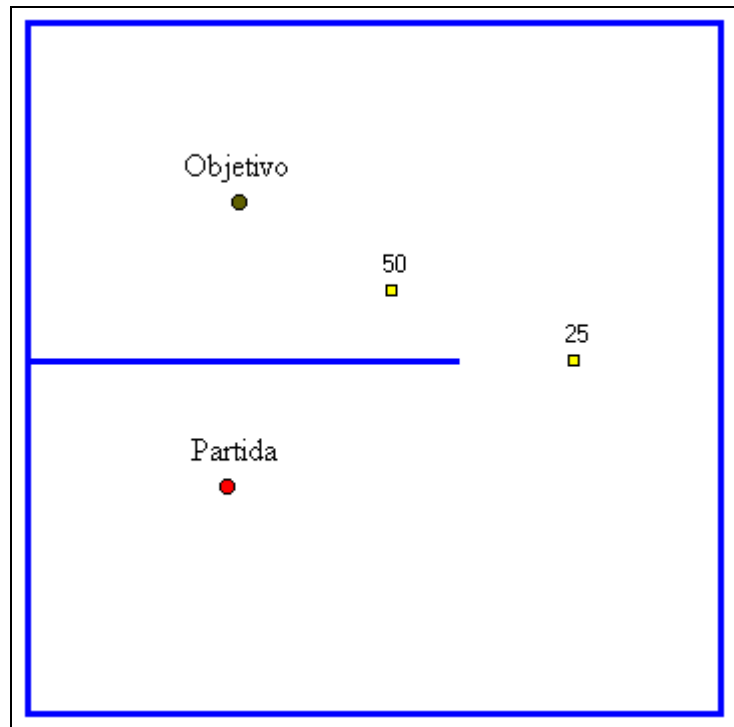


Ilustración 132 Escenario B. Complejidad: Media Baja.

Escenario C

El escenario C representa un problema de complejidad media alta debido a la disposición de los obstáculos, al punto de partida y a la posición del objetivo. Existen zonas donde las entradas a los controladores neuronales son muy parecidas, sin embargo el comportamiento debe ser bastante diferente. Este escenario ha sido utilizado en los test destinado a evaluar el desempeño de las estrategias neuroevolutivas más prometedoras y, al igual que el escenario A, en la evaluación de distintos tipos de arquitecturas neuronales.

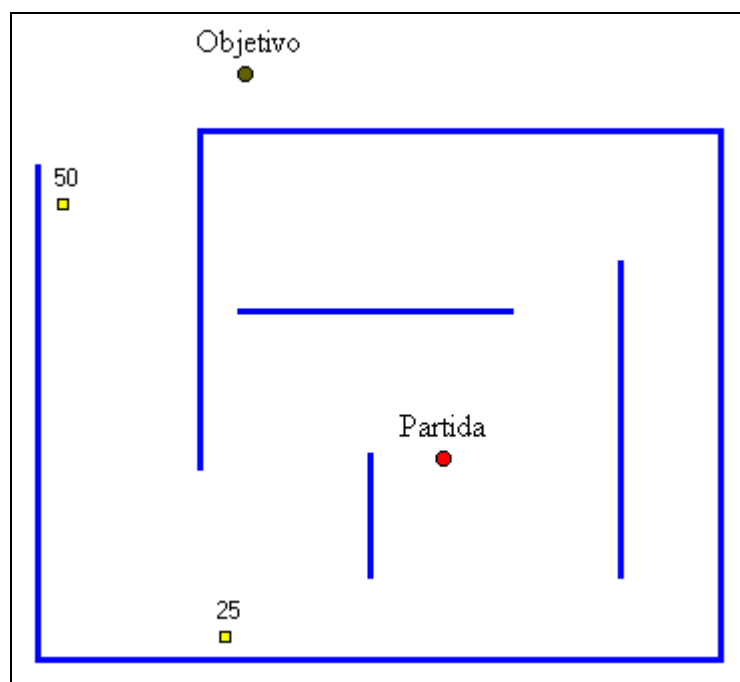


Ilustración 133 Escenario C. Complejidad: Media Alta.

Escenario D

Representa un problema de alta complejidad para los agentes. Existen zonas donde las entradas a los controladores neuronales, a excepción de la neurona que detecta el tiempo de simulación, son exactamente iguales, sin embargo el comportamiento debe ser totalmente diferente. Claro que es posible hallar una trayectoria donde esto no suceda. Se han dado más precisiones sobre la dificultad de un escenario de estas características en el apartado “Los agentes” en la pág. 175 –ver Ilustración 83–.

Este escenario ha sido utilizado en los test destinado a evaluar el desempeño de las estrategias neuroevolutivas más prometedoras.

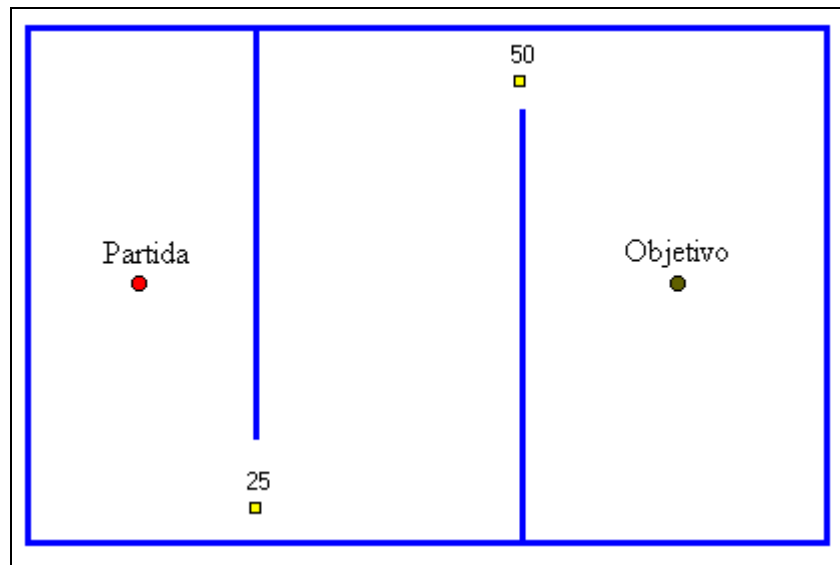


Ilustración 134 Escenario D. Complejidad: Alta

Escenario E

Este escenario no representa la complejidad del anterior en relación a la disposición de los obstáculos, pero presenta una distinta: el objetivo es móvil. Estamos así en presencia de un problema de búsqueda y captura de presa. Obsérvese además que el objetivo –la presa– no se encuentra a la vista de los agentes al comienzo de la simulación.

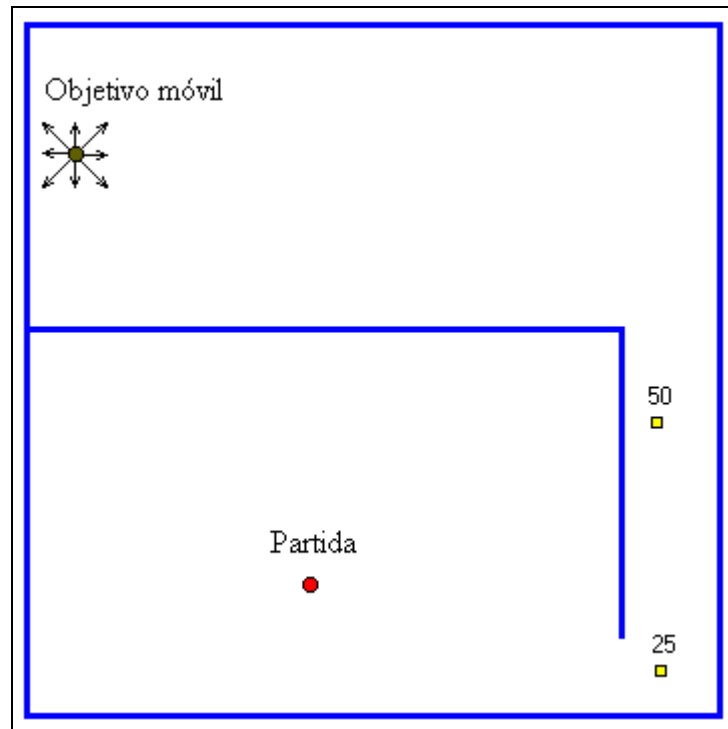


Ilustración 135 Escenario E. Complejidad: Media Alta

Evaluación de Arquitectura neuronal

Para evaluar el desempeño de distintas arquitecturas neuronales se han efectuado 14 ensayos diferentes E01, E02, ... E14. Para estos ensayos, previos a la evaluación de las estrategias neuroevolutivas, se debe determinar cual de ellas utilizar en su realización. Se ha elegido usar SANE porque, como ya mencionamos es un método bien documentado por sus desarrolladores quienes lo han probado con éxito en muchos casos diferentes superando el desempeño medio de las estrategias tradicionales. Además es frecuentemente citado y tomado como referencia por otros investigadores.

Pruebas sobre la topología y cantidad de neuronas ocultas

En primera instancia se desea conocer cómo se desempeñan las dos grandes clases de arquitecturas neuronales que existen y que se han implementado en el presente trabajo, me refiero a las redes de tipo feedforward y a las de tipo recurrentes. Simultáneamente se intentará determinar un número razonable de neuronas ocultas para conformar redes eficientes en la resolución de los problemas planteados evitando grandes estructuras que consumen demasiados recursos de cómputo.

Para estos primeros 7 ensayos se ha utilizado el escenario A de baja complejidad y una población de 50 agentes. Todas las redes neuronales poseen 10 neuronas de entrada –4 para detectar los objetivos– y todas ellas son multisigmoides y con término de tendencia. Salvo en E07 cada evolución consta de 75 generaciones. Veamos el esquema de estas pruebas:

Ensayo	Características	Motivación
E01	Feedforward, 3 neuronas ocultas	Comparar entre sí para determinar ventajas en el rendimiento surgidas de la arquitectura.
E02	Feedforward, 6 neuronas ocultas	
E03	Feedforward, 10 neuronas ocultas	
E04	Recurrente, 3 neuronas ocultas	
E05	Recurrente, 6 neuronas ocultas	
E06	Recurrente, 10 neuronas ocultas	
E07	Feedforward, 10 neuronas ocultas, 22 conexiones, 200 generaciones.	Se pretende comprobar cuánto favorece una mayor cantidad de conexiones y si las redes Feedforward pueden alcanzar el rendimiento de las Recurrentes con evoluciones de mayor número de generaciones.

Veamos los resultados obtenidos:

Para mayor claridad se presentarán los resultados en varios gráficos. En todos ellos y subsiguientes se aconseja observar la escala del eje Y, pues es posible que se utilicen distintas para facilitar la visualización en cada caso.

Redes feedforward

Comenzamos por el análisis comparativo entre las redes feedforward para determinar cómo influye la cantidad de neuronas ocultas en esta clase de redes.

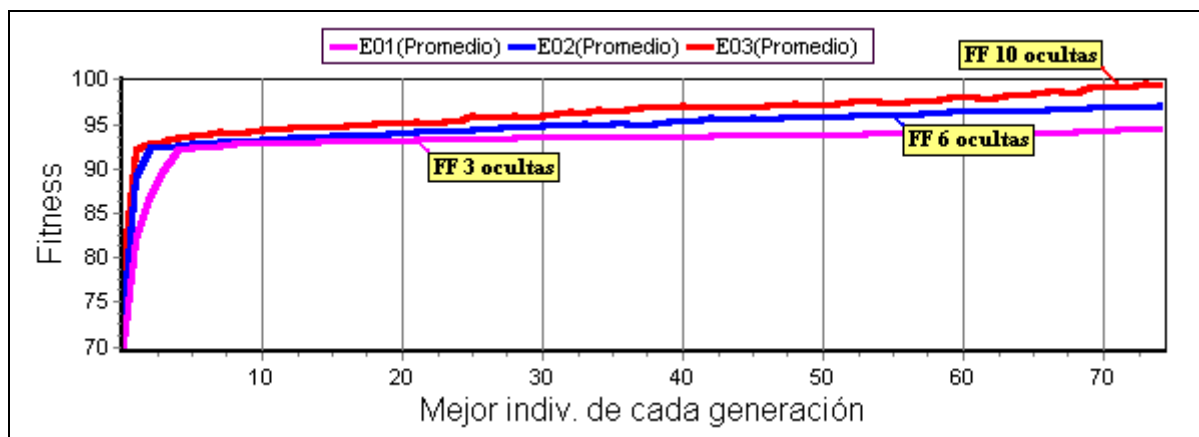


Ilustración 136 Información del fitness obtenido, utilizando SANE, por redes feedforward con distinta cantidad de neuronas ocultas.

Claramente se consigue mejor rendimiento aumentando la cantidad de neuronas ocultas en las arquitecturas neuronales feedforward. No obstante, se debe destacar que la diferencia evidenciada, aunque es fácilmente apreciable, no resulta demasiado grande si se comparan con otras diferencias que surgirán a medida que avancemos y se presenten otros resultados.

Sin embargo, las ventajas de utilizar redes con mayor número de neuronas ocultas se manifiesta más claramente por el aumento en la cantidad de éxitos obtenidos por generación, como lo muestra el siguiente gráfico.

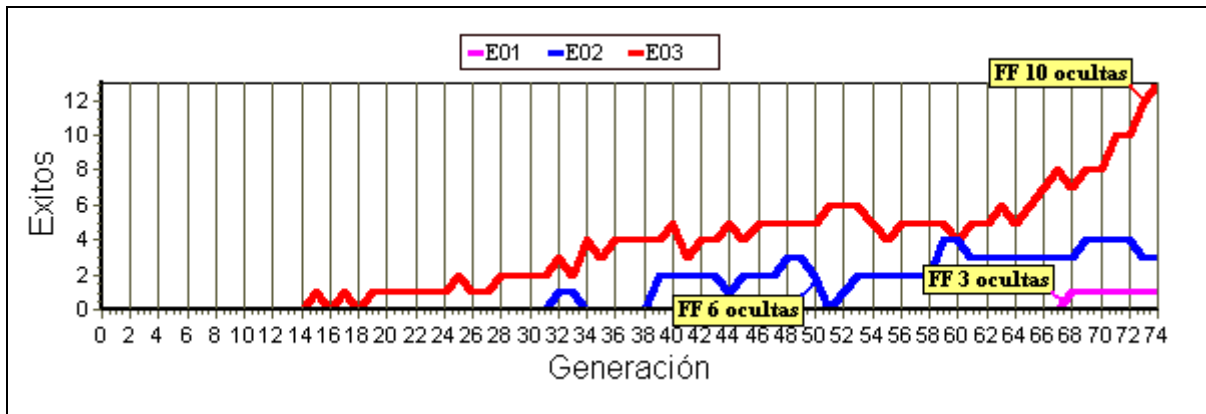


Ilustración 137 Información de la cantidad de éxitos obtenidos, utilizando SANE, por redes feedforward con distinta cantidad de neuronas ocultas.

Las diferencias aquí son más evidentes. Observemos cómo en las últimas generaciones aumenta la cantidad de éxitos en redes con 10 neuronas ocultas. Recordemos que todos los ensayos efectuados constan de 30 evoluciones cada uno. Así podemos interpretar el gráfico anterior de la siguiente forma: Si se realiza una evolución con redes feedforward con 10 neuronas ocultas se tendrá una probabilidad aproximada de tener éxito en la generación 74 de 13/30, sin embargo, si se utilizan redes feedforward con sólo 3 neuronas ocultas, esta probabilidad se reduce a 1/30.

Redes Recurrentes

Se presentará ahora la comparación entre las redes recurrentes con distinta cantidad de neuronas ocultas entre sí y también respecto a las del tipo feedforward. Para ello se introduce en el gráfico al mejor representante de estas últimas. Obsérvese que la escala del eje Y es distinta a la utilizada en la Ilustración 136.

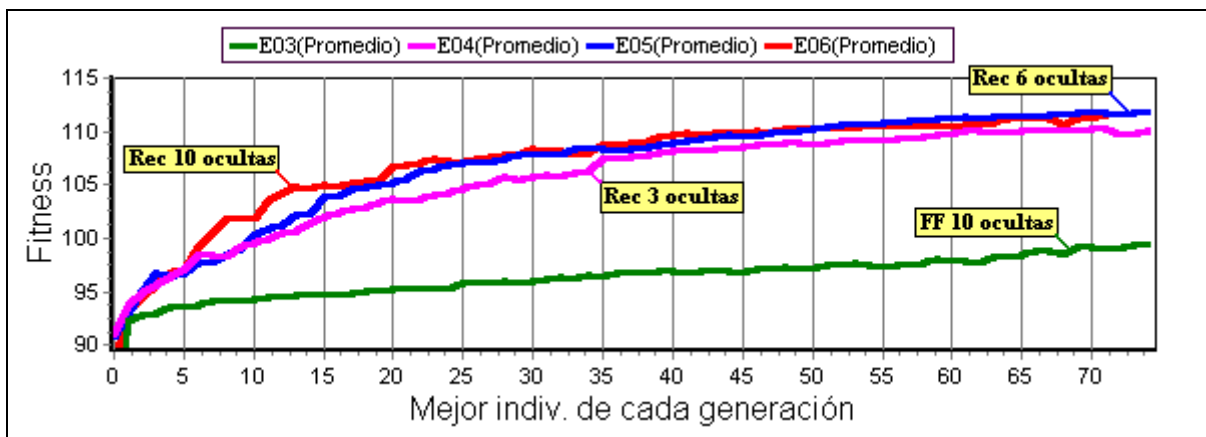


Ilustración 138 Comparación del fitness obtenido, utilizando SANE, por redes recurrentes con distinta cantidad de neuronas ocultas entre sí y con el mejor representante de las redes feedforward surgido en el análisis anterior.

En el gráfico anterior se observa una marcada superioridad en el rendimiento de la evolución de redes recurrentes respecto a la evolución de redes feedforward. También se evidencia una relación directamente proporcional entre la cantidad de neuronas ocultas y el fitness obtenido por las redes recurrentes. No obstante esta diferencia se hace menor a medida que avanzan las generaciones a tal punto que hacia la generación 70 la utilización de 6 neuronas ocultas ó 10 parece arrojar los mismos resultados.

Un análisis similar surge del estudio comparativo en la cantidad de éxitos obtenidos por las distintas arquitecturas evaluadas.

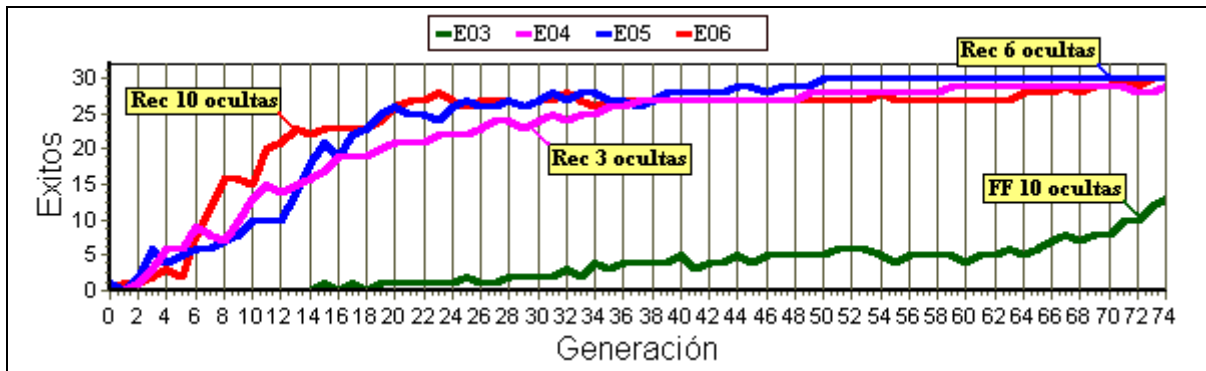


Ilustración 139 Comparación de la cantidad de éxitos obtenidos, utilizando SANE, por redes recurrentes con distinta cantidad de neuronas ocultas entre sí y con el mejor representante de las redes feedforward surgido en el análisis anterior.

Obsérvese cuanta diferencia existe en relación con las redes feedforward. Aquí se puede decir que si realizamos una evolución de redes recurrentes con 6 neuronas ocultas, tendremos una probabilidad cercana al 100% de hallar un individuo que resuelva satisfactoriamente el problema a partir de la generación número 50. También se puede decir que si realizamos una evolución de sólo 10 generaciones con redes recurrentes con 10 neuronas ocultas tendremos una probabilidad aproximada de 50% de tener éxito con alguna de ellas.

Evidentemente la evolución de redes recurrentes es ampliamente superior a la evolución de redes feedforward puesta de manifiesto en la velocidad de evolución. Sin embargo un parámetro de la estrategia neuroevolutiva elegida –SANE–, podría estar influyendo en el rendimiento a favor de las redes recurrentes. Este parámetro es la cantidad de conexiones a codificar por neurona en los cromosomas de la población de unidades. Como se recordará, se utiliza el valor sugerido por el software desarrollado que lo calcula como la cantidad total de neuronas con las que aquella codificada podría conectarse. De esta manera se fijó en 12 la cantidad de conexiones para las redes feedforward y en 22 para las redes recurrentes con 10 neuronas ocultas.

Por lo dicho en el párrafo anterior, resultará ilustrativo comparar los resultados con la evolución de redes feedforward con 10 neuronas ocultas y con 22 conexiones codificadas en los cromosomas de la población de unidades. Para ello se ha realizado el ensayo E07 que pretende también averiguar qué sucede más allá de la generación 75.

La insistencia con las redes feedforward está fundada en el hecho de que éstas suponen en general una menor carga de procesamiento que las recurrentes y por lo tanto serán preferibles en muchos casos. Dependiendo de las circunstancias un proceso evolutivo más largo pero que asegure un resultado con menor carga computacional puede ser ventajoso respecto de otro que asegure un buen resultado en pocas generaciones pero que implique decisiones lentas en la etapa de producción, debido a los recursos de cómputo requeridos. Un ejemplo podría ser un sistema de tiempo real donde la respuesta a tiempo es crítica una vez que el sistema es puesto en marcha.

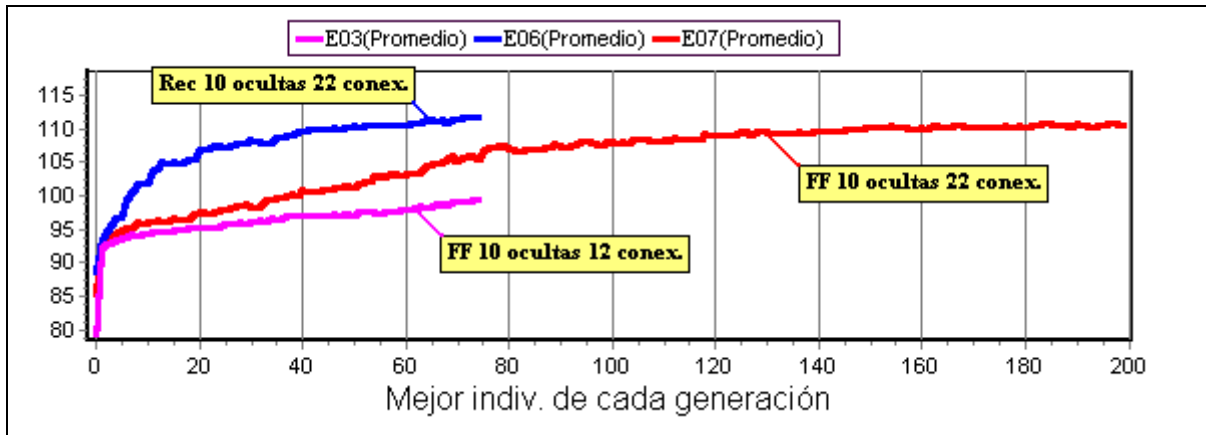


Ilustración 140 Comparación del fitness obtenido, utilizando SANE, por redes feedforward con 22 conexiones por unidad con otras arquitecturas analizadas anteriormente.

Como se puede apreciar en el gráfico anterior la mayor cantidad de conexiones codificadas por neurona supone un mejor rendimiento en las redes feedforward pero de todas maneras sigue siendo inferior al conseguido por las redes recurrentes. No obstante puede observarse que si se realizan procesos evolutivos más extensos, es decir, con mayor número de generaciones, el rendimiento de las redes feedforward sigue mejorando.

Se puede concluir entonces que las redes recurrentes evolucionan más rápido que las feedforward, pero ¿Son absolutamente superiores en su rendimiento? Es decir, ¿La mejor red recurrente es insuperable por la mejor red feedforward? El siguiente gráfico despejará esta incógnita.

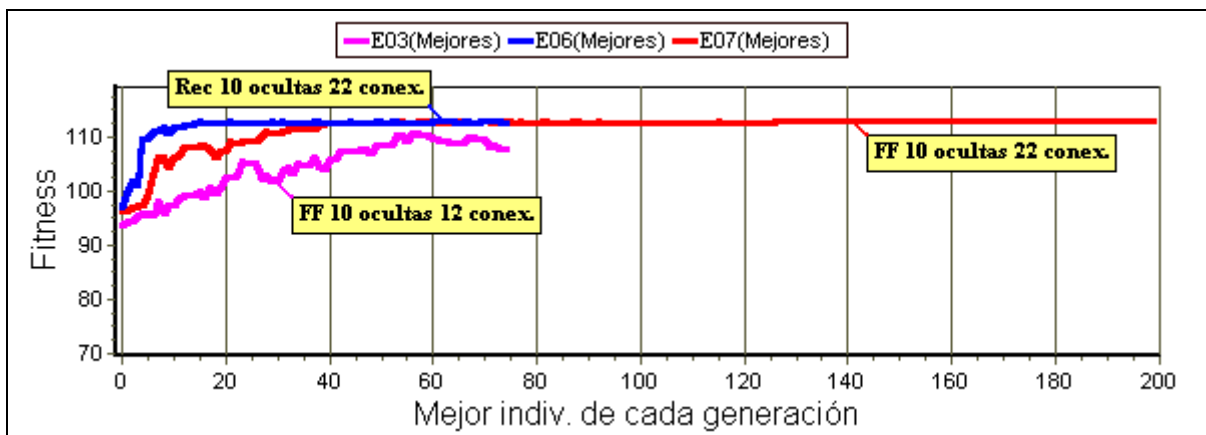


Ilustración 141 Información del fitness obtenido, utilizando SANE, por el mejor individuo de cada generación seleccionado entre las 30 evoluciones realizadas.

Se aprecia claramente que las redes feedforward son capaces de igualar el rendimiento de las recurrentes. Del gráfico se desprende que si se realizan 30 evoluciones es muy factible que se consiga una red del tipo feedforward con igual desempeño que una recurrente a partir de la relativamente temprana generación número 40. Como se dijo anteriormente, este hecho es importante porque está significando la posibilidad de hallar una red con mayor velocidad de respuesta, sin pérdida de calidad, a costa seguramente de repetir un mayor número de veces el proceso evolutivo.

Ahora se analizará cómo influye la conexión de tendencia y la evolución de la función de transferencia en el rendimiento de las redes neuronales. Para que la cantidad de pruebas a realizar no crezca en forma exponencial cada vez que se intente escudriñar alguna nueva

característica, se irán paulatinamente fijando ciertos parámetros a medida que avancemos en las pruebas.

Así se eligen las redes recurrentes con 6 neuronas ocultas para seguir adelante con las demás pruebas. La elección se fundamenta en la mayor velocidad de evolución de este tipo de redes y en que el rendimiento mostrado por redes recurrentes con 6 neuronas ocultas es similar al de redes recurrentes con 10 neuronas ocultas, pero implica menor carga de procesamiento que éstas últimas mejorando su velocidad de respuesta. Es decir, parece estar claro que la arquitectura elegida es la que presenta mejor relación entre carga de procesamiento y performance.

Pruebas sobre conexión de tendencia y función de transferencia

Para completar el análisis sobre las arquitecturas neuronales se han realizado 7 nuevos ensayos.

A partir de E05, cambiando las características multisigmoide y de conexión de tendencia, se han llevado a cabo los ensayos E08, E09 y E10. Recordemos que E05, y por ende los nuevos ensayos basados en él se realizan sobre un escenario simple, el escenario A. Para observar cómo afectan las características estudiadas en un escenario más complejo se repiten la misma clase de experimentos sobre el escenario C, realizando así los ensayos E11, E12, E13 y E14.

Ensayo	Características	Motivación
E08	Sin tendencia, no multisigmoide	Comparar entre sí y con E05 para determinar cómo afecta el rendimiento la evolución de función de transferencia y la existencia del término de tendencia.
E09	Con tendencia, no multisigmoide	
E10	Sin tendencia, multisigmoide	
E11	Con tendencia, multisigmoide. Sobre escenario C.	Comparar entre sí para determinar cómo afecta el rendimiento la evolución de función de transferencia y la existencia del término de tendencia en un escenario de mayor complejidad
E12	Sin tendencia, no multisigmoide. Sobre escenario C.	
E13	Con tendencia, no multisigmoide. Sobre escenario C.	
E14	Sin tendencia, multisigmoide. Sobre escenario C.	

Veamos los resultados obtenidos:

Primero se analizará el rendimiento de las redes para el escenario A de baja complejidad.

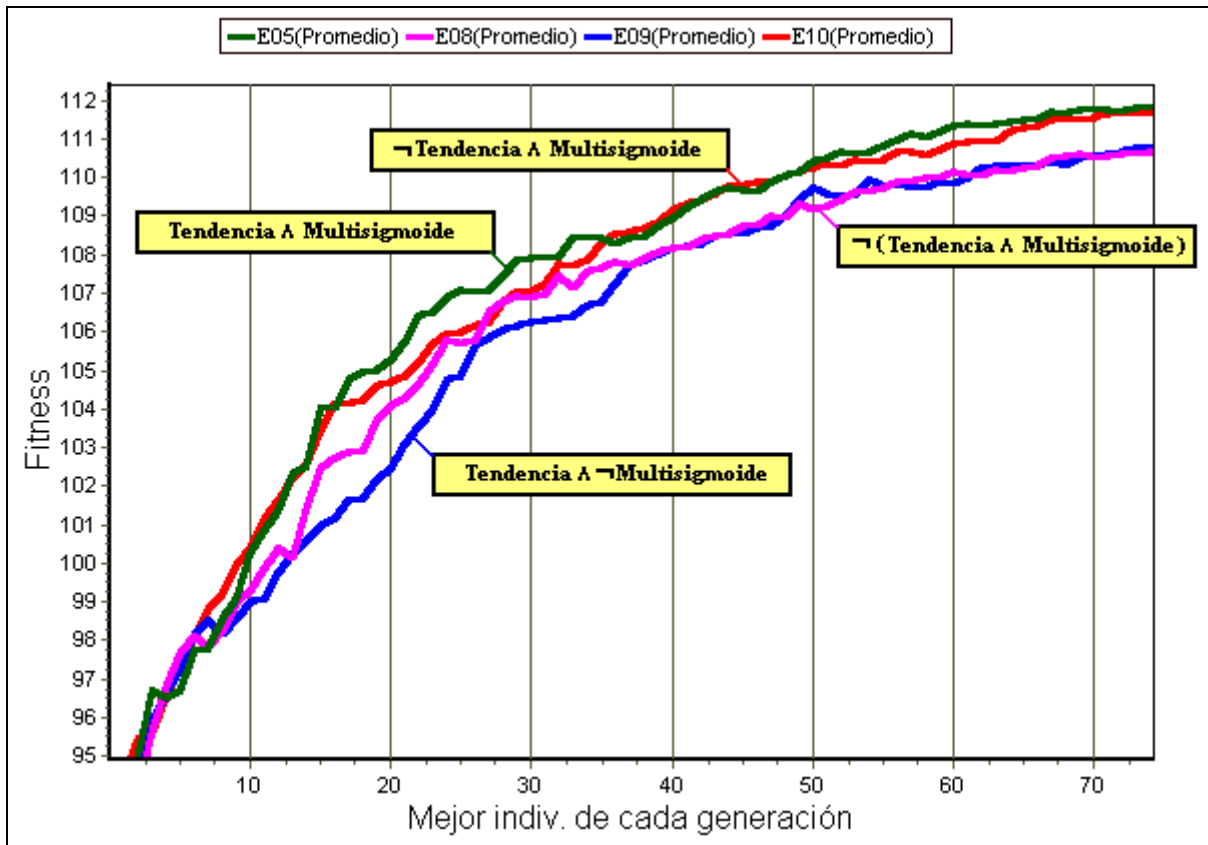


Ilustración 142 Impacto del término de tendencia y característica multisigmoide

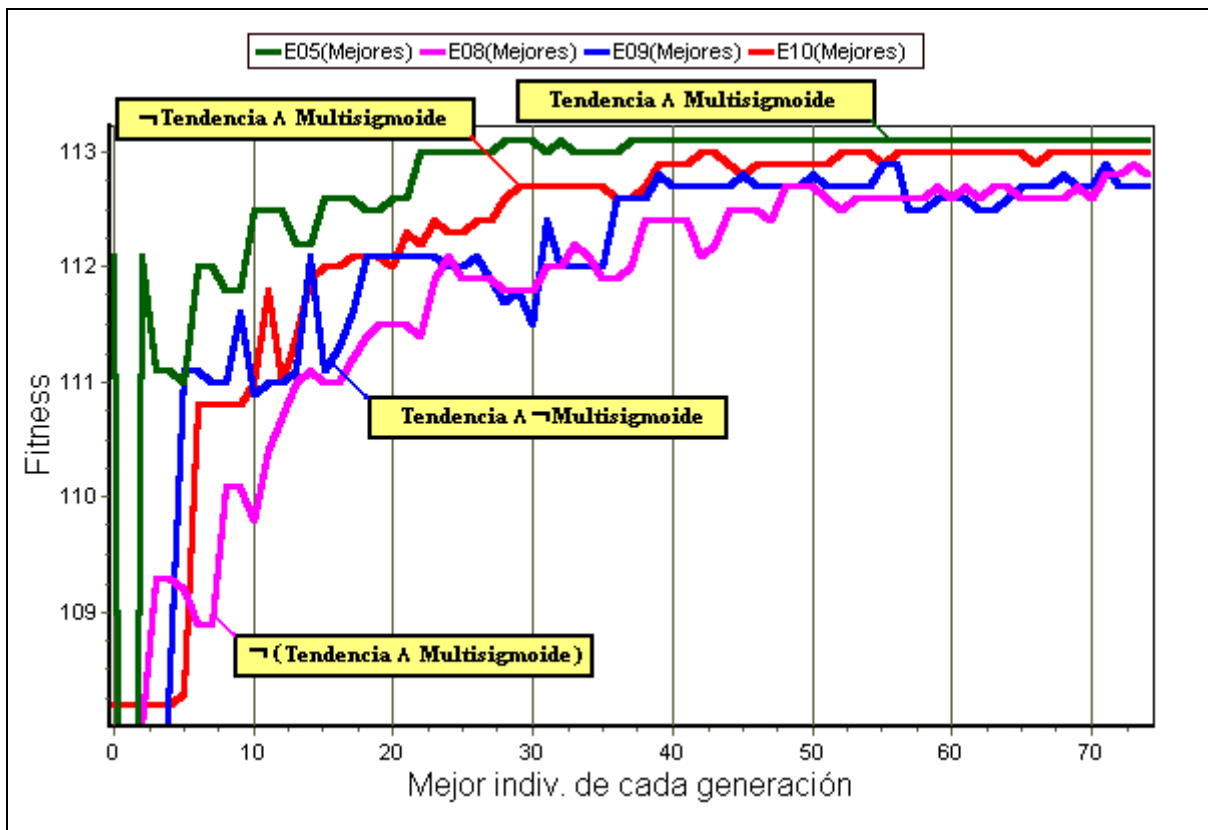


Ilustración 143 Impacto del término de tendencia y característica multisigmoide en las mejores redes obtenidas durante las 30 evoluciones.

Primero comentaré la Ilustración 142 donde se presentan las curvas del fitness promediado entre las 30 evoluciones de cada ensayo. Allí puede observarse claramente que las redes recurrentes multisigmoide poseen mejor rendimiento que aquellas que no lo son. Lo que no queda claro es cómo afecta la conexión de tendencia pues por momentos parece perjudicial – observar la curva E09 que está por debajo de E08 para un gran número de generaciones– aunque en ocasiones se muestra beneficiosa –observar que la curva E05 está por encima de E10 la mayor parte de las generaciones–.

Para arrojar luz en esta cuestión se ha incluido la Ilustración 143 donde se presenta el fitness de las mejores redes obtenidas entre los 30 procesos evolutivos. Aquí se evidencia más claramente la ventaja de utilizar tendencia respecto de no utilizarla, la curva E05 está por encima de E10 y la curva E09 por encima de E08 para la mayor cantidad de generaciones.

De acuerdo a estos gráficos se puede concluir que la conexión de tendencia es beneficiosa aunque no tanto como la evolución de función de transferencia. Como es de esperar, la presencia de ambas características produce aún mejores resultados.

No obstante se debe señalar que las diferencias aquí presentadas son muy pequeñas. Obsérvese la escala del eje del Fitness donde se ha aplicado un zoom considerable para poder apreciar las distintas curvas.

Se han realizado las mismas pruebas para un escenario de mayor grado de complejidad como lo es el escenario C, para observar si estas diferencias son más notorias cuando más difícil es la tarea a resolver. A continuación presentamos los resultados obtenidos.

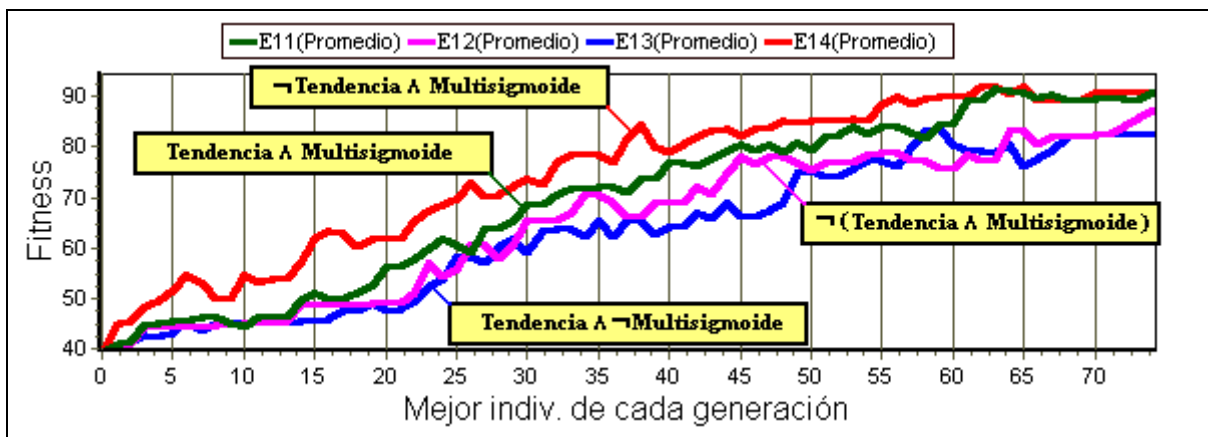


Ilustración 144 Impacto del término de tendencia y característica multisigmoide en la realización de una tarea de complejidad media alta –escenario C–.

Si observamos la escala del eje del Fitness, podemos apreciar que las diferencias manifestadas aquí son mayores a las presentadas en la Ilustración 142. Esto es razonable, ya que un problema sencillo puede ser resuelto muy bien por muchas arquitecturas, pero a medida que se complica la tarea a realizar ya no todas responderán de la misma forma, evidenciándose las más aptas para hacerlo.

En la ilustración anterior aparece con claridad la ventaja de utilizar redes multisigmoide respecto de no utilizarlas. Sin embargo, la conexión de tendencia parece ser más perjudicial que beneficiosa, al menos en las generaciones inferiores a la número 60 –observar la curva E11 alcanza el rendimiento de E14 en las proximidades de la generación 60, pero en las anteriores se mantiene definitivamente por debajo de ella–.

No obstante, se debe reconocer en la conexión de tendencia combinada con la característica multisigmoide, cierta capacidad de definición para alcanzar la meta que no se aprecia en

ausencia de cualquiera de estas dos particularidades. Esta situación se evidencia en el siguiente gráfico.

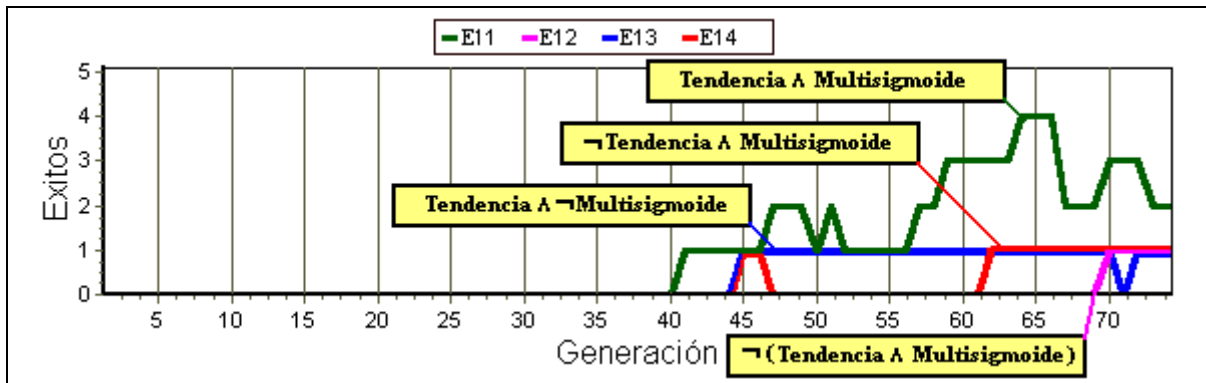


Ilustración 145 Impacto del término de tendencia y característica multismoide en la cantidad de éxitos obtenidos durante las 30 evoluciones para un escenario de complejidad media alta –escenario C–.

Se puede concluir que la evolución de la función de transferencia que se ha implementado en este trabajo –multismoide– mejora en todos los casos el rendimiento de las redes neuronales y no se encuentran motivos para no utilizarla.

El punto conflictivo será definir la utilidad de la conexión de tendencia. Como se explicó en el capítulo “Redes Neuronales Artificiales y Neuroevolución” para problemas de clasificación, en ciertas arquitecturas el término de tendencia es indispensable pues permite trasladar la recta de decisión del origen de coordenadas posibilitando así ciertas clasificaciones que de otra manera no podrían hacerse si no es cambiando el umbral de la función de transferencia en cada nodo de la red. Sin embargo, en el dominio de problemas aquí definido, la conexión de tendencia no es para nada necesaria y a veces pareciera perjudicial.

Para brindar una apreciación sobre la utilidad del término de tendencia se distinguirán dos propósitos. Si sólo interesa obtener un buen fitness –por ejemplo con agentes que logren posicionarse en las cercanías de su destino– pero no se requiere alcanzar efectivamente el objetivo una buena elección sería no utilizar conexión de tendencia pues podría degradar el fitness promedio.

Si, por el contrario, se requiere que el agente alcance el objetivo como puede ser en el caso de una simulación de captura de presa donde llegar cerca sin alcanzar el objetivo no produce ningún beneficio, se debería utilizar conexión de tendencia pues aumenta la probabilidad de lograrlo. Creo que este segundo caso es el más interesante y por ello aquí se utilizará término de tendencia.

En base a los datos hasta aquí presentados, se puede concluir que una buena elección para el dominio de problemas planteado será la utilización de redes recurrentes con 6 neuronas ocultas, conexión de tendencia y evolución de función de transferencia. Precisamente esta arquitectura neuronal es la adoptada para proseguir con los ensayos neuroevolutivos ahora destinados al análisis comparativo de las 6 estrategias implementadas en este trabajo aplicadas a la resolución de escenarios de distinta complejidad.

Evaluación de Estrategias Neuroevolutivas

Se realizará ahora el análisis de los datos obtenidos de varios ensayos destinados a poner en evidencia las diferencias en el rendimiento de las distintas estrategias neuroevolutivas implementadas. Si bien en principio se hará un estudio sobre las 6 estrategias presentadas,

luego se focalizará sobre ESP con Blueprints y GNE por ser los dos métodos que aportan aspectos novedosos.

Se han utilizado los escenarios B, C, D y E que presentan distintas complejidades para evaluar cómo se desempeñan las estrategias en distintas circunstancias. Se tomará como referente comparativo el desempeño de SANE en las distintas situaciones por razones antes mencionadas.

Pruebas sobre escenario B

Se ha elegido el escenario B para realizar un análisis comparativo del rendimiento de todas las estrategias implementadas en el presente trabajo. Se han llevado a cabo 7 ensayos neuroevolutivos a tal efecto.

Debido a los requerimientos de ESP sobre la cantidad de redes neuronales a evaluar que ya se han comentado bajo el título “Debilidad del método” –pág. 220–, se han realizado dos ensayos para este método con distinto tamaño de población para poder hacer así un análisis más certero.

A continuación presentamos el esquema de estos ensayos:

Ensayo	Características	Motivación
E15	Neuroevolución Simple	Comparar entre sí para determinar ventajas y desventajas de cada una de las estrategias neuroevolutivas implementadas.
E16	Neuroevolución Simple Elitista	
E17	SANE	
E18	ESP Clásico 80 agentes, 20 unidades	
E19	ESP Clásico 320 agentes, 80 unidades	
E20	ESP con Blueprints	
E21	GNE	

Veamos los resultados obtenidos:

Para simplificar los gráficos se irán comparando las estrategias por subgrupos. Se presenta en principio la relación existente entre la Neuroevolución Simple, Simple Elitista y SANE.

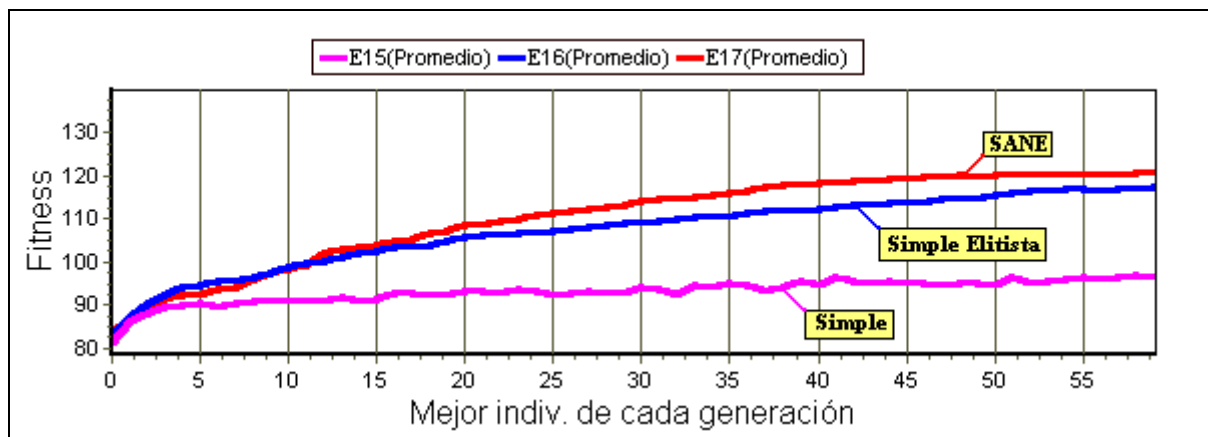


Ilustración 146 Información del fitness alcanzado por los agentes de acuerdo a la utilización de las estrategias NE simple, NE simple elitista y SANE.

La ilustración anterior es más que clara. SANE es mejor que la neuroevolución Simple Elitista, y esta última es mejor que la Neuroevolución Simple. Es llamativa, sin embargo, la gran diferencia de rendimiento que existe entre la segunda y tercera estrategia ranqueada.

Recordemos que la neuroevolución Simple Elitista, a diferencia de la neuroevolución Simple, toma el mejor individuo de cada generación y pasa dos copias idénticas, salvo eventual mutación, a la próxima generación. Esta sencilla modificación tiene un gran impacto sobre el desempeño del método mejorándolo sustancialmente.

Esta relación de desempeño entre las 3 estrategias también se manifiesta en la siguiente ilustración.

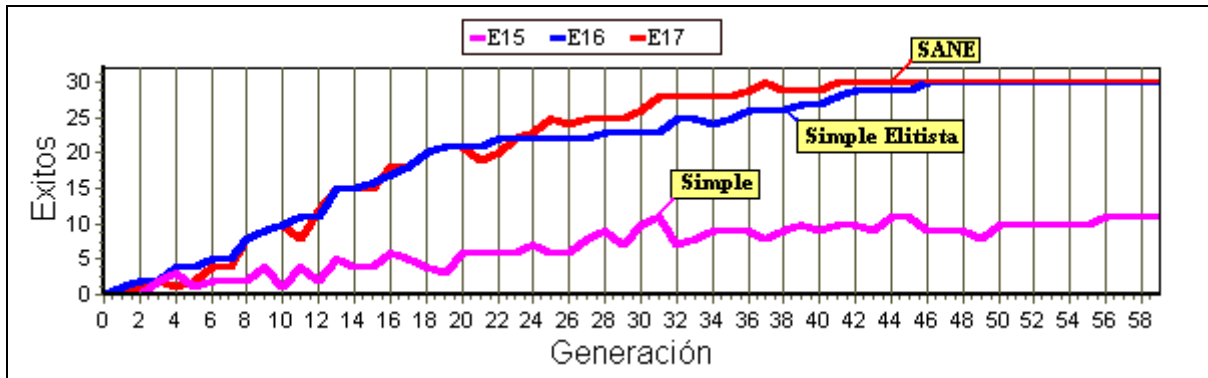


Ilustración 147 Información de la cantidad de éxitos obtenidos por cada generación durante las 30 evoluciones realizadas de acuerdo a la utilización de las estrategias NE Simple, NE Simple Elitista y SANE.

Nuevamente SANE ocupa el primer lugar en el podio, seguida de cerca por NE Simple Elitista y finalmente, muy rezagada encontramos a NE Simple. Obsérvese que podemos pronosticar para las estrategias SANE y NE Simple Elitista con mucha probabilidad de acierto que cualquier evolución alcanzará la meta a lo sumo en la generación 45.

Veamos ahora que podemos concluir respecto del método ESP. Recordemos que en este método, al no existir población de blueprints las redes neuronales se construyen por la selección aleatoria de una neurona en cada una de las subpoblaciones de unidades. El fitness de cada unidad se calcula como el promedio del obtenido por todas las redes neuronales en las que haya participado. Esto hace necesario construir un gran número de redes neuronales para asegurar que cada unidad de las subpoblaciones tenga chance de participar en varias redes. De no ser así, existirá el riesgo de que no todas las unidades sean seleccionadas. Una unidad no seleccionada tendrá valor de fitness 0, aunque sea la mejor existente en la subpoblación.

Al exponer ESP –ver “ESP –Enforced subpopulations–” en pág. 215– se ha indicado esta situación como una debilidad del método pues requiere mayores recursos de cómputo. En el dominio de problemas aquí definido, implica trabajar con una población de agentes de tamaño bastante superior al de cada una de las subpoblaciones de unidades. Por lo tanto o bien se reduce el tamaño de estas subpoblaciones o bien se aumenta la cantidad de agentes.

Se han realizado ambas opciones en los ensayos E18 y E19. La relación elegida ha sido de 1 a 4, es decir se construyen 4 veces más redes neuronales que el tamaño de las subpoblaciones de unidades. Así en E18 se reduce el tamaño de las subpoblaciones a 20 y se sigue utilizando una población de agentes de 80 individuos. En cambio, en E19 se deja las subpoblaciones con 80 unidades cada una y se aumenta a 320 el tamaño de la población de agentes.

Los resultados pueden observarse en el siguiente gráfico.

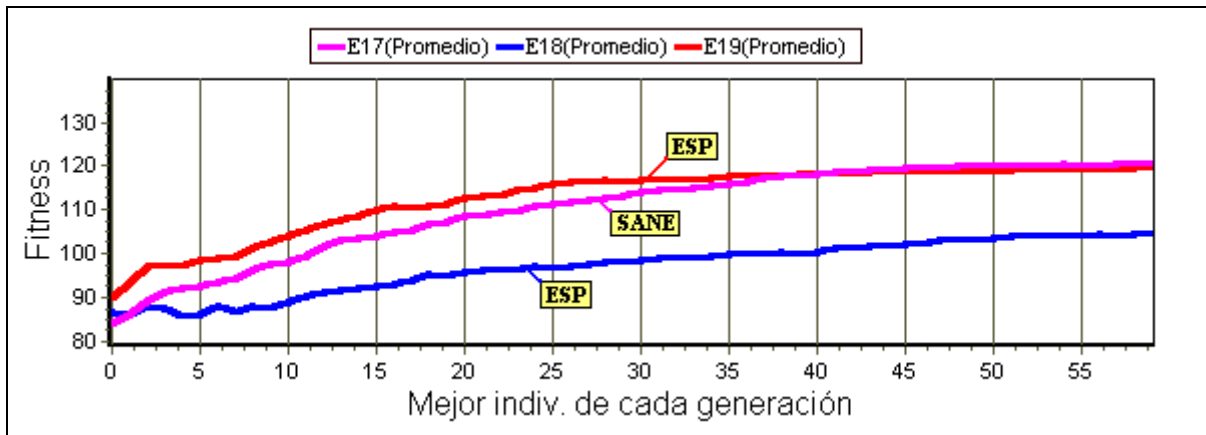


Ilustración 148 Información del fitness promedio alcanzado utilizando SANE y dos configuraciones de ESP.

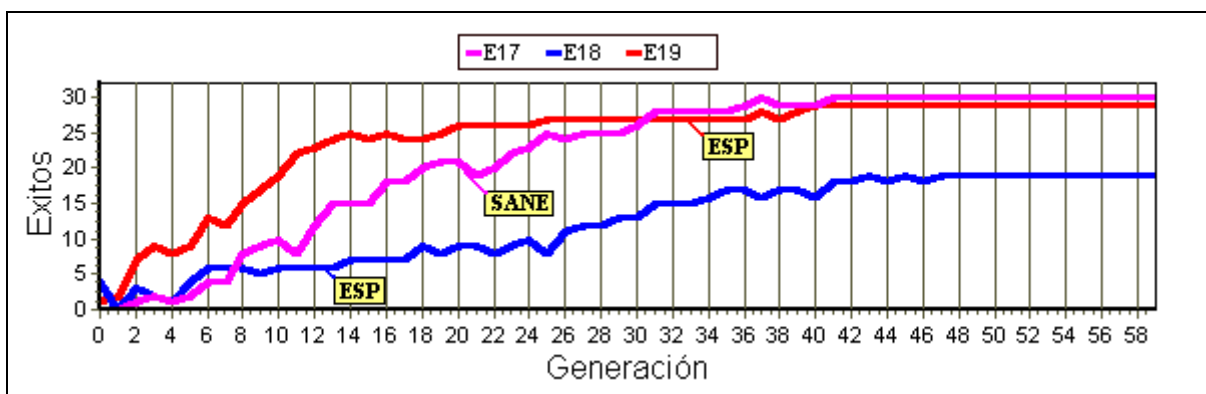


Ilustración 149 Información de la cantidad de éxitos obtenidos por cada generación a lo largo de las 30 evoluciones realizadas, utilizando SANE y dos configuraciones de ESP.

Ambas ilustraciones evidencian la misma situación. Si para la estrategia neuroevolutiva ESP se elige una población de agentes de igual tamaño a la que utilizamos con SANE –80 individuos en este caso– y subpoblaciones de 20 unidades, se obtiene un rendimiento muy por debajo del alcanzado por SANE. Si por el contrario utilizamos ESP con una población de agentes 4 veces más grande a la utilizada con SANE para contar con subpoblaciones de 80 unidades, se observa una mejora considerable respecto a este último pero sólo en las primeras generaciones. Al avanzar la evolución, más allá de la generación número 40 SANE alcanza y supera a ESP.

Se debe aclarar además que utilizar una población de 320 agentes como se hizo en el ensayo neuroevolutivo E19 agrega muchos minutos de procesamiento por cada evolución haciéndolo poco atractivo como alternativa a SANE.

Aún si mejorara sustancialmente la performance de SANE la gran cantidad de tiempo extra para llevar a cabo una evolución no lo convierten en una buena opción, sobre todo existiendo otras como ESP con blueprint y GNE que como veremos más adelante se presentan como ventajosas en muchas situaciones.

A partir de este momento, se comenzará con el análisis más interesante consistente en evaluar el comportamiento de ESP con población de blueprints, una hibridación entre SANE y ESP que se ha propuesto aquí, y GNE, método de neuroevolución incremental basado en grupos de redes neuronales que se presenta como alternativa viable en este trabajo de grado.

Recordemos que se ha implementado GNE utilizando dos etapas evolutivas. En la primera los agentes deben aprender a alcanzar su destino partiendo desde una posición intermedia del

recorrido que deberán transitar en la segunda etapa. Por lo tanto, en los gráficos presentados, debemos comparar las curvas de rendimiento sólo a partir del comienzo de la segunda etapa que es fácilmente identificable por la caída abrupta del fitness. Debido a que el escenario B no presenta demasiadas complicaciones, en el ensayo E21 se ha podido utilizar GNE con una muy corta primera etapa de sólo 3 generaciones.

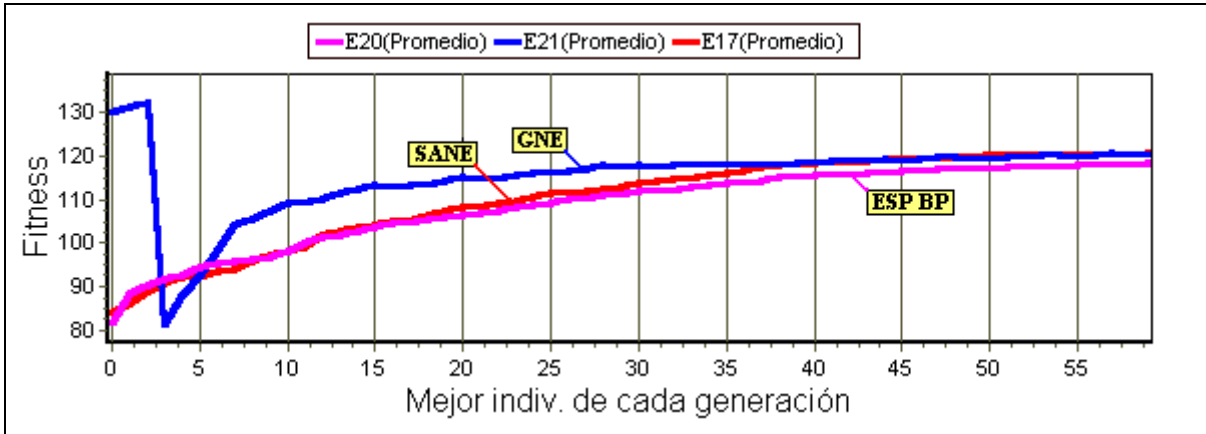


Ilustración 150 Información del fitness promedio alcanzado utilizando SANE, ESP con población de Blueprints y GNE –primera etapa de 3 generaciones–

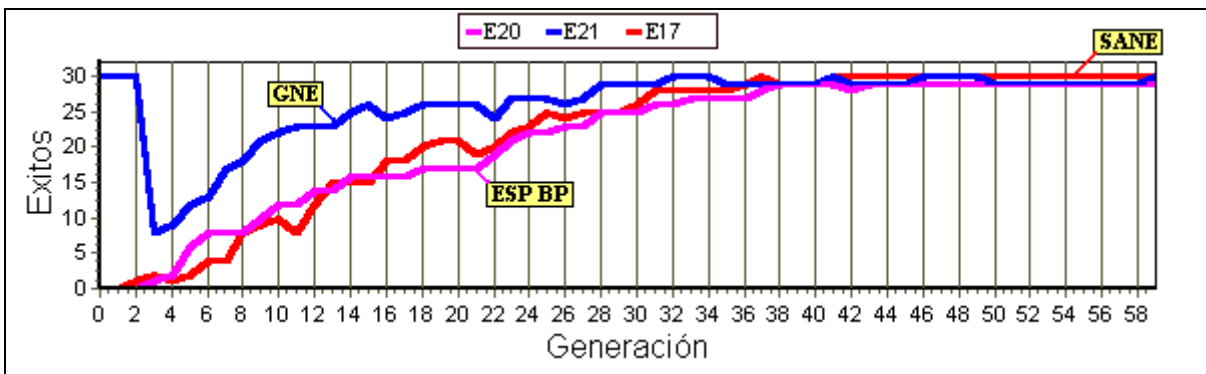


Ilustración 151 Información del total de éxitos alcanzado durante las 30 evoluciones, utilizando SANE, ESP con población de Blueprints y GNE –primera etapa de 3 generaciones–

Las dos ilustraciones anteriores ponen en evidencia la misma situación. ESP con población de Blueprints muestra un rendimiento ligeramente inferior a SANE salvo durante las primeras 15 generaciones donde la relación es la inversa. GNE, en cambio, aparece con igual desempeño que SANE pasadas las 40 generaciones, pero se muestra marcadamente superior en la primera mitad de la evolución. Por otro lado si prestamos atención a los mejores individuos logrados por cada estrategia debemos distinguir en primer lugar a GNE, seguido de ESP con población de Blueprints y finalmente SANE, como lo muestra la siguiente ilustración:

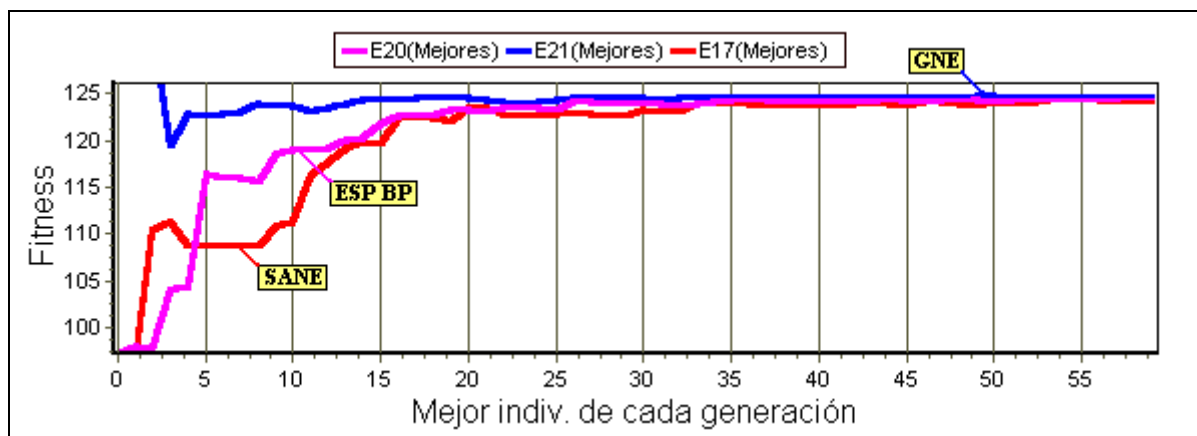


Ilustración 152 Información de los mejores fitness alcanzado durante las 30 evoluciones, utilizando SANE, ESP con población de Blueprints y GNE –primera etapa de 3 generaciones–.

Ya se ha mencionado que un problema que no representa grandes retos como podría ser el escenario B, en definitiva será bien resuelto por la mayoría de las estrategias neuroevolutivas. Por lo tanto es entendible no encontrar grandes diferencias de rendimiento sobre todo a medida que avanza la evolución hacia generaciones más altas. A continuación se comenzará con el análisis sobre escenarios más complejos.

Pruebas sobre escenario C

El escenario C representa un nivel de complejidad bastante mayor al escenario B. Por lo tanto resulta interesante observar el desempeño de las estrategias neuroevolutivas aquí desarrolladas. Como se explicó anteriormente, se proseguirá evaluando ESP con población de Blueprint y GNE tomando como referente comparativo a SANE –muchas veces citado por los investigadores del área y confiablemente superior a las alternativas neuroevolutivas tradicionales–.

Se reutilizarán los datos obtenidos durante el ensayo E11 que se usaron para evaluar el rendimiento de las arquitecturas neuronales y en el cual se utilizó SANE para la evolución. De esta manera, cambiando sólo la estrategia neuroevolutiva se obtuvo la configuración para los ensayos E22 y E23.

Ensayo	Características	Motivación
E22	ESP con población de Blueprints	Comparar entre sí y con E11 –SANE– para determinar ventajas y desventajas de estas estrategias en un escenario más complejo.
E23	GNE. 20 generaciones para la primera etapa.	

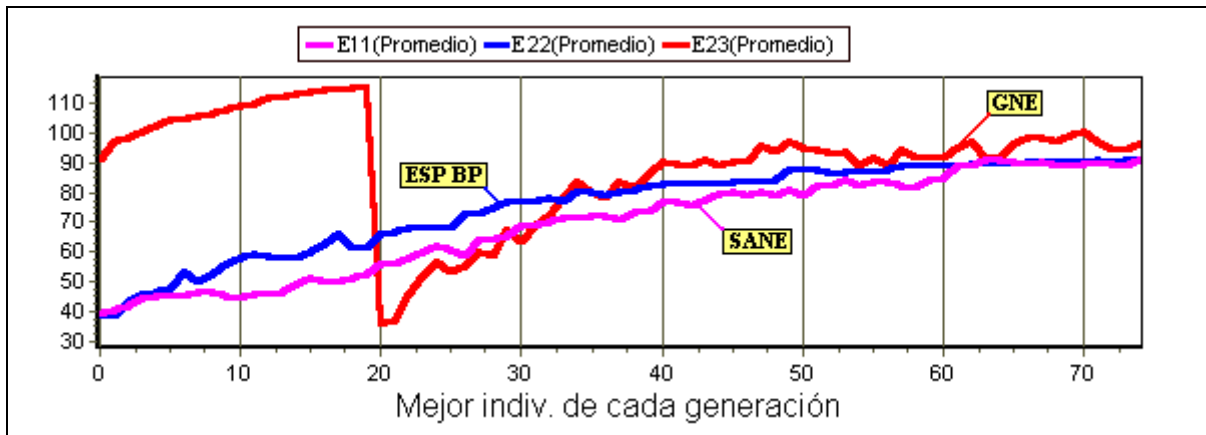


Ilustración 153 Información del fitness promedio alcanzado utilizando SANE, ESP con población de Blueprints y GNE –primera etapa de 20 generaciones–, sobre el escenario C.

La ilustración anterior muestra que para el escenario C la estrategia ESP con población de Blueprints es superior a SANE pues alcanza un mejor fitness promedio en menos generaciones. En la ilustración siguiente se apreciará que además se consiguen individuos de calidad no alcanzada por SANE.

El análisis comparativo que involucre a GNE debe realizarse a partir de la generación número 20 que representa el inicio de la última etapa de método. Se observa claramente las ventajas de utilizar este método en relación a los anteriores. Un rápido crecimiento de la curva GNE hace que a sólo 15 generaciones de iniciada la segunda etapa –generación 35– se supere el fitness promedio de las otras dos estrategias y permanezca así hasta finalizar la evolución.

Veamos qué sucede con estas estrategias al graficar el fitness del mejor individuo por generación seleccionado entre las 30 evoluciones que conforman el ensayo neuroevolutivo.

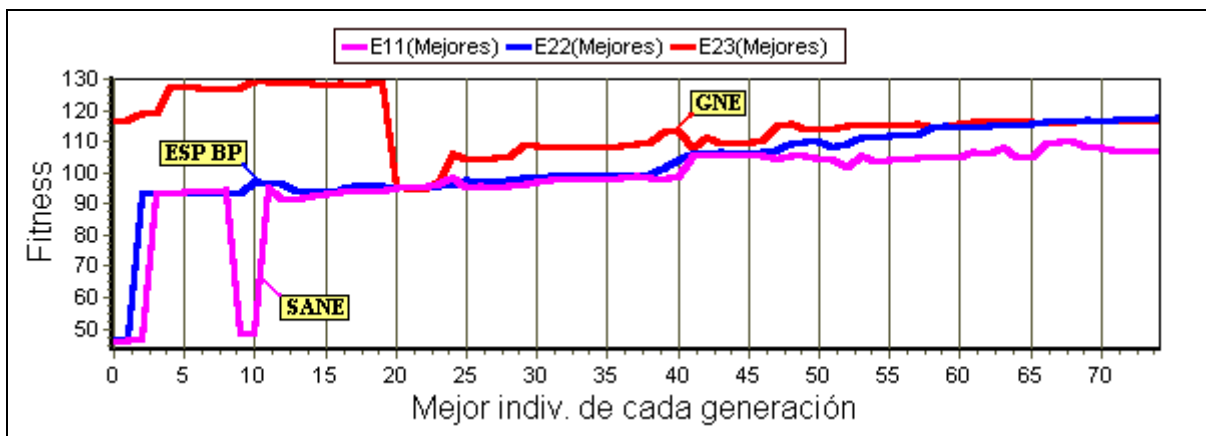


Ilustración 154 Información de los mejores fitness alcanzado durante las 30 evoluciones, utilizando SANE, ESP con población de Blueprints y GNE –primera etapa de 3 generaciones–, sobre el escenario C.

Ya se ha anticipado que ESP con población de Blueprints es capaz de generar individuos de fitness superior, que SANE no puede igualar –al menos, no lo hizo durante las 30 evoluciones que conforman el ensayo–. Similar deducción puede hacerse sobre GNE respecto a las demás estrategias. Puede observarse que GNE siempre conseguirá el individuo de mejor fitness, sólo posiblemente igualado por alguno generado por ESP con población de Blueprints en las generaciones más altas.

Aún falta analizar con qué probabilidad se alcanzará el objetivo al utilizar alguna de estas tres estrategias estudiadas.

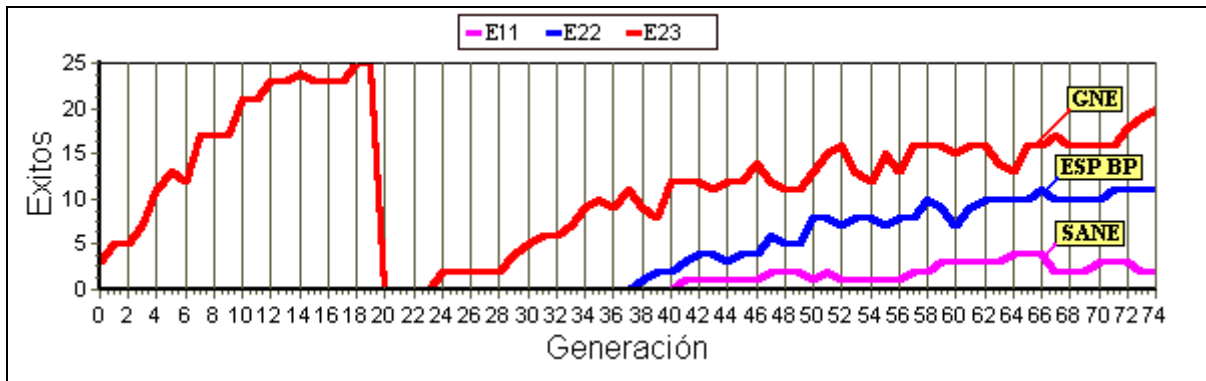


Ilustración 155 Información del total de éxitos alcanzado durante las 30 evoluciones, utilizando SANE, ESP con población de Blueprints y GNE –primera etapa de 20 generaciones–, sobre el escenario C

Nuevamente en la cima del ranking se posiciona GNE, superando cómodamente a ESP con población de Blueprints, quien a su vez, vuelve a superar con igual comodidad a SANE.

Aparentemente la ventaja de ESP con población de Blueprints respecto de SANE y la ventaja de GNE respecto del primero parecen potenciarse con la complejidad del problema a resolver. Obsérvese el rendimiento obtenido sobre el escenario B y compárese con el obtenido sobre el más complejo escenario C. Para confirmar o disconfirmar esta hipótesis se repitieron las pruebas sobre el escenario D de alta complejidad. Las razones que hacen a este escenario difícil de resolver por los agentes aquí implementados, ya han sido debidamente señaladas en este capítulo –ver “Escenario D” pág. 255–.

Pruebas sobre escenario D

Se han realizado 4 ensayos neuroevolutivos sobre el escenario D. La motivación principal de estas pruebas ya se ha expuesto más arriba, pero además se intentará dilucidar una nueva incógnita sobre el método GNE a saber: ¿Es importante para una etapa evolutiva una buena etapa anterior? El interrogante surge porque, precisamente sobre este escenario, se han observado algunas evoluciones como la siguiente.

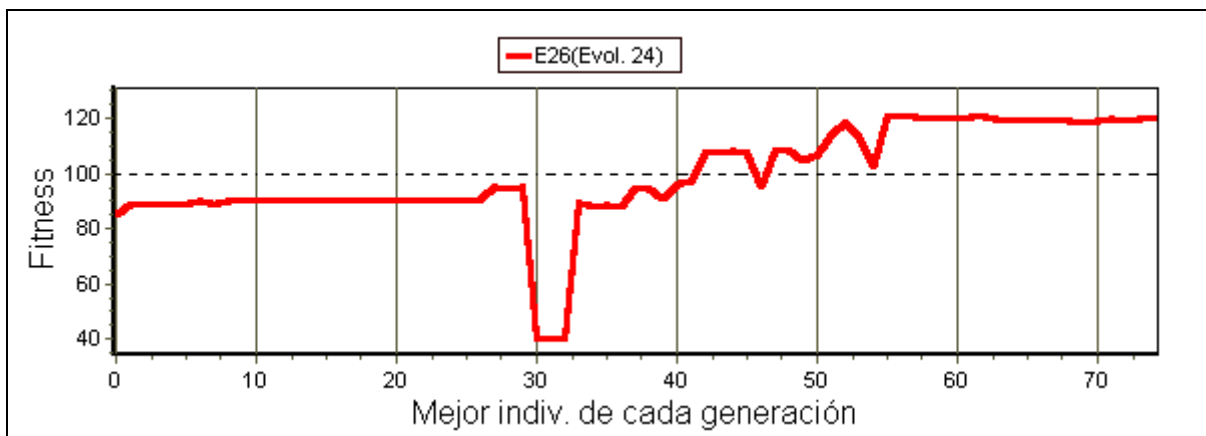


Ilustración 156 Evolución GNE con 30 generaciones para la primera etapa, que evidencia la posibilidad de llegar a la meta en la segunda etapa aún cuando no se alcance en la primera. Recuérdese, que los agentes que logran alcanzar el objetivo reciben un valor de fitness mayor o igual a 100 –línea punteada–.

Obsérvese que aún no habiendo logrado alcanzar el objetivo en la primera etapa –de menor complejidad que la siguiente–, éste ha sido alcanzado en la segunda. Es decir un mal

desempeño de la primera red obtenida, puede ser corregido por la siguiente etapa evolutiva del método.

De esta manera, se han diseñado los siguientes ensayos neuroevolutivos

Ensayo	Características	Motivación
E24	SANE	Comparar entre sí para determinar ventajas y desventajas de estas estrategias en el escenario D de complejidad alta. Analizar el impacto de la primera etapa en la segunda del método GNE.
E25	ESP con población de Blueprints	
E26	GNE, 30 generaciones para la primera etapa	
E27	GNE, 45 generaciones para la primera etapa	

Veamos los resultados obtenidos:

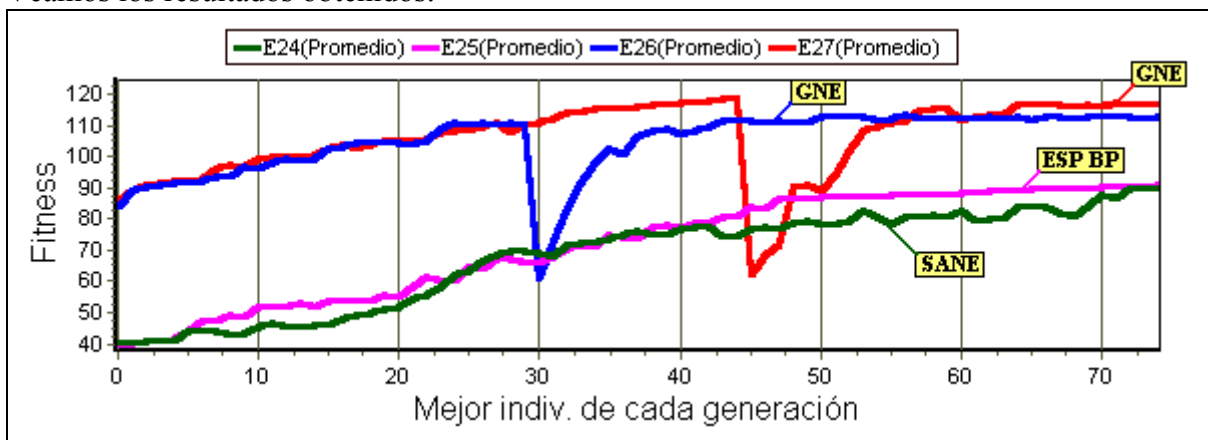


Ilustración 157 Información del fitness promedio alcanzado sobre el escenario D, utilizando SANE, ESP con población de Blueprints y GNE con 30 y con 45 generaciones para la primera etapa.

En la ilustración anterior apreciamos la misma relación que se observó sobre el escenario C. Lo que cambia es la magnitud de la diferencia observada sobre la superioridad del método GNE respecto de los otros dos. Podemos ver también que una primera etapa GNE más larga producirá grupos con primeras redes neuronales de mejor fitness y esto redundará en un mayor rendimiento final.

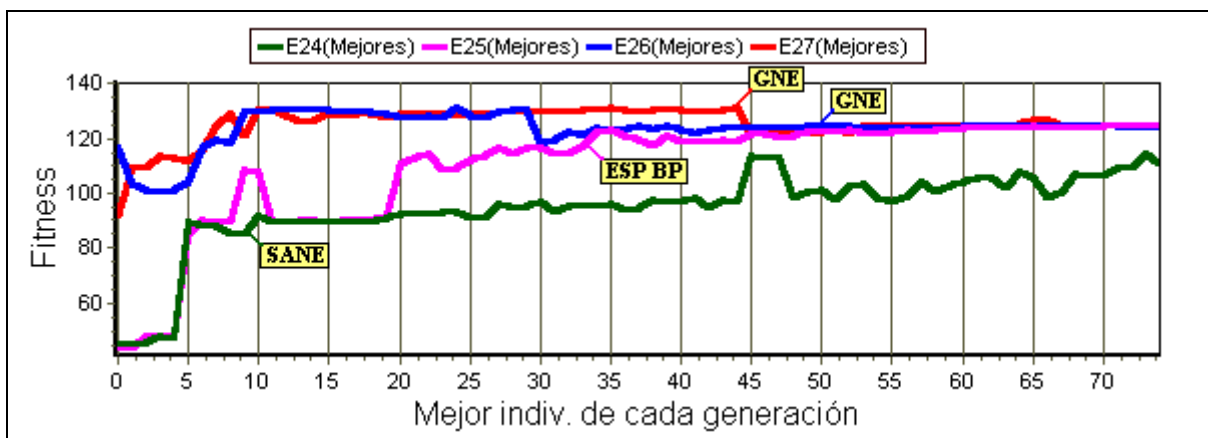


Ilustración 158 Información de los mejores individuos generados para resolver el escenario D, utilizando SANE, ESP con población de Blueprints y GNE con 30 y con 45 generaciones para la primera etapa.

Al igual que lo ocurrido sobre el escenario C, al hablar de los mejores individuos, se debe hablar de GNE y ESP con blueprints, no pudiendo SANE en ningún momento igualar el desempeño de aquellas estrategias. No se aprecian diferencias significativas en la utilización de 30 o 45 generaciones para la primera etapa del método GNE. No obstante, veamos que ocurre con la cantidad de éxitos obtenidos por generación durante las 30 evoluciones.

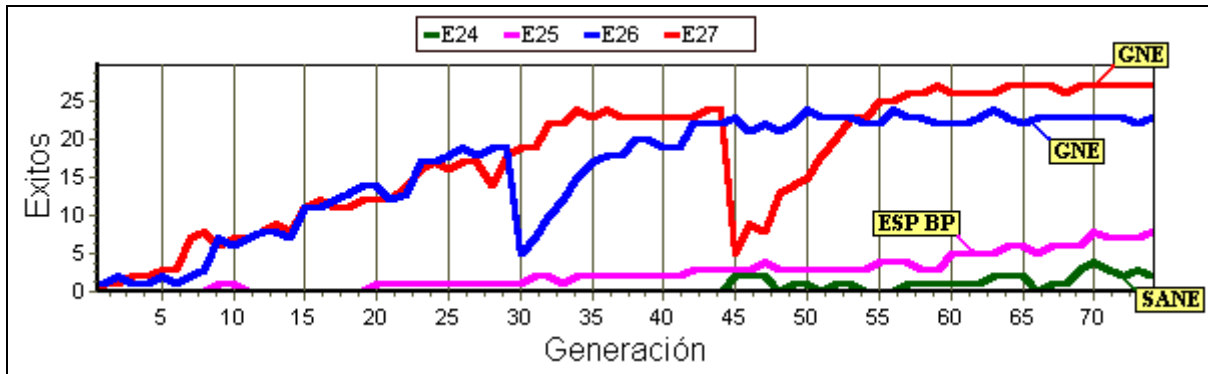


Ilustración 159 Información del total de éxitos logrados sobre el escenario D a lo largo de las 30 evoluciones del ensayo, utilizando SANE, ESP con población de Blueprints y GNE con 30 y con 45 generaciones para la primera etapa.

Nuevamente se repite la relación hallada en las pruebas realizadas sobre el escenario C y al igual que en el caso del promedio del fitness, llama la atención la diferencia existente entre el método GNE y los demás. De la ilustración anterior, también se deduce que la realización de una buena primera etapa acrecienta las probabilidades de alcanzar la meta en la segunda.

Se debe concluir entonces que el método GNE se manifiesta marcadamente superior a los demás cuando el escenario se complica. También se debe admitir que sí es importante una buena etapa evolutiva para mejorar el fitness de la siguiente, aunque el método es capaz de corregir deficiencias de una etapa anterior, se ve claramente que es ventajoso que esta situación no se haga presente.

Se podría dar por concluida aquí la sesión de pruebas y testeos. No obstante aún falta probar la eficiencia del método en un tipo de problema algo distinto a los hasta aquí planteados. Me refiero a un escenario con el objetivo en movimiento. Esto plantea una dificultad de distinta naturaleza que aún no se ha evaluado. Por lo tanto se presentan a continuación los ensayos realizados sobre el escenario E.

Pruebas sobre escenario E (presa en movimiento)

Se han realizado 3 ensayos neuroevolutivos para determinar el rendimiento de las estrategias sobre este escenario.

Ensayo	Características	Motivación
E28	SANE	Comparar entre sí para determinar ventajas y desventajas de estas estrategias sobre un escenario con objetivo en movimiento.
E29	ESP con población de Blueprints	
E30	GNE, 20 generaciones para la primera etapa	

Veamos los resultados obtenidos:

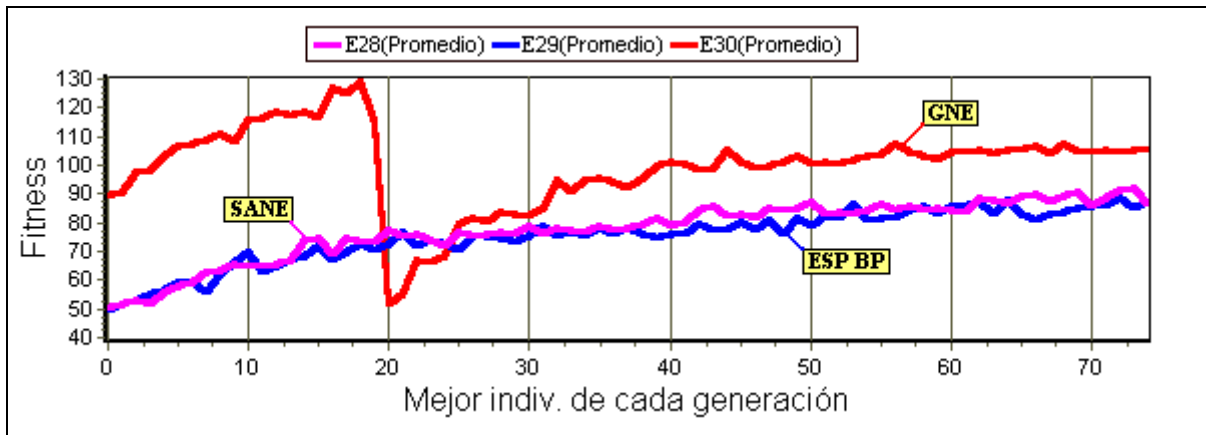


Ilustración 160 Información del fitness promedio alcanzado sobre el escenario E, utilizando SANE, ESP con población de Blueprints y GNE con 20 generaciones para la primera etapa.

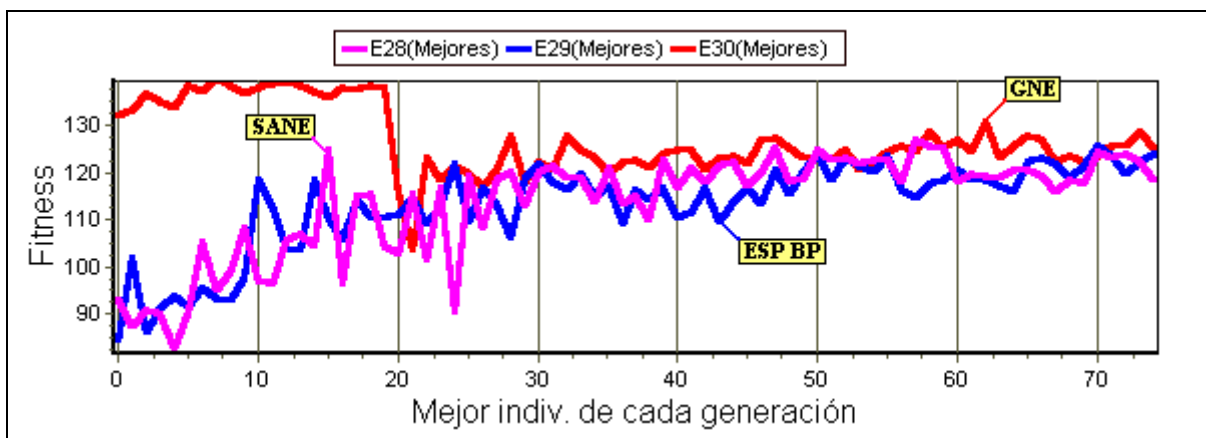


Ilustración 161 Información de los mejores fitness alcanzados sobre el escenario E durante las 30 evoluciones, utilizando SANE, ESP con población de Blueprints y GNE con 20 generaciones para la primera etapa.

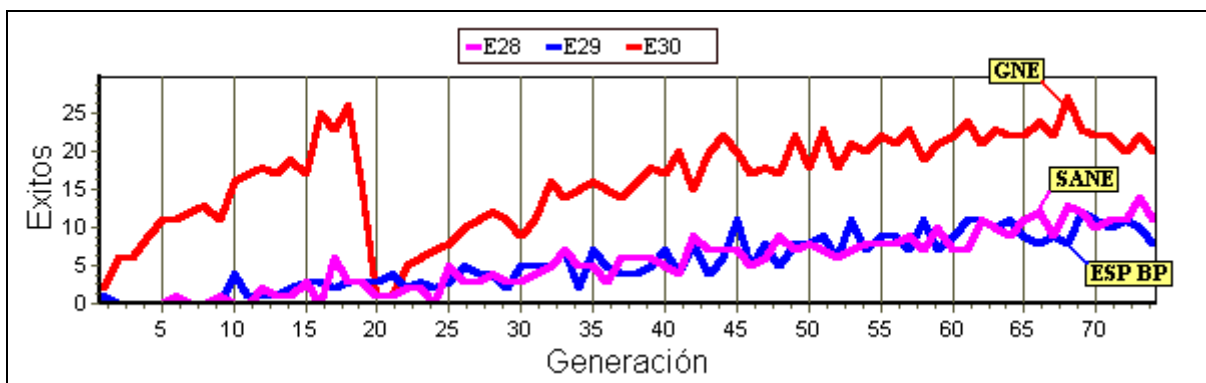


Ilustración 162 Información sobre el total de éxitos obtenidos en el escenario E durante las 30 evoluciones, utilizando SANE, ESP con población de Blueprints y GNE con 20 generaciones para la primera etapa.

En las tres ilustraciones anteriores, puede observarse claramente la superioridad en todos los casos de GNE respecto a los otros dos métodos. Sin embargo para este tipo de problemas de captura de presa ESP con población de Blueprints no parece mejorar el desempeño de SANE comportándose de forma muy similar.

Conclusiones del análisis de los datos

A continuación se resumen las conclusiones extraídas del análisis de los datos presentados anteriormente.

Sobre las arquitecturas neuronales

Se puede concluir que el desempeño de las evoluciones de redes neuronales recurrentes, aventaja ampliamente al evidenciado por las evoluciones de redes feedforward. Esta afirmación se fundamenta en la evidencia presentada sobre la capacidad de las poblaciones de redes recurrentes para evolucionar individuos que resuelven satisfactoriamente el problema planteado en menor cantidad de generaciones –Ilustración 136, Ilustración 138–.

Pero esta no es la única característica beneficiosa. La evolución de redes recurrentes garantiza una mayor probabilidad de alcanzar la meta deseada –Ilustración 139–. Dicho de otra manera, es más resistente a los efectos nocivos de una población inicial “desafortunada” aleatoriamente construida. Estos efectos nocivos, en poblaciones de redes feedforward suelen ocasionar evoluciones con dificultades para progresar hacia fitness más altos.

Debe notarse que si se pretende hallar rápidamente un buen candidato para realizar una tarea dada, será preferible evolucionar redes neuronales recurrentes. No obstante, para el dominio de problemas planteado aquí, si se dispone del tiempo y recursos de cómputo necesarios para extender lo que haga falta el aprendizaje evolucionario, las redes feedforward pueden ser la mejor opción, sobre todo si se intenta minimizar el tiempo de respuesta en la etapa de producción.

La afirmación del párrafo anterior se basa en la evidencia recolectada sobre la posibilidad de obtener redes feedforward –generalmente más veloces que las recurrentes y por lo tanto preferibles en muchos casos–, sin pérdida de calidad, a costa seguramente de repetir un mayor número de veces el proceso evolutivo –Ilustración 141–.

Se debe enfatizar que esta conclusión está referida al dominio de problemas elegido, pues es sabido que en situaciones donde se requiere implementar algún tipo de memoria a corto plazo las redes feedforward, a diferencia de las recurrentes, no tienen ninguna chance de resolverlo. Por lo tanto, lo que debe destacarse es que se ha comprobado que las redes de tipo feedforward, razonablemente menos potentes, han demostrado resolver con igual pericia que las recurrentes la clase de problemas aquí implementados.

Respecto al número de neuronas ocultas a utilizar hemos detectado que en general a mayor cantidad, mejor fitness. Pero esta relación es más evidente en las redes feedforward, donde además mejora sustancialmente la probabilidad de alcanzar la meta –Ilustración 136, Ilustración 137, Ilustración 138, Ilustración 139–. En todos los casos debe existir un compromiso aceptable entre carga de procesamiento y resultado, debiendo recurrirse a la experimentación para aproximar un valor adecuado.

En cuanto a la evolución de función de transferencia –multisigmoide–, no cabe duda que resulta beneficiosa tanto para redes feedforward como recurrentes –Ilustración 142, Ilustración 143, Ilustración 144, Ilustración 145–.

Finalmente, respecto a la conexión de tendencia nada se puede decir de ella tomada en forma aislada, pero combinada con la característica multisigmoide, proveen a la evolución de cierta capacidad de definición para alcanzar la meta deseada que no se aprecia en ausencia de cualquiera de estas dos particularidades –Ilustración 145–.

Sobre las estrategias neuroevolutivas implementadas

Se resumen las conclusiones sobre cada una de las seis estrategias neuroevolutivas implementadas en el presente trabajo de acuerdo a los datos arrojados por los ensayos presentados anteriormente.

Neuroevolución simple

Indudablemente, la peor performance de todas las estrategias testeadas. La evolución se caracteriza por un comportamiento sumamente errático y de lenta convergencia. En el mundo de la Computación Evolutiva estos síntomas suelen tratarse con una cuota de elitismo, y así lo hicimos en el presente trabajo.

Neuroevolución simple con Algoritmo Genético Elitista

Como se podría esperar, el elitismo ha resultado beneficioso para encausar la errática evolución de la estrategia anterior hacia los mejores fitness. No obstante, no deja de ser sorprendente cuán beneficioso ha sido, y cómo una pequeña modificación al algoritmo anterior, consistente en formar una elite con dos copias del mejor individuo para pasarlos directamente a la próxima generación –solo afectados por una posible operación de mutación– impacta tan fuertemente en el desempeño de la evolución –Ilustración 146, Ilustración 147–.

SANE

Concordando con las conclusiones de sus autores [Moriarty96] [Moriarty97_1] [Moriarty97_2], quienes demostraron las ventajas de una coevolución cooperativa en la búsqueda de soluciones a problemas de control, SANE se ha manifestado superior a las dos estrategias tradicionales antes mencionadas –Ilustración 146, Ilustración 147–.

Además, ha resultado preferible a ESP en muchos casos por presentar una mejor relación entre carga computacional y desempeño del proceso evolutivo.

ESP

Ya se ha mencionado que debido a la forma de construir las redes y calcular el fitness ESP necesita evaluar un número mucho mayor de redes neuronales que el tamaño de las subpoblaciones de unidades, para asegurar que cada unidad de éstas tenga chance de participar en varias redes, condición necesaria para una justa asignación del valor de fitness – ver “Debilidad del método” en la pág. 220–.

He evaluado este requisito como un aspecto negativo en el dominio de problemas aquí planteado, donde son factibles soluciones utilizando otras estrategias con poblaciones relativamente pequeñas de 80 redes neuronales.

Recordemos que la fase más "pesada" de la neuroevolución es la evaluación de las redes para la obtención del fitness. Por lo tanto si pretendemos conservar una carga computacional similar a la generada por las demás estrategias no se debe aumentar la cantidad de agentes, sino trabajar con subpoblaciones de unidades sensiblemente más pequeñas.

Los resultados obtenidos utilizando una relación de 1 a 4 entre el tamaño de las subpoblaciones y la cantidad de redes construidas –Ilustración 148, Ilustración 149– revelan que equiparar la carga computacional de ESP con las demás estrategias degrada el rendimiento de este método a niveles muy inferiores al manifestado por SANE, seguramente debido a la imposición de trabajar con subpoblaciones de unidades muy pequeñas –20 individuos–.

Si por el contrario utilizamos ESP con subpoblaciones de un tamaño aceptable -80 unidades- evaluándose 320 redes neuronales construidas a partir de ellas, se observa una mejora considerable respecto del rendimiento mostrado por SANE en las primeras generaciones, pero finalmente, avanzada la evolución SANE alcanza y vuelve a superar a ESP. Pero aún si mejorara sustancialmente la performance respecto de SANE, utilizando por ejemplo una relación 1 a 10 como lo hacen los autores del método en [Gomez97], la gran cantidad de tiempo extra para llevar a cabo una evolución la convierten en una estrategia no atractiva para afrontar el tipo de problemas planteado en este trabajo de grado.

ESP con población de Blueprint

Un problema que no representa grandes retos, será bien resuelto por la mayoría de las estrategias neuroevolutivas. Pero la adversidad, suele hacer claras las diferencias. La ventaja de ESP con población de Blueprints respecto de SANE se evidencia con mayor claridad a medida que el problema a resolver se hace más complejo. Así el fitness promediado de esta estrategia resultó apenas inferior a SANE al resolver el poco complejo escenario B, – Ilustración 150–, pero decididamente mejor en aquellos con mayores dificultades como los escenarios C y D. –Ilustración 153, Ilustración 157–.

Ahora bien, si hablamos del mejor individuo generado debemos remarcar que en todos los casos, sencillos o complejos, ESP con población de blueprint ha superado a SANE – Ilustración 152, Ilustración 154, Ilustración 158–. Esto significa que para un cierto número de procesos evolutivos existe mayor probabilidad de conseguir una mejor red neuronal utilizando ESP con población de blueprint que SANE.

Estas deducciones se aplican a problemas con objetivos estáticos, ya que en el caso de objetivos móviles no se aprecian diferencias de rendimiento entre ESP con población de Blueprints y SANE –Ilustración 160, Ilustración 161, Ilustración 162–.

GNE

Este método se ha mostrado superior a ESP con población de blueprints y a SANE en todos los casos de pruebas testeados, ya sea de menor o mayor complejidad, con objetivo estático o móvil. La única excepción descansa en el buen papel desempeñado por SANE, sobre el escenario B de poca complejidad, que en las últimas generaciones consigue igualar el rendimiento de GNE en fitness promedio y en total de éxitos obtenidos –Ilustración 150, Ilustración 151–.

Aquí se hace muy notorio cómo crece la brecha de rendimiento que separa GNE respecto de la segunda estrategia mejor ranqueada –generalmente ESP con población de Blueprints– conforme se afrontan problemas más complejos, tanto en fitness promedio –Ilustración 150, Ilustración 153, Ilustración 157– como en total de éxitos obtenidos por generación – Ilustración 151, Ilustración 155, Ilustración 159–. Este hecho es relevante pues precisamente se ha desarrollado el método como una estrategia de neuroevolución incremental para la resolución de tareas complejas.

Se ha observado además en todos los casos que GNE siempre conseguirá el individuo de mejor fitness, sólo posiblemente igualado por alguno generado utilizando ESP con población de Blueprints –Ilustración 152, Ilustración 154, Ilustración 158–.

Finalmente también ha demostrado ser el gran vencedor en la resolución de un escenario con objetivo móvil –búsqueda y captura de presa–, ranqueándose en primer lugar en los tres aspectos medidos: fitness promedio, mejor individuo y total de éxitos alcanzados –Ilustración 160, Ilustración 161, Ilustración 162–.

Por último se debe concluir que, si bien el método GNE es capaz de corregir los efectos de una deficiente etapa anterior es deseable que esta situación no se haga presente, pues una buena etapa evolutiva previa favorece un más alto rendimiento –Ilustración 157, Ilustración 159–.

Así finaliza el presente capítulo en el que se intentó identificar ventajas y desventajas de las distintas arquitecturas y métodos de neuroevolución utilizados. En el próximo se presentan las conclusiones finales con las que se da cierre a este trabajo de grado.

Conclusiones Finales

La teoría de la evolución por selección natural presentada por Darwin explica las adaptaciones y toda la diversidad biológica de manera mecánica, algorítmica, sin necesidad de propósito, voluntad o inteligencia alguna.

Con el experimento de Stanley Miller, que logró sintetizar moléculas orgánicas a partir de compuestos inorgánicos, la *continuidad* de la materia quedó demostrada y la frontera entre lo inerte y lo vivo dejó de representar un salto cualitativo como muchos creían en ese entonces.

La materia viva es un sistema organizado de sustancias que por sí mismas no son vivientes. Sin embargo, a partir de un determinado nivel de organización muy especial de tales componentes químicos, aparece la manifestación vital como un estado de actividad física y química, que confiere unas propiedades características, de las que carecen los seres inertes.

La selección natural que propone Darwin, junto con el azar que generó las primeras moléculas autorreplicativas hace unos 4000 millones de años, alcanzan para describir coherentemente la historia de la vida en este planeta, incluyendo sus orígenes, su diversificación, y por supuesto, la aparición del "*Homo sapiens sapiens*", es decir, de nosotros mismos como especie. De esta manera, puede decirse que toda la vida existente sobre la Tierra es el resultado de procesos algorítmicos ciegos. Así, ya no se necesita el protagonismo de un Dios creador y diseñador del universo. Esta es la idea peligrosa de Darwin a la que se refiere el título del libro del filósofo norteamericano Daniel Dennett. Obviamente este resultado es duramente atacado por los creacionistas, sin embargo, la discusión es irrelevante en este contexto.

Pero la teoría de evolución por selección natural, concebida para explicar el origen de las especies, sirvió para su propósito y para algo mucho más importante. Con ella se inició una de las revoluciones del pensamiento más grande de la historia, influyendo considerablemente en el desarrollo de toda la filosofía. No es de extrañar que una teoría tan simple y con gran poder explicativo, haya sido aplicada en contextos ajenos a aquel donde se engendró.

Hoy poseemos una ventaja con la que Darwin no contó. Se conocen los mecanismos inherentes a la transmisión de los caracteres hereditarios. La ciencia de la genética ha avanzado significativamente desde que el monje austríaco Gregor Mendel realizara sus hoy conocidos experimentos de cruzamiento, pero ignorados completamente por sus contemporáneos. Actualmente, la bioquímica y genética que estudian la herencia y la variación, no sólo han aportado nuevas pruebas confirmando que la evolución ha existido, sino también los elementos necesarios para comprender los mecanismos del proceso evolutivo.

Estos conocimientos genéticos encastrados perfectamente con el cuerpo teórico darwiniano suelen denominarse neodarwinismo y nuevamente sus influencias van más allá de los confines de las ciencias biológicas. Por citar un ejemplo, la teoría del famoso biólogo Richard Dawkins, constituye una generalización del evolucionismo para aplicarlo a construcciones culturales. Dawkins sostiene que la cultura evoluciona mediante la supervivencia diferencial de los *memes* –término acuñado por él por analogía a *genes*–, o unidades mínimas de información y replicación cultural como una idea, un concepto, una técnica, una costumbre, etc., que utilizan a nuestras mentes para replicarse y que un medio cultural selecciona en función de su idoneidad para permitir o facilitar el progreso. Esta teoría significa una contribución muy importante a la universalización del evolucionismo, que ahora resulta aplicable a la cultura de igual manera que a la biología. Ha sido criticada por algunos científicos pero considerada seriamente por otros tantos estudiosos, entre ellos Daniel Dennett, quien la utilizó para desarrollar una teoría de la conciencia.

Pero como personas dedicadas a la informática, seguramente nos interese más las consecuencias de la penetración del neodarwinismo –en definitiva, un conjunto de memes con alto nivel de replicación– en los dominios de esta disciplina, siendo la Computación Evolutiva, el producto de esta invasión.

Asumiendo que la solución potencial a un problema puede ser representada por un conjunto de parámetros codificados en una estructura lineal, que adecuadamente recibe el nombre de cromosoma, los algoritmos evolucionarios han demostrado ser eficientes en tareas de búsqueda y optimización. Aquí la selección natural expuesta por Darwin, hará presión para que la evolución genere individuos mejores adaptados, o lo que es lo mismo, soluciones más eficientes.

Aunque atacados desde algún sector de la Inteligencia Artificial, la efectividad en los resultados de los algoritmos genéticos –hoy día los representantes más importantes de la Computación Evolutiva– los han hecho merecedores de reconocimiento como métodos viables para resolver problemas de búsqueda complejos.

El hecho de trabajar sobre la codificación de las posibles soluciones –genotipo– y no sobre las soluciones mismas –fenotipo– facilita su aplicación prácticamente a cualquier problema, explotando una información fácilmente disponible, y al mismo tiempo permitiendo una gran independencia del problema. No obstante, cuestionamientos recientes a la base teórica de los algoritmos genéticos han demostrado que un único algoritmo no puede ser la herramienta perfecta de optimización, contradiciendo la supuesta robustez –efectividad sin perder generalidad– y superioridad de los algoritmos genéticos respecto de otras técnicas de búsqueda.

Independientemente de la justificación formal, creo que los desarrollos en Neuroevolución, e incluso los resultados obtenidos en el presente trabajo, sugieren que no existe un algoritmo genérico óptimo, y siempre será mejor la adaptación al problema de interés. De no ser así, los esfuerzos por probar distintas estrategias neuroevolutivas habrían sido en vano pues ninguna podría superar el rendimiento de una evolución estándar.

La búsqueda de la mejor arquitectura de una red neuronal artificial, o del conjunto de valores de sus pesos para resolver una tarea determinada, constituyen unos de los problemas que la Computación Evolutiva resuelve eficientemente. Estos son los terrenos de la Neuroevolución, donde la utilización de algoritmos evolucionarios da respuesta a algunos inconvenientes propios de las redes neuronales artificiales. Entre otras ventajas, manejan mejor el problema de la búsqueda global en superficies del error complejas, multimodales y no diferenciables, disminuyendo la probabilidad de caer en mínimos locales, son más simples de implementar y pueden hibridarse fácilmente con algoritmos de entrenamiento tradicionales.

La Neuroevolución, al igual que muchos otros paradigmas, presenta debilidades y fortalezas, adecuándose mejor a la realización de ciertas tareas que a otras. Como digno representante del conexionismo, ha demostrado ser prometedora en simulaciones de conductas observadas en animales inferiores, especialmente en el área de reconocimiento de patrones, clasificación y control automático. Sin embargo su actuación ha sido mucho más débil en áreas en las que los sistemas simbólicos –Inteligencia Artificial clásica– han simulado con éxito muchas capacidades del pensamiento humano de alto nivel como por ejemplo el análisis gramatical.

El paradigma de la Neuroevolución es joven y con muchas posibles líneas de investigación. El hecho de encontrar evidencia sobre su capacidad para resolver un gran número de problemas, más eficientemente que otras técnicas, junto a discusiones abiertas con finales no resueltos, lo convierten en un objeto de estudio atractivo y muy interesante.

Puede observarse una actividad considerable en el área, con congresos y material publicado todos los años. Es frecuente toparse con artículos que proponen alguna nueva estrategia neuroevolutiva, o extensión de una existente, argumentando mejoras y beneficios respecto de otras. No obstante, me ha llamado la atención una serie de discusiones, de idas y venidas, de críticas y aprobaciones, de opiniones distintas, a veces contradictorias, y la falta de resultados contundentes que declaren definitivamente una postura superior a la otra. Todo ello, sumado a la histórica contienda entre el simbolismo y el subsimbolismo hacen de la Neuroevolución una interesante área de investigación.

Entre las disputas no resueltas, se encuentran las heredadas de la Computación Evolutiva. Así existen diferentes consideraciones sobre la importancia de los dos operadores principales: el crossover y la mutación. Mientras que los Algoritmos Genéticos usan al primero como base para la evolución, otros prefieren la llamada evolución primitiva, basada únicamente en mutación. También hay contradicciones sobre la codificación genética, Holland y Goldberg aconsejan utilizar un alfabeto de baja cardinalidad, en lo posible binario. No obstante, los recientes cuestionamientos al teorema de esquemas y la evidencia empírica podrían inclinar la tendencia hacia otras representaciones, por lo menos en algunos casos concretos. El uso de alfabetos binarios puede ser desventajoso en problemas de alta dimensionalidad produciendo cromosomas extremadamente largos que perjudican la evolución.

Pero además la Neuroevolución es generadora de sus propias discusiones aún sin resolver. Existen investigadores que han presentado pruebas empíricas en donde el entrenamiento evolucionario puede ser significativamente más rápido y confiable que el Backpropagation. Sin embargo otros investigadores han presentado resultados donde las variantes más rápidas de Backpropagation son insuperables aún por hibridaciones AG-BP –una técnica que corre un Algoritmo Genético primero y luego un Backpropagation–. Sin embargo otros artículos reportan excelentes resultados utilizando hibridación de entrenamiento evolucionario y algoritmos de gradiente descendente. Estas discrepancias señalan que no existe todavía un claro ganador en términos del mejor algoritmo de entrenamiento, sino que sus desempeños se ven afectados por la naturaleza del problema.

También existen contradicciones sobre la importancia del *problema de permutación* que surge de la equivalencia funcional entre distintas redes creadas por permutaciones de sus nodos. Esto causa el hecho indeseable que genotipos distintos produzcan fenotipos que funcionan de igual forma. Según algunos investigadores el problema de permutación disminuye la eficiencia del crossover llevando a varios a abandonarlo como operador genético. Sin embargo hay quienes sostienen que el problema de permutación no es tan severo como se ha supuesto.

Pero existe aún más controversia respecto al uso del crossover en la Neuroevolución. Mientras que por un lado se ha mostrado que este operador genético puede ser útil e importante para incrementar la eficiencia de la evolución para algunos problemas, por otro lado su utilización está cuestionada por algunos investigadores. La razón de este cuestionamiento descansa en la representación distribuida del conocimiento que mantienen las redes neuronales, lo que sería incompatible con la existencia de buenos bloques constructores en el genotipo –necesarios para el buen rendimiento del crossover–. Así la recombinación de una parte de la red con otra parte de otra es probable que destruya ambas.

En definitiva, existen muchos aspectos aún por resolver, discusiones por clarificar y líneas abiertas de estudio en neuroevolución. Pero no solamente hace falta realizar investigación sobre este paradigma, sino también, sobre la forma de articular con la Inteligencia Artificial simbólica, o clásica. Algunos autores, incluido Marvin Minsky sostienen que la controversia entre los enfoques top-down y bottom-up se debe al deseo de ambas comunidades de

Inteligencia Artificial de alcanzar una teoría que sea pequeña, elegante, potente y correcta para explicar la inteligencia en su totalidad, algo así como ocurre con la física, que se concentra en minimizar la cantidad y variedad de sus afirmaciones. Pero esto no ha sido logrado por ninguna de las dos corrientes. Por el contrario ambas han demostrado tener limitaciones serias. Ni los sistemas puramente conexionistas ni los puramente simbólicos parecen aptos para resolver el tipo de comportamiento intelectual de niños pequeños. Seguir dividiendo las aguas puede resultar incorrecto, quizá deba invertirse en investigación para construir un puente satisfactorio entre los logros de ambos enfoques.

Ya en el campo de lo particular, respondiendo a mis propias expectativas detalladas en la introducción del presente trabajo debo remarcar algunos resultados concretos.

- Creo haber presentado a lo largo de al menos dos capítulos, un conjunto de conocimientos interesantes, extrainformáticos, en relación a los principios inspiradores de la Computación Evolutiva y la Neurocomputación. No he dedicado poco tiempo a ello, intentando cubrir aspectos que la literatura informática suele tratar muy ligeramente.
- En cuanto a la clase genérica de problemas denominada “*sequential decision task*”, toda la experimentación presentada en este trabajo constituye un claro ejemplo de la viabilidad de la utilización de técnicas neuroevolutivas para afrontar este tipo de situaciones.
- La experimentación con distintas arquitecturas ha arrojado como resultado que la evolución de redes neuronales recurrentes es más rápida que las de redes feedforward y garantiza una mayor probabilidad de alcanzar la meta. No obstante, no son más potentes para el tipo de problemas aquí resueltos, siendo posible conseguir redes feedforward de igual rendimiento, sólo que debe invertirse mayor tiempo para ello. Por otro lado, la evolución de función de transferencia ha resultado beneficiosa, más aún si se realiza sobre arquitecturas con conexión de tendencia.
- De las pruebas realizadas para comparar el desempeño de las distintas estrategias neuroevolutivas se ha verificado claramente que SANE se desempeña mejor que las estrategias tradicionales y ofrece una mejor relación de fitness sobre carga de procesamiento que ESP. Sin embargo, ESP con población de Blueprint, ha superado el rendimiento de SANE en aquellos problemas de mayor complejidad. Por último, GNE ha mostrado mejor rendimiento que las demás estrategias en todos los casos de prueba, no sólo consiguiendo mejores soluciones, sino también en menor cantidad de generaciones.
- El mejor rendimiento evidenciado por GNE pone de manifiesto que simular el aprendizaje incremental y la organización funcional cerebral, por medio de grupos de redes neuronales artificiales que se organizan para resolver conjuntamente una tarea determinada, es útil y debe ser explotado.
- El rendimiento de una red neuronal que resuelve satisfactoriamente un determinado problema, generalmente se degrada ante el cambio repentino de una o más variables del entorno. Habitualmente se corre un nuevo algoritmo evolutivo o de entrenamiento para conseguir otra red que resuelva el problema bajo las nuevas condiciones. Se desaprovecha así, el conocimiento valioso que la red considerada obsoleta aún posee. Basándome en la naturaleza de estrategia incremental de GNE, y en el buen rendimiento evidenciado para

hallar una nueva red que complemente la funcionalidad de otra –u otras–, creo que existen suficientes indicios que marcan su potencialidad para afrontar esta clase de transiciones mucho más eficientemente.

Finalmente, creo que los resultados obtenidos en el presente trabajo alcanzan para considerar a los grupos neuronales como una alternativa viable, y muchas veces ventajosa para la resolución de ciertos problemas complejos.

Es posible que GNE muestre también un buen rendimiento en otra clase de problemas distinta a los estudiados aquí. Por ejemplo, en aquellos típicos de aprendizaje supervisado como una aplicación de reconocimiento óptico de caracteres. Quizá una organización funcional donde distintas redes se especialicen en el reconocimiento de subconjuntos de símbolos aventaje al enfoque tradicional en que todo el trabajo lo realiza una sola red neuronal. Pero esto aún necesita de la comprobación empírica.

Considero que existen muchos aspectos en relación a los grupos neuronales para investigar. Entre otras cosas pueden estudiarse distintas estrategias en la delegación de control dentro del grupo, determinación automática de etapas y objetivos incrementales, evolución simultánea de todas las redes integrantes y la aplicación del método a distintas clases de problemas.

Aquí finaliza el presente trabajo de grado. Deseo haber conseguido exponer de forma clara y precisa tanto los conceptos técnicos de los desarrollos presentados como los de carácter informativo provenientes de otras disciplinas distintas de la Informática. Pero además, creo haber presentado un conjunto interesante de resultados junto a una nueva estrategia neuroevolutiva de buen rendimiento y perspectiva futura que debe animar a realizar estudios en esa dirección.

Referencias

Libros y enciclopedias

- [Castrillon95] Castrillon, Laura Viviana. "Memoria Natural y Artificial". Impresora y Encuadernadora Progreso, S. A. de C. V. (IEPSA), Calz de San Lorenzo, 244; 09830 México, D. F. Octubre de 1995.
- [COMBI] Enciclopedia COMBI VISUAL,. Ediciones DANA E. Barcelona.1976. Tomo 2. Capítulo: "Evolución"
- [Diaz97] Diaz, José Luis. "El Ábaco, la Lira y la Rosa. Las Regiones del Conocimiento". Impresora y Encuadernadora Progreso, S.A. (IEPSA), calzada de San Lorenzo 244, 09830 México, D.F. Febrero de 1997.
- [ECNMAE] Enciclopedia de Ciencias Naturales Medio Ambiente y Ecología. Oriente S.A. Barcelona 1997. Tomo 2. Capítulo: "Zoología"
- [Encarta98] Enciclopedia Microsoft® Encarta® 98 © 1993–1997 Microsoft Corporation. Artículos: "Evolución", "Darwin, Charles Robert", "Genética", "Célula".
- [Freeman93] Freeman, James A. & Skapura, David M. "Redes neuronales Algoritmos, aplicaciones y técnicas de programación". Addison–Wesley, 1991. Versión en español de: Rafael García -Bermejo Giner. Addison–Wesley Iberoamericana 1993.
- [Goldberg89] Goldberg, David.E. "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison–Wesley, 1989.
- [Hecht-Nielsen91] Hecht-Nielsen, Robert. "Neurocomputing". Addison-Wesley Publishing Company.1991
- [Hilera95] Hilera González, J. y Martínez Hernando, V. "Redes neuronales artificiales: fundamentos, modelos y aplicaciones". Madrid : RA-MA. 1995.
- [KVITCA] Kvitca, Adolfo M. "Resolución de Problemas con Inteligencia Artificial". Editorial KAPELUZ.
- [Pasantes97] Pasantes, Hermina. "De Neuronas, Emociones Y Motivaciones". Impresora y Encuadernadora Progreso, S. A. de C. V. (IEPSA), Calz de San Lorenzo, 244; 09830 México, D. F. Agosto de 1997.
- [Sagan_1] Sagan, Carl. "COSMOS" © Carl Sagan Productions, Inc.,1980. Traducción castellana de Miguel Muntaner ©Editorial Planeta S.A.,1998. Colección: La Línea del Horizonte. 16ª edición. Capítulo "Una voz en la fuga cósmica".
- [Sagan_2] Sagan, Carl. "THE DRAGONS OF EDEN". Random House, Nueva york. Traducción castellana de Rafael Andreu. Capítulo: "Genes y cerebro".
- [Tapia96] Tapia, Ricardo "Las Células de la Mente". Impresora y Encuadernadora Progreso, S. A. de C. V. (IEPSA), Calz. de San Lorenzo, 244; 09830 México, D.F. Diciembre de 1996

Papers y artículos

- [Balakrishnan95] Karthik Balakrishnan and Vasant Honavar. "Evolutionary Design of Neural Architectures. A Preliminary Taxonomy and Guide to Literature". Artificial Intelligence

Research Group. Department of Computer Science. Iowa State University. Ames, Iowa 50011-1040, U.S.A.

- [Barros] Barros, Patricio. “¿La Física? ¡Pero si es muy fácil!” Reproducción digital del libro “Física recreativa” de Yakov Perelman. 13ª edición. Editorial MIR, Moscú, 1968. Publicado en: [ESC_FISICA] <http://www.efis.ucr.ac.cr/educacion/>
- [Beasley93] Beasley, David.; Bull David R.; Martin, Ralph R. “An Overview of Genetic Algorithms” Department of Computing Mathematics, University of Cardiff; Department of Electrical and Electronic Engineering, University of Bristol. University Computing, 1993 15(2) 58-59.
- [Bestoso] Bestoso, Emilio Santiago. “El Cerebro: Enigma de la Evolución”. Revista electrónica del Hospital Escuela “General José Francisco de San Martín” Corrientes Argentina. URL: <http://www.corrientes.com.ar/instituciones/Hospitalescuela/revista.htm/>
- [Bruce01] Joseph Bruce and Risto Miikkulainen. “Evolving Populations Of Expert Neural Networks” Department of Computer Sciences, The University of Texas at Austin To appear in Proceedings of the Genetic and Evolutionary Computation Conference. (GECCO-2001, San Francisco, CA), 251--257. San Francisco, CA: Kaufmann, 2001.
- [Cárdenas] Cárdenas Fernando; Lamprea Marisol. “El Cerebro: aquella inestable matriz”. Departamento de Psicobiología, Universidad de São Paulo -Brasil. Publicado en [PSYCO]
- [Cases] Cases, Blanca. “VIDA ARTIFICIAL Un enfoque desde la Informática Teórica”. Dpto. de lenguajes y Sistemas Informáticos de la Universidad del País Vasco. Publicado en URL: <http://www.geocities.com/CapeCanaveral/9802/3d4ca000.htm>
- [Castillo] P.A.Castillo, J Carpio, J. J. Merelo, A. Prieto, V. Rivas, G. Romero. “Evolving Multilayer Perceptrons”. Department of Architecture and Computer Technology University of Granada (Spain). Department of Computer Science University of Jaen (Spain).
- [Coello97] Coello Coello, Carlos A. “La Computación Evolutiva en el Contexto de la Inteligencia Artificial”. Investigador Titular del LANIA. Publicado en [LANIA] 1997.
- [CottaPorras98] Cotta Porras, Carlos. Tesis doctoral “Un Estudio de las Técnicas de Hibridación y su Aplicación al Diseño de Algoritmos Evolutivos”. Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga. 28 de Mayo de 1998.
- [FloresMendez99] Flores Méndez, Alejandro. Gómez Ramírez, Eduardo. “Las Redes Neuronales Celulares y su aplicación en el Procesamiento de Imágenes utilizando MATLAB”. Laboratorio de Investigación y Desarrollo de Tecnología Avanzada UNIVERSIDAD LA SALLE. México, D.F., México. Publicado en IV Simposio Iberoamericano de Reconocimiento de Patrones. La Habana, CUBA, Marzo 21-26 de 1999.
- [Gascón] Manuel de la Herrán Gascón “Notas sobre computación evolutiva”, publicado en el sitio web [GAIA]
- [Gascón98] Manuel de la Herrán Gascón “Computación evolutiva” Publicado en marzo de 1998 en la revista "Solo Programadores" (<http://www.towercom.es>) nº 43 y revisado en abril de 1999 por el autor. Extraído del sitio web [REDCIENT].

- [Gomez97] Faustino Gomez and Risto Miikkulainen. “Incremental Evolution Of Complex General Behavior” Department of Computer Sciences, The University of Texas at Austin. Adaptive Behavior, 5:317-342, 1997.
- [Gomez99] Faustino Gomez and Risto Miikkulainen. “Solving non-markovian Control Tasks with Neuroevolution” Department of Computer Sciences, The University of Texas at Austin. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99, Stockholm, Sweden), 1356-1361. San Francisco, CA: Kaufmann. 1999.
- [GómezMartínez97] Gómez Martínez, Mario. “Redes Neuronales Artificiales” Contenido de los seminarios sobre redes neuronales artificiales dictados en San Nicolás y en Rosario. Universidad Tecnológica Nacional Facultad Regional San Nicolás Secretaría de Ciencia y Tecnología Grupo Ingeniería del Conocimiento. 1997.
- [Gutierrez00_1] Gutiérrez, Claudio. “La coevolución del cerebro y el lenguaje dentro de una perspectiva humanista”. Lección conmemorativa del 25 aniversario del Centro Regional de Limón de la Universidad de Costa Rica, 12 de mayo de 2000. Publicado en: [GUTIERREZ] http://claudiogutierrez.com/Coevolucion_cerebro_lenguaje.html
- [Gutierrez00_2] Gutiérrez, Claudio. “El genoma humano”. 2000. Publicado en [GUTIERREZ]: http://claudiogutierrez.com/El_genoma_humano.html
- [Gutierrez01] Gutiérrez, Claudio. “Larga marcha de la vida”. 2001. Publicado en [GUTIERREZ]: http://claudiogutierrez.com/La_larga_marcha_de_la_vida.html
- [Gutierrez95] Gutiérrez, Claudio. “Lo que la neurología puede decirle a la Inteligencia Artificial”. Outline de conferencia al grupo de Inteligencia Artificial en la Universidad de Delaware. 1995. Publicado en [GUTIERREZ]: http://claudiogutierrez.com/Lo_que_IA_puede_aprender.html
- [Gutierrez97_1] Gutiérrez, Claudio. “La selección natural”. 1997. Publicado en [GUTIERREZ]: http://claudiogutierrez.com/Lo_que_IA_puede_aprender.html
- [Gutierrez97_2] Gutiérrez, Claudio. “El algoritmo informático”. 1997. Publicado en [GUTIERREZ]: http://claudiogutierrez.com/El_algoritmo_informatico.html
- [HanYong01] Chern Han Yong and Risto Miikkulainen. “Coooperative Coevolution of Multi-Agent Systmes” Department of Computer Sciences, The University of Texas at Austin. Technical Report AI-01-287, February 2001.
- [Larrañaga] Larrañaga, Pedro. “Algoritmos Genéticos” Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco.
- [LopezBlanco97] Lopez Blanco, Myriam. “La mente sólo es cerebro”. Suplemento de Salud de Elmundo.es 20 de Marzo de 1997.
- [Merelo97] Merelo Guervós, Juan Julián. “Informática Evolutiva”. Tutorial de informática evolutiva impartido como curso de doctorado de 30 horas en la Universidad del País Vasco en mayo de 1997. Publicado en: <http://geneura.ugr.es/~jmerelo/ie/>
- [Minsky] Marvin Minsky. “Simbolismo versus Conexionismo”. Recopilado en Antología de Epistemología e Informática. (compilada por Claudio Gutiérrez) URL: <http://claudiogutierrez.com/bid-fod-uned/>
- [Moriarty96] Moriarty David Eric & Miikkulainen Risto. “Efficient Reinforcement Learning through Symbiotic Evolution”. Department of Computer Sciences, The University of Texas at Austin. Austin, TX 78712. Machine Learning, 22,11-33 (1996) Editor:Leslie Pack Kaelbling, 1996.

[Moriarty97_1] Moriarty David Eric. "Symbiotic Evolution of Neural Networks in Sequential Decision Tasks". Department of Computer Sciences, The University of Texas at Austin Ph.D. Dissertation; Technical Report AI97-257, January 1997.

[Moriarty97_2] Moriarty David E. & Miikkulainen Risto. "Forming Neural Networks Through Efficient and Adaptive Coevolution". Information Sciences Institute, University of Southern California. Department of Computer Sciences, The University of Texas at Austin. Evolutionary Computation 5:373-399, 1997.

[Palmer2001] Palmer, A., Montaña, J.J. y Jiménez, R. "Tutorial sobre Redes Neuronales Artificiales: El Perceptrón Multicapa" Área de Metodología de las Ciencias del Comportamiento. Facultad de Psicología. Universitat de les Illes Balears. Publicado en Psicología.com REVISTA ELECTRÓNICA DE PSICOLOGÍA. Vol. 5, No. 2, Julio 2001. ISSN 1137-8492. <http://www.psiquiatria.com/psicologia/revista/61/>

[PerezBergquist01] Andrés Santiago Pérez-Bergquist "Applying ESP and Region Specialists to Neuro-Evolution For Go" Department of Computer Sciences, The University of Texas at Austin. Honors Thesis, Technical Report CSTR01-24, May 2001.

[Rodríguez96] Roríguez R. Rodolfo J. "Modelos Cognoscitivos Para La Filosofía Contemporanea De La Mente". Revista de Filosofía de la Universidad de Costa Rica. XXXIV, 423-432. 1996.

[XinYao96] Xin Yao and Yong Liu. "Ensemble Structure of Evolutionary Artificial Neural networks". Computational intelligence Group, School of Computer Science University College, The University of new South Wales. Australian Defence Force Academy, Canberra, ACT, Australia 2600. 1996.

[XinYao99] Xin Yao. "Evolving Artificial Neural networks". School of Computer Science The University of Birmingham Edgbaston, Birmingham B15 2TT. Proceedings of the IEEE, 87(9):1423-1447, September 1999.

Sitios Webs

[ACIENT] Aula Científica Terra URL: http://www.terra.es/ciencia/aula_cientifica/portada.htm/. Artículo: "Curso de genética".

[AEI] Antología de Epistemología e Informática (compilada por Claudio Gutiérrez) URL: <http://claudiogutierrez.com/bid-fod-uned/>

[ALDEAEDU] AldeaEducativa.com. URL: <http://www.aldeaeducativa.com>. Artículo: "Siguen las dudas sobre el número de genes del ser humano" (publicado el 27/08/2001).

[AVB] Aula Virtual de Biología de la Universidad de Murcia. URL: <http://www.um.es/~molecula/indice.htm/>. Capítulos: "Ácidos nucleicos", "Duplicación, Transcripción y Traducción".

[ENCICLONET] Enciclopedia en línea EncicloneT. URL: <http://www.encyclonet.com/>. Artículos: "Evolución biológica", "Genética mendeleana", "Genética Molecular", "Meiosis", "Herencia biológica", "Mutación", "Replicación del ADN".

[ESC_FISICA] Escuela de Física. Universidad de Costa Rica. URL: <http://www.efis.ucr.ac.cr/>

[GAIA] Artificial LifeGenetic Algorithms Learning Machine URL: <http://www.aircenter.net/gaia/index.html>

[GUTIERREZ] Obras selectas en línea. Claudio Gutierrez, Universidad de Costa Rica URL:
<http://claudiogutierrez.com/>

[LANIA] Laboratorio Nacional de Informática Avanzada URL: <http://www.lania.mx/>

[PSYCO] Psycología.com Psicología Científica en Internet URL: <http://www.psycologia.com/>

[REDCIENT] Red Científica. URL: <http://www.redcientifica.com/>. Artículos sobre
Inteligencia Artificial.

Apéndice A. Ensayos realizados

E01

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 150

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,3,2) FeedFordward, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 12
Precisión para la codificación de pesos: 16 bits

E02

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 300

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,6,2) FeedFordward, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 12
Precisión para la codificación de pesos: 16 bits

E03

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 500

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,10,2) FeedFordward, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 12
Precisión para la codificación de pesos: 16 bits

E04

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 250

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,3,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 15
Precisión para la codificación de pesos: 16 bits

E05

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 400

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E06

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 600

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,10,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 22
Precisión para la codificación de pesos: 16 bits

E07

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 200
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 500

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,10,2) FeedForward, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 22
Precisión para la codificación de pesos: 16 bits

E08

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 400

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Sin Término de tendencia, No Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E09

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 400

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, No Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E10

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 50
Tamaño de la población de Unidades: 400

SIMULACIÓN:

Pasos de simulación: 500
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: A.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Sin Término de tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E11

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: C.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E12

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: C.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Sin Término de tendencia, No Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E13

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: C.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, No Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E14

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: C.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Sin Término de tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E15

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: NE. Simple
Cantidad de evoluciones: 30
Cantidad de generaciones: 60
Tamaño de la población de Agentes: 80
Prob. de crossover: 0,8 Prob. de mutación: 0,001

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E16

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: NE. Simple Elitista
Cantidad de evoluciones: 30
Cantidad de generaciones: 60
Tamaño de la población de Agentes: 80
Prob. de crossover: 0,8 Prob. de mutación: 0,001

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E17

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 60
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E18

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: ESP Clásico
Cantidad de evoluciones: 30
Cantidad de generaciones: 60
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 20

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E19

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: ESP Clásico
Cantidad de evoluciones: 30
Cantidad de generaciones: 60
Tamaño de la población de Agentes: 320
Tamaño de la población de Unidades: 80

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E20

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: ESP con Blueprints
Cantidad de evoluciones: 30
Cantidad de generaciones: 60
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 80

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E21

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: GNE
Cantidad de evoluciones: 30
Cantidad de generaciones: 60 (3 para la primera etapa)
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700 (700 para la primera etapa)
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: B.esc (Partida desde SubMeta25 para la primera etapa)

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E22

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: ESP con Blueprints
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 80

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: C.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E23

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: GNE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75 (20 para la primera etapa)
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700 (700 para la primera etapa)
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: C.esc (Partida desde SubMeta25 para la primera etapa)

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E24

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: D.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E25

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: ESP con Blueprints
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 80

SIMULACIÓN:

Pasos de simulación: 700
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: D.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E26

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: GNE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75 (30 para la primera etapa)
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700 (700 para la primera etapa)
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: D.esc (Partida desde SubMeta25 para la primera etapa)

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E27

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: GNE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75 (45 para la primera etapa)
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 700 (700 para la primera etapa)
Cantidad de objetivos: 1 (estático)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: D.esc (Partida desde SubMeta25 para la primera etapa)

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E28

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: SANE
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 1000
Cantidad de objetivos: 1 (en movimiento)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: E.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E29

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: ESP con Blueprints
Cantidad de evoluciones: 30
Cantidad de generaciones: 75
Tamaño de la población de Agentes: 80
Tamaño de la población de Unidades: 80

SIMULACIÓN:

Pasos de simulación: 1000
Cantidad de objetivos: 1 (en movimiento)
Tamaño de Agente: 6
Tamaño de Objetivos: 6
Escenario: E.esc

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide
Cantidad de conexiones por neurona: 18
Precisión para la codificación de pesos: 16 bits

E30

NEURO-EVOLUCIÓN:

Clase de Neuroevolución: GNE

Cantidad de evoluciones: 30

Cantidad de generaciones: 75 (20 para la primera etapa)

Tamaño de la población de Agentes: 80

Tamaño de la población de Unidades: 640

SIMULACIÓN:

Pasos de simulación: 1000 (400 para la primera etapa)

Cantidad de objetivos: 1 (en movimiento)

Tamaño de Agente: 6

Tamaño de Objetivos: 6

Escenario: E.esc (Partida desde SubMeta50 para la primera etapa)

ARQUITECTURA RNA:

RN=(10,6,2) Recursiva, Con Tendencia, Multisigmoide

Cantidad de conexiones por neurona: 18

Precisión para la codificación de pesos: 16 bits