

APLICACIÓN DE COMITÉS DE EXPERTOS EN EL MODELADO DEL COMPORTAMIENTO HUMANO EN SITUACIONES DE DISTORSIÓN VISUO-MOTORA

Antonio Guerrero González
Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Cartagena, Antonio.Guerrero@ucpt.es

Juan Luis Pedreño Molina
Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Cartagena, Juan.PMolina@ucpt.es

Juan López Coronado
Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Cartagena, JI.Coronado@ucpt.es

José Luis Contreras Vidal
Department of Kinesiology and Neuroscience and Cognitive Science Program
University of Maryland, College Park, MD 20742-2611, USA, pepeum@wam.umd.edu

Antonio Candel Ruiz
Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Cartagena

Javier Molina Villaplana
Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Cartagena, Javier.molina@upct.es

Resumen

En este trabajo se presenta una aplicación de las arquitecturas neuronales basadas en comités de expertos en donde dan una buena solución en un problema de modelado del comportamiento visuo-motor humano. La intención de este artículo ha sido la de mostrar la validez de las redes en comités sobre una aplicación de modelado de sistemas biológicos. Se trata por tanto de mostrar un modelo del comportamiento psicofísico humano, es decir, del comportamiento de las arquitecturas neuronales biológicas mediante redes neuronales artificiales en donde las redes en comité de expertos forman parte de la arquitectura global del modelo.

Palabras Clave: redes neuronales, comités de expertos, transformaciones visuo-motoras.

1 INTRODUCCIÓN

El desarrollo de los sistemas en redes neuronales ha permitido afrontar con éxito problemas de

clasificación e interpolación de funciones¹. Son múltiples las arquitecturas neuronales y modelos de aprendizaje que permiten abordar estos problemas. Entre otras configuraciones, podemos destacar:

- Redes MLP.
- Redes RBF, para la interpolación de funciones no lineales.
- Redes SOM, que resuelven tareas de *clustering*.
- Redes RNN.

Empleando configuraciones clásicas, la labor de obtener la relación entre los espacios entrada y salida se lleva a cabo como un problema único. Sin embargo, cabe la posibilidad de descomponer la tarea global en una serie de tareas más simples, de manera que la solución del problema complejo se obtenga como una determinada combinación de las soluciones de los problemas simples. En los problemas que nos ocupan, esto se traduce en dividir el espacio entrada en diferentes subespacios o regiones.

¹ En lo que sigue, problema de regresión

El equivalente en redes neuronales al principio “divide y vencerás” son arquitecturas que aúnan varias redes (expertos), cada una correspondiente a uno de estos subespacios. Son los denominados sistemas en comité (*committee machines*) [4].

2 SISTEMAS EN COMITÉ

Los sistemas en comité (arquitecturas y modelos de aprendizaje) se pueden clasificar en dos categorías fundamentales:

- Estructuras estáticas, en las que las respuestas de los expertos se combinan según un mecanismo en el que no influye el valor de la entrada para dar lugar a la salida final de la red. A esta categoría pertenece la arquitectura *ventaja del conjunto* y el entrenamiento *boosting*.
- Estructuras dinámicas, donde el valor de la entrada influye en la manera en que se integran las respuestas parciales de los expertos para dar lugar a la salida global de la red. Entre ellas se encuentra el modelo mezcla de expertos y sus variantes.

2.1 ESTRUCTURAS ESTÁTICAS

Dentro de estas arquitecturas se pueden distinguir los sistemas basados en ventaja del conjunto y los sistemas con entrenamiento *boosting*.

2.1.1 Ventaja del Conjunto

Esta arquitectura está compuesta por una serie de redes neuronales (expertos) cuyas salidas se combinan linealmente para dar lugar a la respuesta de la red. Durante el proceso de aprendizaje, todos los expertos son entrenados con los mismos datos, pero pueden diferir entre sí en las condiciones iniciales del proceso de entrenamiento.

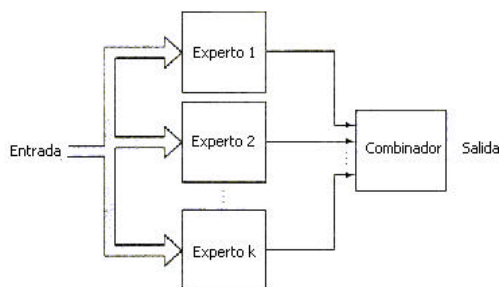


Figura 1. Ventaja del conjunto

La utilización de esta configuración presenta las siguientes ventajas:

- Al entrenar en paralelo las redes expertas, el tiempo de entrenamiento es mucho menor que el que se emplearía en ajustar todos los parámetros de una red multicapa clásica.
- Además, dado que el número de parámetros a ajustar es menor, disminuye el riesgo de sobreajustar los datos de entrenamiento.

Siguiendo la notación:

$x^{(n)}$: vector entrada.

$y_i^{(n)}$: vector salida del experto i .

La estimación realizada por la red (salida global) es una función de las salidas parciales de los expertos:

$$\hat{y}^{(n)} = \Psi(y_1^{(n)}, y_2^{(n)}, \dots, y_K^{(n)}) \quad (1)$$

En el caso más básico, cada experto es una red en perceptrón simple.

2.1.2 Entrenamiento *boosting*

Es otro de los métodos pertenecientes a las denominadas *estructuras estáticas*. Difiere de los sistemas basados en *ventaja del conjunto* en que en la configuración *boosting* los expertos son entrenados con grupos de datos con distribuciones diferentes. En realidad, se trata de un método de entrenamiento que permite mejorar el aprendizaje de cualquier algoritmo neuronal, y que incluimos en este análisis por su común utilización en redes basadas en comités de expertos.

• *Boosting* por Filtrado

El sistema en comité incluye tres expertos que denominaremos en lo que sigue “primero”, “segundo”, “tercero”. Sea un problema clásico de clasificación de patrones, estos expertos son entrenados individualmente de la siguiente manera:

1. El primer experto es entrenado convencionalmente con un set de N_1 ejemplos.
2. Dado un sistema aleatorio que proporcione una salida booleana, si ésta es 0, pasamos datos a través del primer experto hasta que clasifique uno incorrectamente. Este patrón mal clasificado entrará a formar parte del set de entrenamiento del segundo experto.
3. Si resulta 1, cogemos el primer patrón correctamente clasificado por el primer experto para formar parte del entrenamiento del segundo experto.
4. Este proceso no finaliza hasta que el primer experto ha filtrado las N_1 muestras. Siguiendo este proceso, está claro que si ahora pasáramos a través del primer experto el set de entrenamiento obtenido del filtrado, tendríamos un porcentaje

de error del 50%. Esto es, el grupo de datos con los que aprenderá el segundo experto tiene una distribución totalmente diferente a la que tenía el set de entrenamiento del primer experto.

- Una vez que el segundo experto ha sido entrenado en la manera habitual, formamos un nuevo set de entrenamiento para el tercer experto. Para ello, pasamos un nuevo patrón a través de los expertos 1 y 2. Si ambos lo clasifican en la misma categoría, se descarta ese patrón. En caso de que las decisiones de los expertos difieran, el patrón pasa a formar parte del set de entrenamiento del tercer experto. Este proceso continúa hasta que han pasado por los dos primeros expertos N_1 muestras.

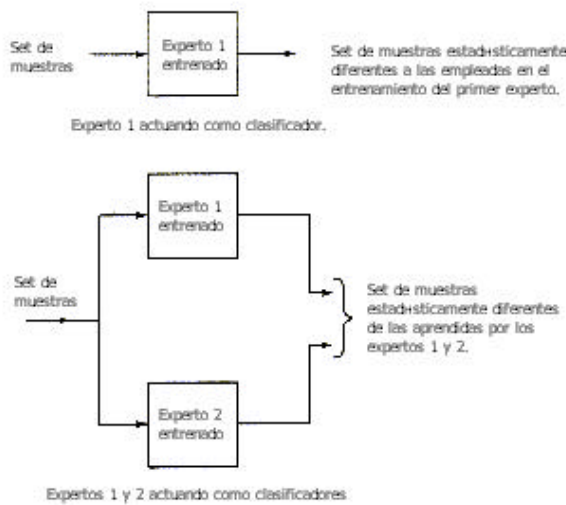


Figura 2. Boosting por filtrado

- **AdaBoost**

Como fácilmente se puede deducir, la limitación principal del entrenamiento mediante *boosting* por filtrado estriba en la necesidad de una gran cantidad de patrones en la muestra. El método *AdaBoost* suple este handicap permitiendo la reutilización de los datos de la muestra.

2.2 ESTRUCTURAS DINÁMICAS

Dentro de esta arquitectura cabe resaltar los sistemas de mezcla de expertos y los sistemas de mezcla jerárquica de expertos que no es más que una extensión del primero.

2.2.1 Mezcla de expertos

Esta configuración está integrada por K módulos que denominamos *redes expertas* o simplemente expertos y una unidad llamada usualmente *gating network* que realiza una función de mediación entre los diferentes expertos. Cada experto es una red especializada en una determinada región del espacio entrada, de ahí la

necesidad de la *gating network*, que modula las respuestas parciales de los expertos antes de dar lugar a la salida global del sistema.

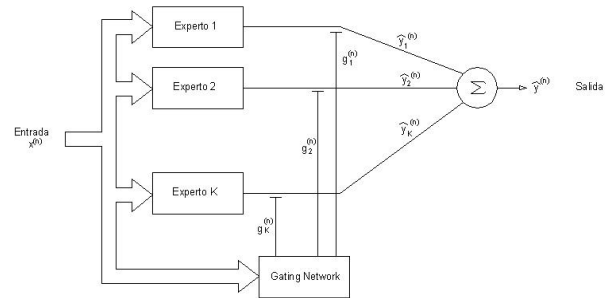


Figura 3. Mezcla de expertos

Sean las variables aleatorias Y (salida) y X (entrada). El problema de regresión puede ser considerado como aquél en el que se desea obtener la relación entre las variables X e Y a partir de la probabilidad de que un valor aleatorio de X de lugar a un determinado valor de Y . En este modelo, teniendo en cuenta que en la salida global de la red intervienen los K expertos y *gating network*, la descomposición entre las variables X e Y se establece según:

$$P(Y | X) = \sum_Z P(Z | X)P(Y | X, Z) \quad (2)$$

donde Z es la variable que determina qué experto da lugar a cada dato.

De este modo, si denotamos al experto i como \mathbf{x}_i , tenemos que, para un set de N puntos de entrenamiento, Z está constituida por N vectores $\mathbf{z}^{(n)}$ tales que:

$$z_i^{(n)} = 1 \quad (3)$$

si la salida $\mathbf{y}^{(n)}$ se obtiene a partir de la entrada $\mathbf{x}^{(n)}$ y los parámetros asociados al experto \mathbf{x}_i . En otro caso,

$$z_i^{(n)} = 0 \quad (4)$$

Los parámetros de la red gating quedan agrupados en el vector \mathbf{v} . De este modo, la contribución de la red puerta en la expresión (1) puede ser escrita como:

$$P(\mathbf{z}^{(n)} | \mathbf{x}^{(n)}, \mathbf{v}) \quad (5)$$

Los expertos intervienen según unos pesos \mathbf{w} ,

$$P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, z_i^{(n)} = 1, \mathbf{w}_i) \quad (6)$$

Para el caso general de una arquitectura en mezcla de expertos donde se tengan K expertos $\{\mathbf{x}_i\}$ y N pares $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}$, la expresión (1) queda:

$$P(Y | X, \mathbf{q}) = \prod_{n=1}^N \sum_{i=1}^K P(z_i^{(n)} = 1 | \mathbf{x}^{(n)}, \mathbf{v}) P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, z_i = 1, \mathbf{w}_i) \quad (7)$$

donde \mathbf{q} es el vector global de parámetros

$$\mathbf{q} = \{\mathbf{v}, \mathbf{w}_i\} \quad (8)$$

Utilizando la notación abreviada:

$$P(\mathbf{x}_i | \mathbf{x}^{(n)}, \mathbf{v}) = P(z_i^{(n)} = 1 | \mathbf{x}^{(n)}, \mathbf{v}) \quad (9)$$

$$P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, \mathbf{x}_i, \mathbf{w}_i) = P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, z_i^{(n)} = 1, \mathbf{w}_i) \quad (10)$$

Para una serie de patrones $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}$, la aproximación dada por el modelo queda:

$$\hat{\mathbf{y}}^{(n)} = E(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, \mathbf{q}) = \sum_{i=1}^K P(\mathbf{x}_i | \mathbf{x}^{(n)}, \mathbf{v}) E(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, \mathbf{x}_i, \mathbf{w}_i) = \sum_{i=1}^K g_i^{(n)} \hat{\mathbf{y}}_i^{(n)} \quad (11)$$

donde hemos utilizado la notación abreviada:

$$g_i^{(n)} = P(\xi_i | \mathbf{x}^{(n)}, \mathbf{v}) \quad (12)$$

$$\hat{\mathbf{y}}_i^{(n)} = E(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, \xi_i, \mathbf{w}_i) \quad (13)$$

En el caso más simple, cada experto puede ser considerado como un filtro lineal. De este modo, la salida de un experto ξ_i vendrá dada por el producto escalar del vector entrada \mathbf{x} por el vector \mathbf{w} en que se almacenan los pesos de la sinapsis correspondiente:

$$\hat{\mathbf{y}}_i = \mathbf{w}_i^T \mathbf{x}^{(n)} \quad (14)$$

Las salidas de la red ($\mathbf{g}_i^{(n)}$) pueden ser interpretadas como la probabilidad de que ante una determinada entrada $\mathbf{x}^{(n)}$ en la salida global intervenga el experto \mathbf{x}_i .

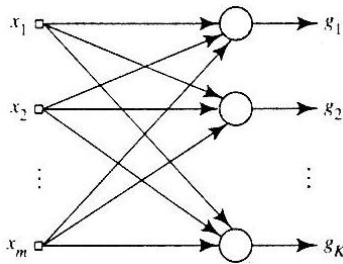


Figura 4. Configuración clásica de una *gating network*

La presencia de la *gating network* se inscribe dentro del concepto *modular* de Nowlan, Jacobs y Hinton, que describieron a este tipo de redes desde una perspectiva competitiva. Desde este punto de vista, la *gating network* se establece para dar lugar a una competición entre los expertos, de modo que la mayor contribución en la salida la tenga la red experta ganadora en la *gating network*. Recordemos que cada experto está especializado en una región concreta del espacio entrada, y este sistema competitivo permite que en para cada entrada trabaje el experto que puede dar una salida más adecuada.

La *gating network* suele estar constituida por una capa de entrada alimentada por el vector \mathbf{x} y otra de salida, tal y como se muestra en la Figura 4. La salida de la *gating network* son los valores g_k , correspondientes a los K expertos, de manera que:

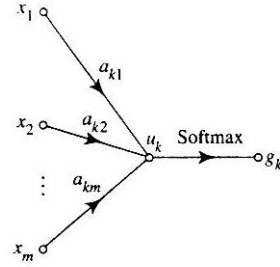


Figura 5. Variables en red *gating*

$$u_i = \mathbf{a}_i^T \mathbf{x}^{(n)} \quad (15)$$

$$g_i = \frac{\exp(u_i)}{\sum_{j=1}^K \exp(u_j)} \quad (16)$$

Esta operación puede ser vista como una generalización de la función logística para una entrada de n dimensiones. Esta función de activación se denomina *softmax*, y se trata de una expresión diferenciable mediante la que se regula la participación de cada experto en la salida global. De este modo, en la salida no sólo contribuye la red experta “ganadora” de la competición en la *gating network*, (la correspondiente al valor más alto de g_i), sino que todos los expertos influyen en la señal de salida \mathbf{y} .

$$\mathbf{y}^{(n)} = \sum_{i=1}^K g_i^{(n)} \hat{\mathbf{y}}_i^{(n)} \quad (17)$$

Podemos asumir que el error ϵ_i sigue una distribución gaussiana con media cero y varianza unitaria que se verifican para cualquier valor de i .

$$E[\epsilon_i] = 0 \quad , \quad \text{var}[\epsilon_i] = 1 \quad (18)$$

2.2.2 Mezcla jerárquica de expertos

En el modelo anterior, mediante una sola red *gating*, el espacio de entrada queda dividido en diferentes subespacios o regiones. Si los expertos y la red *gating* están constituidos por arquitecturas de perceptrón simple, estas divisiones del espacio entrada serán lineales. Esto limita la capacidad del modelo para modelar funciones no lineales. Para solventar esto podemos considerar estructuras multiperceptrón para los expertos y la red *gating*. Otra opción es hacer que los expertos sean a su vez redes en mezcla de expertos. Es lo que se conoce como mezclas jerárquicas de expertos. Si lo visualizamos como una estructura en árbol, las hojas contendrían las redes expertas, mientras que los nodos no terminales estarían constituidos por las redes de puerta.

De este modo, y análogamente al análisis realizado para las arquitecturas en mezcla de expertos, podemos escribir la probabilidad condicional de obtener Y a partir de X mediante un modelo en mezcla jerárquica de expertos con 3 niveles i, j, k como:

$$\prod_n P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}) \quad (19)$$

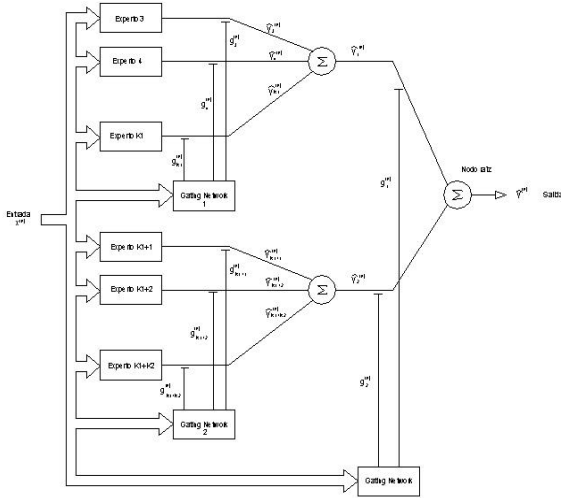


Figura 6. Mezcla jerárquica de expertos

, siendo:

$$\begin{aligned} P(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) = & \sum_i P(z_i^{(n)} = 1 | \mathbf{x}^{(n)}) \sum_j P(z_j^{(n)} = 1 | z_i^{(n)} = 1, \mathbf{x}^{(n)}) \\ & \sum_k P(z_k^{(n)} = 1 | z_j^{(n)} = 1, z_i^{(n)} = 1, \mathbf{x}^{(n)}) \\ & P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, z_k^{(n)} = 1, z_j^{(n)} = 1, z_i^{(n)} = 1) \end{aligned} \quad (20)$$

La densidad de probabilidad condicional:

$$\begin{aligned} P(Y | X, \mathbf{q}) = & \prod_{n=1}^N \sum_i P(z_i^{(n)} = 1 | \mathbf{x}^{(n)}, \mathbf{v}_0) \sum_j P(z_j^{(n)} = 1 | z_i^{(n)} = 1, \mathbf{x}^{(n)}, \mathbf{v}_j) \\ & \sum_k P(z_k^{(n)} = 1 | z_j^{(n)} = 1 | z_i^{(n)} = 1 | \mathbf{x}^{(n)}, \mathbf{v}_j) \\ & P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, z_k^{(n)} = 1, z_j^{(n)} = 1, z_i^{(n)} = 1, \mathbf{w}_k) \end{aligned} \quad (21)$$

De modo que la salida de la red en este caso:

$$\begin{aligned} \hat{\mathbf{y}}^{(n)} = E(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, \mathbf{q}) = & \sum_i P(z_i^{(n)} = 1 | \mathbf{x}^{(n)}, \mathbf{v}_0) \\ & \sum_j P(z_j^{(n)} = 1 | z_i^{(n)} = 1, \mathbf{x}^{(n)}, \mathbf{v}_j) * \\ & \sum_k P(z_k^{(n)} = 1 | z_j^{(n)} = 1, z_i^{(n)} = 1, \mathbf{x}^{(n)}, \mathbf{v}_j) \\ & E(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, z_k^{(n)} = 1, z_j^{(n)} = 1, z_i^{(n)} = 1, \mathbf{w}_k) = \\ & \sum_i g_i^{(n)} \sum_j g_j^{(n)} \sum_k g_k^{(n)} \hat{\mathbf{y}}_k^{(n)} \end{aligned} \quad (22)$$

Empleando la notación abreviada:

$$\begin{aligned} P(z_k^{(n)} = 1 | z_j^{(n)} = 1, z_i^{(n)} = 1, \mathbf{x}^{(n)}, \mathbf{v}) \equiv & \\ P(\mathbf{x}_k | An(\mathbf{x}_k), \mathbf{x}^{(n)}, \mathbf{v}) \end{aligned} \quad (23)$$

Podemos escribir:

$$\begin{aligned} \hat{\mathbf{y}}^{(n)} = & \sum_i P(\mathbf{x}_i | \mathbf{x}, \mathbf{v}_0) \sum_j P(\mathbf{x}_j | An(\mathbf{x}_j), \mathbf{x}, \mathbf{v}_j) \sum_k P(\mathbf{x}_k | An(\mathbf{x}_k), \mathbf{x}, \mathbf{v}_j) \\ & E(\mathbf{y} | \mathbf{x}, \mathbf{x}_k, An(\mathbf{x}_k), \mathbf{w}_k) \end{aligned} \quad (24)$$

donde, $An(\mathbf{x}_p)$ es el camino más corto desde el experto \mathbf{x}_p hasta el nodo de salida.

3 ALGORITMOS PARA EL APRENDIZAJE DE LAS ARQUITECTURAS DE MEZCLA DE EXPERTOS

La principal propiedad de cualquier red neuronal es su capacidad para aprender del entorno, de manera que el comportamiento del modelo mejore a través de este proceso de aprendizaje.

Matemáticamente, el aprendizaje se traduce en el reajuste de los pesos sinápticos en cada iteración del entrenamiento.

Citando a Mendel y McClaren (1970): “Aprendizaje es un proceso en el que los parámetros de una red neuronal se adaptan durante un período en el que la red es estimulada por el medio en el que se encuentra. Los diferentes tipos de aprendizaje difieren entre sí en la manera en que tienen lugar dichos cambios en los parámetros”.

3.1 ALGORITMOS DE APRENDIZAJE SUPERVISADO

Este paradigma de aprendizaje lo constituyen modelos que emplean en el entrenamiento de la red una colección de patrones entrada-salida deseada $\{x^{(n)}, y^{(n)}\}$.

Se trata de reajustar los pesos neuronales minimizando una determinada función del error global. Consideremos que la salida de la red es un vector. El error cometido en la estimación es:

$$\epsilon = \hat{\mathbf{y}}^{(n)} - \mathbf{y}^{(n)} \quad (25)$$

Seguendo el algoritmo clásico de aprendizaje por error-corrección, la función a minimizar sería el valor instantáneo de la energía del error:

$$\hat{\mathbf{I}}^{(n)} = \frac{1}{2} \epsilon^2 = \frac{1}{2} (\hat{\mathbf{y}}^{(n)} - \mathbf{y}^{(n)})^T (\hat{\mathbf{y}} - \mathbf{y}^{(n)}) \quad (26)$$

donde $\hat{\mathbf{y}}^{(n)}$ responde a la ecuación (24). De este modo la actualización de los pesos de los expertos, tendente a minimizar esta función:

$$\Delta w_{ij} = -\mathbf{a} \frac{\partial \hat{\mathbf{I}}^{(n)}}{\partial w_{ij}} \quad (27)$$

Un procedimiento análogo podría emplearse con los pesos de la red *gating*. Sin embargo, dado que en esta arquitectura disponemos de varias redes que se pueden especializar en regiones diferentes del espacio entrada, sería más interesante considerar una función del error en la *gating network* que nos permitiera incorporar elementos de aprendizaje por refuerzo, y cuya minimización condujera a la especialización antes mencionada. Una gran colección de patrones permitiría en un momento dado entrenar a cada experto con los patrones de la región del espacio entrada en que lo quisiéramos especializar. Sin embargo, en raras ocasiones esto es posible. Conviene emplear algoritmos en los que todos los pares entrada-salida sean útiles en el entrenamiento de la totalidad de las redes expertas. Matemáticamente, esto se traduce en lo siguiente:

Si consideramos la función:

$$\bar{\epsilon}(t) = \mathbf{a} \epsilon(t) + (1 - \mathbf{a}) \bar{\epsilon}(t-1) \quad (28)$$

ante un descenso brusco del error entre dos etapas del entrenamiento:

$$\epsilon(t) < \mathbf{g} \bar{\epsilon}(t-1) \quad (29)$$

hacemos que las variables binarias λ_{WTA} y λ_{NT}

$$\mathbf{I}_{WTA} = 1 \quad (30)$$

$$\mathbf{I}_{NT} = 0 \quad (31)$$

y que la salida de la *gating network* para el experto ganador sea 1 ($\tilde{g}_i = 1$).

Si en el algoritmo provocamos que sólo se actualicen los pesos correspondientes a los expertos cuyas salidas parciales hayan sido más aproximadas al valor real a estimar, al final del proceso de entrenamiento obtendremos que cada experto ha optimizado su comportamiento en una región del espacio entrada, de modo que la eficiencia global de la red se incrementa notablemente. En caso de que la diferencia dada por la expresión (29) no sea relevante,

$$\mathbf{I}_{WTA} = 0 \quad (32)$$

$$\mathbf{I}_{NT} = 1 \quad (33)$$

($\tilde{g}_i = 0$) se ajustan los pesos de la *gating* para que las g_i tomen valores neutrales que tiendan a equiparar las contribuciones de los diferentes expertos en la salida global de la red. Considerando lo anterior, la función error característica de la red *gating* puede escribirse:

$$\begin{aligned} \epsilon_G = & \mathbf{I}_{WTA} \frac{1}{2} \sum_{i=1}^K (\tilde{g}_i - g_i)^2 + \\ & \mathbf{I}_{WTA} \frac{1}{2} \left(1 - \sum_{i=1}^K g_i \right)^2 + \mathbf{I}_{WTA} \sum_{i=1}^K g_i (1 - g_i) \\ & + \mathbf{I}_{NT} \frac{1}{2} \sum_{i=1}^K \left(\frac{1}{K} - g_i \right)^2 \end{aligned} \quad (34)$$

de manera que si buscamos que los nuevos pesos de la red *gating* tiendan a disminuir este error, las ecuaciones de actualización:

$$\Delta W_{ij} = -\mathbf{a} \frac{\partial \epsilon_G}{\partial W_{ij}} \quad (35)$$

3.2 ALGORITMOS DE APRENDIZAJE NO SUPERVISADO

Pertenecen a este paradigma de aprendizaje todos aquellos algoritmos de entrenamiento de redes neuronales que sólo pueden recurrir a la información proporcionada por el medio en el que está actuando la red neuronal. En el análisis que sigue nos centraremos en los métodos de aprendizaje por refuerzo, dado que constituyen un conjunto de técnicas de aprendizaje muy habituales en sistemas biológicos.

3.2.1 Aprendizaje por refuerzo

En los modelos que aprenden por refuerzo, un agente computacional aprende a realizar una determinada acción recibiendo un *feedback* de información a través de la interacción con el medio, sin un profesor que explícitamente vaya dando instrucciones o se conozca una serie de patrones entrada-salida susceptible de ser utilizada en el entrenamiento. El agente computacional aprende reforzando situaciones que han dado lugar a acciones correctas e inhibiendo las que conducen a errores. En otras palabras, si una acción tiene como consecuencia que el sistema avance hacia un estado satisfactorio, se refuerza la tendencia a producir dicha acción. En otro caso, dicha tendencia se debilita por penalización. Actualmente existen algoritmos mediante los que el agente no necesita esperar la recepción de los datos del entorno para realizar modificaciones que mejoren el grado de éxito de la acción. Un ejemplo de esto son los *métodos de diferencia temporal*, que detallamos a continuación dado que serán los que aplicaremos a nuestro problema.

3.2.2 Métodos de diferencia temporal

Los métodos de diferencia temporal son una clase de procesos de aprendizaje incrementados diseñados para adaptar los pesos sinápticos a partir de diferencias entre predicciones temporales. Análogamente a como se hace en algoritmos supervisados, podemos establecer una función del error, entendiendo éste como la diferencia entre predicciones sucesivas:

$$\epsilon_{td} = \frac{1}{2} \left\{ \sum_{k=t}^m (V_{k+1} - V_k) \right\}^2 \quad (36)$$

donde m constituye el número de instantes en que realizamos predicciones temporales (V) y adaptamos los pesos según:

$$TD(\mathbf{I}): \Delta w_t = \mathbf{a}(V_{t+1} - V_t) \sum_{k=1}^t \mathbf{I}^{t-k} \nabla_w V_k \quad (37)$$

con λ variando entre 0 y 1.

El cálculo de las predicciones se puede realizar a partir de los parámetros \mathbf{v} y \mathbf{b} :

$$V_t = \mathbf{v} \cdot \mathbf{x}^{(n)} + \mathbf{b} \quad (38)$$

3.2.3 Q-Learning

Este algoritmo se basa en la asignación de valores a una colección de pares estado-reacción. Cada acción es evaluada, de modo que el estado que la produjo es reforzado o penalizado en función de la calidad de la acción. La tendencia es a maximizar los refuerzos producidos en un determinado período de tiempo. La implementación de este algoritmo se basa en los siguientes pasos:

- Determinación del estado actual s_n .
- Selección estocástica de acción a_n .
- Usar s_n y a_n como entrada en la red Q. Salida: $U_a \leftarrow Q_{n-1}(s_n, a_n)$.
- Ejecución de la acción a_n .
- Refuerzo del nuevo estado: $r_n \quad t \leftarrow t_n$.
- Con una nueva acción b , computar mediante la red Q: $Q_{n-1}(t, b)$.
- Cálculo de $u \leftarrow r_n + \max_{b \in \text{acciones}} Q_{n-1}(t, b)$.
- Ajuste de los parámetros de la red Q propagando el error:

$$\Delta U = u - U_a \text{ si } s=s_n \text{ y } a=a_n. \quad (39)$$

$$\Delta U = 0 \text{ en otro caso.} \quad (40)$$

4 EXPOSICIÓN DEL PROBLEMA

Si en la tarea de llevar la mano desde una determinada posición espacial hasta otra posición objetivo la información visual es discordante con el movimiento que se produce, se hace necesario un proceso de aprendizaje que permita la adaptación a la nueva situación del mapa visuo-motor del individuo. El modo en el que el agente modifica su mapa visuo-motor inicial en situaciones de distorsión visuo-motora ha sido descrito en los experimentos de Contreras, Kagerer y Stelmach.

En estos experimentos, se provocó en los individuos la necesidad de establecer nuevas relaciones entre la información visual y dada por el sistema motor rotando artificialmente el espacio visual en el que se enmarca la trayectoria entre los puntos inicial y objetivo. En estas circunstancias, las direcciones en que se producen los movimientos iniciales no son las adecuadas para un correcto alcance de la posición objetivo. Por ejemplo, la dirección correcta si se provoca una rotación de 90° en el espacio visual sería

aquella perpendicular a la línea recta entre los puntos inicial y objetivo (definidora de la trayectoria en caso de no existir ningún tipo de alteración).

Contreras, Kageger y Stelmach [2] analizan la relación entre el proceso de aprendizaje y la evolución temporal de la perturbación comparando la adaptación que se produce cuando se provocan incrementos graduales en la alteración del espacio visual (rotaciones de 0° a 90°) con el efecto de establecer rotaciones súbitas de 90°.

De los resultados obtenidos para el caso gradual (Figura 7), se deduce que progresivas modificaciones en el sistema de referencia visual inducen en el sistema motor del agente pequeñas rotaciones correctoras que no requieren un aprendizaje exhaustivo de la nueva situación, sino que están basadas en un conocimiento de la relación entre la información visual y motora previo al cambio en el entorno visual. Por el contrario, ante un cambio brusco del espacio visual, se observa que el agente ha de realizar un análisis más intenso de su mapeado visuo-motor, planteándose la necesidad de un aprendizaje rápido de la nueva situación.

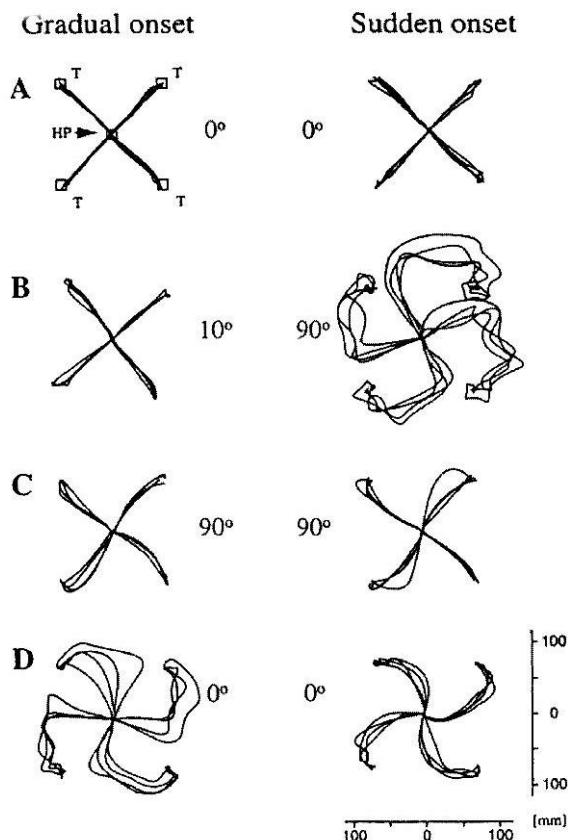


Figura 7. Trayectorias ante diversos grados de distorsión

4.1 MODELO PARA EL COMPORTAMIENTO HUMANO EN SITUACIONES DE DISTORSIÓN VISUO-MOTORA

El modelo propuesto está basado en las redes neuronales propuestas por Bullock, Grossberg y colaboradores [1],[3]. Estas redes modelan las transformaciones visuo-motoras llevadas a cabo en mamíferos superiores en tareas de alcance hacia objetivos seleccionados visualmente. El modelo se consta de dos bloques funcionales entrelazados que se relacionan con el comportamiento consciente e inconsciente del individuo.

En la figura 8 se muestra la estructura del modelo propuesto sobre el cual se le ha incluido la red de comités de expertos. Esta red modela el comportamiento consciente del individuo, es decir la reacción del individuo cuando percibe que se le ha producido una distorsión visual.

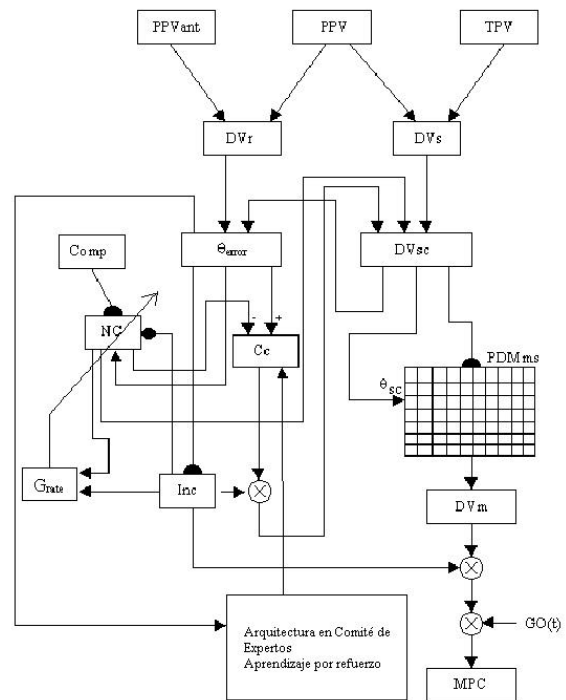


Figura 8. Trayectorias ante diversos grados de distorsión

Los parámetros del modelo son los siguientes:

Parámetros geométricos:

- *PPV*, (*Present Position Vector*), posición espacial actual del apuntador sobre el monitor.
- *TPV*, (*Target Position Vector*), posición espacial del objetivo.

- *PPVant*, (*Anterior Present Position Vector*), posición espacial del apuntador antes de la última acción.
- *DVs*, (*spatial Difference Vector*), vector de diferencias espacial. Se determina como: $DVs=TPV-PPV$ y está expresado en coordenadas polares y tiene de componentes: q_s, d_s .
- *DVsc*, (*compensated spatial Difference Vector*), vector de diferencias espacial compensado. Este vector recibe dos tipos de compensación una denominada consciente y que es función del error percibido en la actuación y otra denominada inconsciente y que es función de la distorsión detectada y aprendida de los ensayos. Se calcula como: $DVsc=DVs+ \text{Compensación Consciente} + \text{Compensación Inconsciente}$.
- *DVr*, (*reaction Difference Vector*), vector de diferencias espacial en coordenadas polares. Determina el efecto de la última acción realizada. Se calcula como: $DVr=PPV-PPVant$ y tiene de componentes: q_r, d_r .
- q_{error} , este vector determina el error en el ángulo cometido en el último ensayo. En el ensayo anterior se deseaba llevar a cabo un movimiento con ángulo de rotación q_{sc} y la acción que se realizó tenía un ángulo de q_r . Esta variable permite obtener información de la última acción realizada y por tanto permite valorar los ajustes internos necesarios.

Parámetros de aprendizaje

- *NC*, (*Neuron Compensator*), efectúa la compensación inconsciente. Su efecto compensador es el de estabilizar el sistema.
- *Cc*, (*Compensación consciente*), mediante la que se modela la acción compensatoria que el agente realiza conscientemente.
- *PDDMms*, (*motor-spatial Position Direction Map*), mapa de la transformación espacial-motora. Recibe como entrada el incremento de distancia d_{sc} y la variable q_{sc} (DV_{sc}) como el elemento de selección de celda. La salida de este mapa es el incremento motor en coordenadas X e Y que el operador realizará sobre el tablero digitalizador.
- *DVm*, (*motor Difference Vector*), donde se almacena la salida de *PDDMms*.
- *MPC*, (*Motor Position Command*), que acumula los incrementos de posición del vector *DVm*.
- *DVm*, comandos de posición motores que se enviarán al sistema de actuación.
- *Inc*, (Incertidumbre), variable que actúa como factor de atenuación sobre la velocidad de los movimientos del sistema de actuación.
- *Grate*, representa una acción de compuerta. Permite que en situaciones de gran incertidumbre la variable *NC* pueda aprender

rápido y posteriormente olvidar ese aprendizaje con gran velocidad.

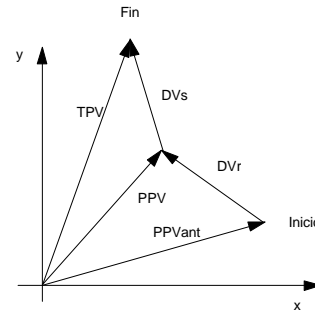


Figura 9. Variables geométricas del modelo.

En el modelo se considera que el comportamiento biológico en la compensación motora se sustenta en dos acciones: consciente e inconsciente.

$$DVsc=DVs+Comp.Consciente+Comp.Inconsciente \quad (41)$$

Una vez establecidas estas modificaciones sobre el vector que determina la orientación y la distancia en la siguiente etapa del acercamiento, se simula la utilización por parte del sistema motor de la información proporcionada por *DVsc* mediante el mapa *PDDMms*. De este modo, finaliza el ciclo por el cual el operador realiza un nuevo movimiento.

4.1.1 Compensación Inconsciente

El error cometido en la tarea de establecer una dirección de movimiento que aproxime a la posición objetivo determina una necesidad de corregir la trayectoria de la que el operador no tiene por qué ser consciente. Esta compensación inconsciente se modela a partir del error angular mediante el término:

$$NC = NC + rango \cdot (q_{error} - NC) \quad (42)$$

El término rango determina la velocidad de aprendizaje del sistema. De este modo, un valor alto de rango implica que *NC* aprende con gran velocidad, pero que también olvida lo aprendido rápidamente.

El efecto de la variable rango puede ser modulado por la acción de la variable booleana *Grate* (los conceptos “puerta abierta” y “puerta cerrada” serán equivalentes a 1 y 0, respectivamente) de la siguiente manera: si el error cometido en la última acción es considerable, la variable incertidumbre (que no es más que una normalización entre 0° y 90° de la variable q_{error}) toma un valor elevado. Esto activará la variable *Grate* (*Grate*=1), de manera que rango se modifica para aumentar la velocidad de aprendizaje de *NC*. La puerta *Grate* permanecerá cerrada mientras la incertidumbre sea baja y el papel de la compensación inconsciente en el cálculo de *DVsc* sea

poco importante. En el cálculo de la incertidumbre se puede considerar que en el caso de un operador humano, a la activación precede un retraso temporal determinado por el hecho de que la acción de la dopamina no es inmediata.

Para conseguir que en situaciones de incertidumbre muy elevada el término NC no se actualice es necesario introducir un factor inhibidor en la expresión del cálculo de la compensación inconsciente.

$$\text{factor_inhibidor} = 0.5000005 - \text{incierto} \quad (43)$$

4.1.2 Compensación Consciente

La contribución del término consciente en la compensación se puede modelar como,

$$\text{Compensa}_{Kcc} = \text{Compensa} + Kcc \cdot (\mathbf{q}_{\text{error}} - NC - \text{Compensa}) \quad (44)$$

Bajo la hipótesis de que el agente humano mejora su comportamiento en la tarea de aproximación a través de un proceso de aprendizaje por refuerzo, el cálculo de la constante Kcc se realiza utilizando una red neuronal en mezcla de expertos que se alimenta del error cometido en el último movimiento.

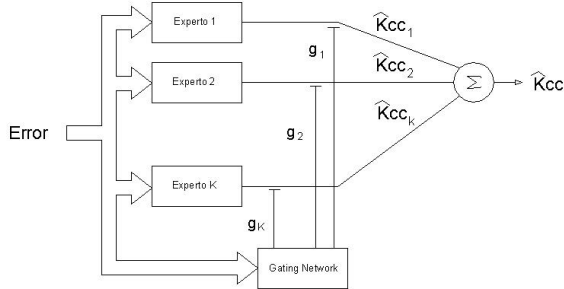


Figura 10. Estructura Neuronal propuesta para la obtención de Kcc .

De este modo, mediante el comité de expertos el espacio de entrada queda dividido en tantos subespacios como redes expertas consideremos, lo que disminuye el tiempo de entrenamiento² frente a la arquitectura clásica de perceptrón múltiple. Además, el hecho de que cada experto sea una red neuronal especializada en una región concreta del espacio de entrada aumenta la eficiencia global de la red.

² Los parámetros en una red en mezcla de expertos se ajustan en un proceso de aprendizaje en paralelo.

5 RESULTADOS

Aplicamos una red de tres expertos (caracterizados por una matriz de pesos w) que aprenden por refuerzo según ecuaciones basadas en el método de aprendizaje en diferencia temporal, en las que el TD error queda definido como:

$$\mathbf{d}(t) = r(t) + \mathbf{g} \cdot P(t) - P(t-1) \quad (45)$$

para obtener en cada acción el valor más adecuado de la constante Kcc . Los parámetros y variables que intervienen en las ecuaciones de esta sección son:

- r : señal de refuerzo que depende de la superación de un determinado umbral de la entrada.
- x : entrada a la red (error acumulado en nuestro caso).
- v : vector de pesos de las mismas dimensiones de la entrada.
- \mathbf{b} , eta_v , eta_b , eta_w , y eta_g : magnitudes escalares.

Las predicciones se calculan como:

$$P(t) = v \cdot x(t) + \mathbf{b} \quad (46)$$

$$P(t-1) = v \cdot x(t-1) + \mathbf{b} \quad (47)$$

v y \mathbf{b} se actualizan según:

$$\Delta v = \text{eta}_v \cdot v \cdot \mathbf{d}(t) \cdot x(t-1) \quad (48)$$

$$\Delta \mathbf{b} = \text{eta}_b \cdot \mathbf{b} \cdot \mathbf{d}(t) \quad (49)$$

Las ecuaciones de actualización de los pesos, en este modelo no supervisado:

$$\Delta w_k = \text{eta}_w \cdot \mathbf{d}(t) \cdot y_k \cdot x(t) \quad (50)$$

$$\Delta W_j = \text{eta}_g \cdot \mathbf{d}(t) \cdot g_j \cdot x(t) \quad (51)$$

donde W_j es el vector fila j -ésimo de la matriz W (matriz de pesos de las neuronas de la *gating network*) y g_j la salida j -ésima de la *gating network*.

Los expertos en el caso gradual sufren una secuencia de activación que queda plasmada en la figura 12.

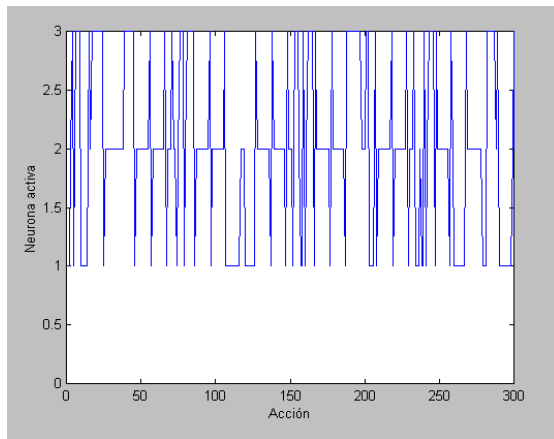


Figura 11. Secuencia de activación de las neuronas en la red de puertas.

De este modo, la variable K_{cc} experimenta un proceso de adaptación que la lleva a establecerse en un rango de valores que en el que el comportamiento del modelo es óptimo.

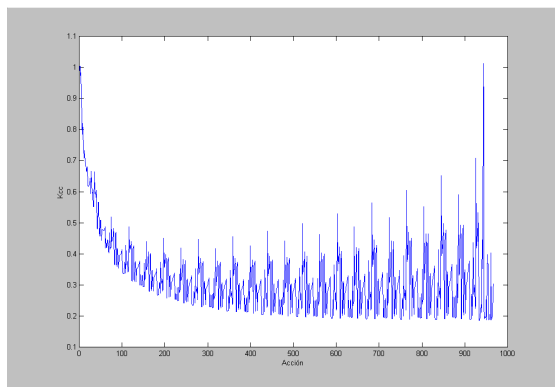


Figura 12. Evolución de K_{cc} en el caso gradual.

6 CONCLUSIONES

En este artículo se ha realizado una revisión de las arquitecturas neuronales en comités de expertos y a su vez se ha realizado una aplicación concreta a un problema de modelado. En este trabajo a su vez se ha mostrado brevemente un modelo del comportamiento visuo-motor humano ante situaciones de distorsión visuo-motora. Este trabajo muestra brevemente una línea importante de investigación de los autores, en donde se trata crear un modelo que se ajuste a los resultados obtenidos en experimentaciones con personas. Se ha mostrado como las arquitecturas en comités de expertos pueden dar una solución a un bloque concreto dentro del modelo global el cual se encarga de modelar el comportamiento consciente de los individuos en situaciones de alteraciones visuo-motoras.

Referencias

- [1] Bullock D., Grossberg F. and Guenther F., "A self organizing of motor equivalent reaching and tool use by a multijoint arm", *Journal of Cognitive Neuroscience*. 5:4, pp 408-435.
- [2] Florian, A. Kagerer, Jose Luis Contreras Vidal, (1997) "Adaptation to gradual as compared with sudden visuo-motor distorsion", *Exp.Brain Res. Springer-Verlag*, pp.115-557-561.
- [3] Gaudiano P. and Grossberg S., "Vector associative maps: unsupervised read time error based learning and control of movements trajectories". *Neural Networks*. Vol 4, 1991, pp. 411-434.
- [4] Waterhouse, S. "Clasificación and Regression using Mixture of Experts". PH.D, University of Cambridge..