

Universidade Federal de Santa Catarina - UFSC  
Programa de Pós-Graduação em Engenharia Elétrica  
Departamento de Engenharia Elétrica  
Grupo de Pesquisas em Engenharia Biomédica - GPEB

**Emergência da Inteligência em Agentes Autônomos através  
de Modelos Inspirados na Natureza**

Tese Submetida à Universidade Federal de Santa Catarina Como Parte dos  
Requisitos para Obtenção do Grau de Doutor em Engenharia Elétrica - Área de  
Concentração em Sistemas de Informação

Mauro Roisenberg

Florianópolis, 1998

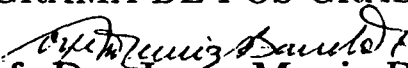
**Emergência da Inteligência em Agentes Autônomos através  
de Modelos Inspirados na Natureza**

**Mauro Roisenberg**

**ESTA TESE FOI JULGADA ADEQUADA PARA OBTENÇÃO  
DO TÍTULO DE**

**DOUTOR EM ENGENHARIA ELÉTRICA**


**ÁREA DE CONCENTRAÇÃO EM SISTEMAS DE  
INFORMAÇÃO, E APROVADA EM SUA FORMA FINAL PELO  
PROGRAMA DE PÓS-GRADUAÇÃO**

  
**Prof. Dr. Jorge Muniz Barreto**

**Orientador**

  
**Prof. Dr. Fernando Mendes de Azevedo**

**Co-Orientador**

  
**Prof. Dr. Adroaldo Raizer**

**Coordenador do Curso de Pós-Graduação em Engenharia  
Elétrica**

**Banca examinadora:**

  
Prof. Dr. Jorge Muniz Barreto, orientador

  
Prof. Dr. Fernando Mendes de Azevedo, co-orientador

  
Prof. Dr. Dante Augusto Barone

  
Prof. Dr. Emmanuel Pisces Lopes Passos

  
Prof. Dr. João Bosco da Mota Alves

  
Prof. Dr. Paulo Sérgio da Silva Borges

*"Imagination is more important than knowledge."*

*Albert Einstein*

*"Alguns homens vêem as coisas como são, e dizem 'Por quê?' Eu sonho com as coisas que nunca foram e digo 'Por que não?' "*

*George Bernard Shaw*

*Para os crentes Deus está no princípio das coisas.  
Para os cientistas no final de toda a reflexão."*

*Max Planck*

*À Mariane, Bruna e Henrique*

# Agradecimentos

Primeiramente gostaria de agradecer ao meu orientador, Prof. Dr. Jorge Muniz Barreto, por toda a sua dedicação, apoio e entusiasmo. Seu bom-humor, sua capacidade intelectual e sua disponibilidade em atender, ouvir e aconselhar foram fatores fundamentais para que este trabalho se realizasse. Encontrei nele mais do que um orientador, mas um amigo e conselheiro. Gostaria de agradecer também ao meu Co-orientador, Prof. Dr. Fernando Mendes de Azevedo, pela sua colaboração incondicional, seu apoio e interesse. Para mim foi uma honra e um privilégio ter podido trabalhar sob a supervisão exemplar destes dois cientistas.

Agradeço também aos membros que compuseram a banca da defesa da Tese, pela paciência em ler o meu trabalho e pelas suas valiosas sugestões e comentários.

Gostaria de agradecer aos colegas e funcionários do GPEB e, em especial, aos colegas João da Silva Dias, Lourdes Matos Brasil, John Wisbek e Roberto Célio Limão pela sua ajuda, interesse e amizade.

Especial agradecimento aos meus familiares, em especial aos meus pais que sempre me deram o amor, o exemplo, a ajuda e o apoio para que eu estudasse. À minha Mãe, pelo carinho e dedicação e ao meu Pai, pelo exemplo de amor à Ciência e à Vida Acadêmica.

Agradeço à CAPES e à UFRGS pelo suporte financeiro durante a realização deste trabalho.

Finalmente, mas não por último, agradeço à minha amada esposa e aos meus adorados filhos, pelo seu amor, apoio e inspiração. Esta Tese é para vocês.

# Publicações

1. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Generalization capacity in neural networks - the ballistic trajectory learning case. In *II Congresso Brasileiro de Redes Neurais*, Curitiba, PR, Oct. 1995.
2. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Specialization versus generalization in neural network learning for ballistic interception movement. In *MELECON'96 IEEE Mediterranean Electrotechnical Conference*, pages 627–630, Bari, Italy, May 1996.
3. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. A neural network that implements reactive behaviored autonomous agents. In *AEN'96 IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, Honolulu, Hawaii, Aug. 1996.
4. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Biological inspirations in neural network implementations of autonomous agents. In D. L. Borges and C. A. A. Kaestner, editors, *Advances in artificial intelligence : proceedings / 13th Brazilian Symposium on Artificial Intelligence*, number 1159 in lecture notes in computer science; Lecture notes in artificial intelligence, pages 211–220. Springer-Verlag, Berlin, Oct. 1996.
5. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Feedforward and recurrent neural networks complexity power - a comparison based on a concrete example. In *III Congresso Brasileiro de Redes Neurais*, pages 1–6, Florianópolis, SC, July 1997.
6. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Modeling behaviors with artificial neural networks. In *WRI'97 Workshop on Intelligent Robotics*, pages 34–45, Brasília, DF, Aug. 1997.
7. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Uma proposta de modelização para agentes autônomos baseada na teoria de sistemas. In *3º*

*SBAI Simpósio Brasileiro de Automação Inteligente*, pages 500–507, Vitória, ES, Sept. 1997.

8. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. On a formal concept of autonomous agents. In *AI'98 IASTED International Conference on Applied Informatics*, Garmisch-Paterkirchen, Germany, Feb. 1998.
9. M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Neural network classification based on the problem. In *IJCNN IEEE International Joint Conference on Neural Networks*, Anchorage, Alaska, May 1998.
10. J. M. Barreto, M. Roisenberg, and F. M. de Azevedo. Developing artificial neural networks for autonomous agents using evolutionary programming. In *ASC'98 IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 283–286, Cancún, México, May 1998.

# Sumário

<b>Publicações</b>	<b>i</b>
<b>Sumário</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Abreviaturas</b>	<b>xii</b>
<b>Resumo</b>	<b>xiii</b>
<b>Abstract</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos da Tese . . . . .	8
1.3 Organização do Texto . . . . .	9
<b>2 Agentes Autônomos</b>	<b>12</b>
2.1 Introdução . . . . .	12
2.2 Alternativas de Implementação . . . . .	15
2.2.1 Programação Direta . . . . .	16
2.2.2 Inteligência Artificial Simbólica . . . . .	17
2.2.3 Implementações baseadas em comportamento . . . . .	18
2.2.4 Outras alternativas . . . . .	21
2.2.5 Arquiteturas de Quadro-Negro e Multi-Agentes . . . . .	23
2.3 A Modelagem de Agentes Autônomos . . . . .	24
2.3.1 Conceitos de Sistemas . . . . .	26
2.3.2 Modelo de Agente . . . . .	38
2.3.3 Modelo de Ambiente . . . . .	43

2.3.4	Interação entre Agente e Ambiente . . . . .	44
2.3.5	Medidas de Desempenho . . . . .	45
<b>3</b>	<b>Comportamento Animal</b>	<b>47</b>
3.1	Introdução . . . . .	47
3.2	Fatores que determinam o comportamento . . . . .	49
3.3	Classes de comportamentos . . . . .	50
3.3.1	Comportamentos estereotipados . . . . .	51
3.3.2	Comportamento racional . . . . .	55
3.4	Autômatos de Estados Finitos como ferramenta para descrever comportamentos . . . . .	58
<b>4</b>	<b>Aprendizado</b>	<b>64</b>
4.1	Introdução . . . . .	64
4.2	O que é Aprendizado? . . . . .	65
4.3	Tipos de Aprendizado . . . . .	66
4.3.1	Aprendizado em Sistemas de IA Clássica . . . . .	66
4.3.2	Aprendizado em Sistemas Naturais . . . . .	68
4.4	Aprendizado e Sistema Nervoso Central . . . . .	72
4.5	Aprendizado no Contexto deste Trabalho . . . . .	74
<b>5</b>	<b>Evolução</b>	<b>75</b>
5.1	Introdução . . . . .	75
5.2	Bases Biológicas da Evolução . . . . .	76
5.2.1	Origens da Vida . . . . .	76
5.2.2	Evolução Biológica . . . . .	77
5.2.3	Fundamentos de Genética . . . . .	79
5.2.4	Genótipo e Fenótipo . . . . .	80
5.3	Evolução do Sistema Nervoso . . . . .	80
5.4	Evolução do Comportamento . . . . .	84
5.4.1	Sistema Nervoso e Comportamento . . . . .	84
5.4.2	Genética e Comportamento . . . . .	85
5.5	Computação Evolucionária . . . . .	86
5.5.1	Algoritmos Genéticos . . . . .	88
5.5.2	Algoritmo Genético Simples . . . . .	89
5.5.3	Programação Evolucionária . . . . .	94
5.5.4	Estratégias Evolucionárias . . . . .	95



5.5.5	Diferenças entre os Algoritmos Genéticos e outros Algoritmos Evolucionários . . . . .	97
5.6	Computação Evolucionária na Geração de Redes Neurais . . . . .	98
<b>6</b>	<b>Redes Neurais</b>	<b>103</b>
6.1	Introdução . . . . .	103
6.2	O Neurônio Biológico . . . . .	104
6.3	Redes de Neurônios - O Cérebro . . . . .	107
6.4	A Abordagem Conexionista . . . . .	108
6.4.1	Histórico . . . . .	108
6.4.2	O Modelo Geral de Neurônio Artificial . . . . .	110
6.4.3	Redes de Neurônios Artificiais . . . . .	114
6.5	Topologias de RNAs . . . . .	115
6.5.1	Redes Diretas . . . . .	116
6.5.2	Redes Recorrentes . . . . .	117
6.6	Alguns Teoremas . . . . .	120
6.7	Aprendizado de RNAs . . . . .	120
6.7.1	Aprendizado em Redes Neurais Diretas - o algoritmo 'Back-propagation' . . . . .	122
6.7.2	Aprendizado em Redes Neurais Recorrentes . . . . .	123
<b>7</b>	<b>Agentes Autônomos e Paradigmas Biológicos</b>	<b>124</b>
7.1	Introdução . . . . .	124
7.2	Idéias Gerais . . . . .	124
7.3	Capacidade de Generalização de RNAs . . . . .	127
7.3.1	O Problema do Aprendizado de Trajetórias . . . . .	129
7.3.2	Implementação dos Experimentos . . . . .	130
7.3.3	RNAs com Capacidade de Extrapolação . . . . .	134
7.4	Topologias de RNAs e suas Potencialidades Computacionais . . . . .	136
7.4.1	Complexidade de Redes Neurais . . . . .	137
7.4.2	Teoremas de Complexidade de Redes Neurais . . . . .	138
7.4.3	Classes de Problemas . . . . .	140
7.4.4	A Paridade como um problema de ordem infinita . . . . .	141
7.5	Implementando Comportamentos com RNAs . . . . .	143
7.5.1	Taxias e Reflexos . . . . .	143
7.5.2	Comportamentos Reativos e Instintivos . . . . .	146
7.6	Aprendizado em Rede Neurais Recorrentes . . . . .	151
7.6.1	Um Algoritmo do Tipo Backpropagation . . . . .	152

7.6.2	Um Algoritmo Biologicamente Inspirado . . . . .	154
7.7	Programação Evolucionária para a geração de Agentes Autônomos . .	159
7.7.1	Motivação . . . . .	159
7.7.2	O Sistema EVAG . . . . .	160
7.7.3	Experimentos . . . . .	163
7.8	Uma Rede que Aprende Estados Internos . . . . .	168
<b>8</b>	<b>Epílogo</b>	<b>182</b>
8.1	Conclusões e Trabalhos Futuros . . . . .	182
	<b>Referências Bibliográficas</b>	<b>206</b>

# Lista de Figuras

1.1	Comparação entre os principais paradigmas biológicos e computacionais envolvidos neste trabalho . . . . .	8
2.1	Funcionamento genérico de um Sistema Autônomo . . . . .	14
2.2	Arquitetura em camadas de comportamentos da “subsumption architecture” . . . . .	20
2.3	Esquema Inibidor/Supressor para uma camada de comportamento . . . . .	20
2.4	Módulo de controle motor implementado utilizando a arquitetura de quadro-negro . . . . .	25
2.5	Exemplo de sistema funcional . . . . .	31
2.6	Grafo representando um autômato com dois estados . . . . .	35
2.7	Hierarquia de tipos de sistemas . . . . .	37
2.8	Exemplo de sistema complexo . . . . .	38
2.9	Diagrama de Estados de um AA antes e após um aprendizado . . . . .	41
2.10	Aprendizado como uma transição “especial” de estados . . . . .	42
3.1	Representação esquemática do relacionamento entre hereditariedade, sistema nervoso e aprendizado na determinação do comportamento de um animal . . . . .	49
3.2	Apresentação esquemática da forma de comportamento dominante em série filogenética . . . . .	51
3.3	Um problema de contorno no qual o animal deve primeiramente se afastar do alimento a fim de atingi-lo . . . . .	56
3.4	Máquina de Mealy representando o comportamento de mover-se em direção ao alimento . . . . .	60
3.5	Autômato de Estados Finitos que descreve de modo geral o comportamento da vespa descrito no Exemplo 3.3.1 . . . . .	61
3.6	Dois Autômatos de Estados Finitos descrevendo em diferentes níveis de detalhamento o comportamento de uma barata que procura por um ambiente escuro e ao mesmo tempo foge de perigo . . . . .	62

3.7	Autômato de Estados Finitos que descreve um acasalamento movido por comportamento instintivo . . . . .	63
4.1	Exemplo de uma seqüência de condicionamento clássico . . . . .	70
4.2	Gatos na caixa de Skinner; um deles a ponto de calcar a barra . . . . .	71
4.3	Três labirintos de dificuldade crescente. S é o local de partida e F o local de alimento . . . . .	72
5.1	Ciclo de vida do sistema nervoso apresentando os mecanismos de desenvolvimento, aprendizado, seleção natural e variação genética . . . . .	78
5.2	Exemplo de roleta para seleção . . . . .	91
5.3	Processo conceitual da operação de cruzamento . . . . .	92
5.4	Processo conceitual da operação de mutação . . . . .	92
5.5	Exemplo de um ciclo do Algoritmo Genético Simples . . . . .	93
6.1	Representação simplificada de um Neurônio Biológico . . . . .	106
6.2	Elementos básicos que compõem um Neurônio Artificial . . . . .	112
6.3	Rede Neural Artificial Direta Multi-Camada . . . . .	117
6.4	Rede Neural Artificial Recorrente com uma única camada de neurônios	118
6.5	Um exemplo de rede recorrente de tempo-real . . . . .	119
7.1	Diagrama das relações entre neurônios efetivamente observados em um sistema nervoso biológico . . . . .	126
7.2	Aumento da complexidade do sistema nervoso através de processos evolucionários . . . . .	127
7.3	Rede Neural Direta para geração de trajetória balística . . . . .	130
7.4	Trajetoárias, real e calculada pela rede para ângulos de lançamento de 30°, 45° e 60° . . . . .	131
7.5	Erro máximo entre as trajetórias reais e calculadas pela rede para ângulos de lançamento entre 0° e 90°, com rede treinada para velocidade inicial de 10m/s e ângulos de lançamento de 30° e 60° . . . . .	132
7.6	Erro máximo entre as trajetórias reais e calculadas pela rede para ângulos de lançamento entre 0° e 90°, com rede treinada para velocidade inicial de 10m/s e ângulos de lançamento de 15° e 75° . . . . .	133
7.7	Autômato finito que implementa o cálculo da paridade de um "string" . . . . .	135
7.8	Rede Neural Artificial Recorrente que modela o autômato finito que implementa o cálculo da paridade de um "string" . . . . .	135
7.9	Modelo de Rede Neural Recorrente capaz de implementar um autômato finito . . . . .	140

7.10	Rede Neural Artificial Recorrente sem camada intermediária para tentar implementar o autômato finito que implementa o cálculo da paridade de um “string” . . . . .	142
7.11	Rede Neural Direta equivalente à rede recorrente sem camada intermediária incapaz de aprender o autômato finito da paridade . . . . .	142
7.12	Rede Neural capaz de implementar o comportamento de “movimento em direção ao alimento” . . . . .	144
7.13	Criatura primitiva com sistema nervoso composto apenas de conexões diretas, porém com uma camada de neurônios intermediários . . . . .	146
7.14	Rede Neural Recorrente com camada escondida capaz de aprender os estados de um Autômato de Estados Finitos . . . . .	147
7.15	Rede Neural Direta capaz de aprender as saídas de um Autômatos de Estados Finitos . . . . .	148
7.16	Representação simplificada da formiga simulada com seus sensores e atuadores . . . . .	149
7.17	Plano toroidal onde a formiga é posta para operar e trajetória seguida pela formiga . . . . .	150
7.18	Autômato de Estados Finitos que descreve um possível comportamento para a formiga . . . . .	151
7.19	Rede Neural Artificial Composta que implementa o comportamento descrito para a formiga . . . . .	152
7.20	Autômato finito que descreve o comportamento do agente que resolve o problema das lâmpadas e dos botões . . . . .	154
7.21	Rede Neural Direta equivalente à rede recorrente capaz de aprender o autômato finito da lâmpadas e botões através do algoritmo de back-propagation . . . . .	156
7.22	Configuração inicial do sistema nervoso do agente no instante em que ele é posto para operar no ambiente . . . . .	162
7.23	Ambiente simples contendo agentes e alimentos aleatoriamente distribuídos . . . . .	164
7.24	Rede Neural Feedforward desenvolvida pelo sistema que permite ao agente percorrer o ambiente coletando alimentos . . . . .	165
7.25	Ambiente contendo agentes, alimentos e venenos aleatoriamente distribuídos . . . . .	166
7.26	Algumas soluções encontradas pelo sistema para percorrer o ambiente coletando alimentos ao mesmo tempo em que evita consumir veneno . . . . .	167

7.27 Ambiente cercado por paredes contendo agentes e alimentos aleatoriamente distribuídos . . . . .	168
7.28 Rede Neural Recorrente encontrada pelo EVAG para um agente capaz de operar no ambiente cercado por paredes . . . . .	169
7.29 Interface de operação do sistema BIONNT . . . . .	172
7.30 Rede Neural Recorrente com camada intermediária e neurônios de estado na camada de saída capaz de aprender linguagens regulares . .	173
7.31 Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 1 da Tabela 7.8. Os gráficos são plotados a cada 50 épocas de treinamento . . . . .	178
7.32 Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 2 da Tabela 7.8 . . . . .	179
7.33 Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 3 da Tabela 7.8 . . . . .	180
7.34 Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 4 da Tabela 7.8 . . . . .	180
7.35 Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 5 da Tabela 7.8 . . . . .	181
7.36 Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 6 da Tabela 7.8 . . . . .	181

# Lista de Tabelas

7.1	Conjunto de treinamento para o autômato que implementa o cálculo da paridade . . . . .	143
7.2	Função combinacional que descreve o comportamento de taxia . . . . .	145
7.3	Função combinacional que descreve o novo comportamento . . . . .	145
7.4	Conjunto de treinamento para o AEF que implementa o comportamento das lâmpadas e dos botões . . . . .	155
7.5	Conjunto de treinamento para os estados do AEF que implementa o comportamento da formiga simulada . . . . .	158
7.6	Conjunto de treinamento para os valores de saída do AEF que descreve o comportamento da formiga simulada . . . . .	159
7.7	Caracterização de algumas das linguagens propostas por Tomita [182]	170
7.8	Exemplos positivos e negativos de algumas linguagens investigadas em Tomita [182]. $\xi$ significa a cadeia vazia . . . . .	171
7.9	Resumo dos resultados de velocidade e acuidade reportados por [188] e [9] para aprendizado das linguagens da Tabela 7.7 . . . . .	174
7.10	Resultados de configuração, velocidade, erro mínimo obtido ao final do aprendizado e acuidade das redes recorrentes para aprendizado das linguagens da Tabela 7.7 . . . . .	175
7.11	Conjunto de treinamento para implementar o comportamento de andar pelo ambiente evitando elementos de veneno colocados sempre “atrás” dos alimentos . . . . .	176

# Lista de Abreviaturas

AA	Agente Autônomo
ADN	Ácido Desóxi-Ribonuclêico
AE	Algoritmo Evolucionário
AG	Algoritmo Genético
AGS	Algoritmo Genético Simples
ARN	Ácido Ribonuclêico
BAM	Bi-Directional Associative Memory
CBI	Computador Baseado em Instrução
CRN	Computador Baseado em Redes Neurais
EC	Estímulo Condicionado
EE	Estratégia Evolucionária
ENC	Estímulo Não-Condiciondo
EVAG	Evolutionary Agents Generator
NA	Neurônio Artificial
PDP	Parallel and Distributed Processing
PE	Programação Evolucionária
RC	Reflexo Condicionado
RNA	Rede Neural Artificial
RNC	Reflexo Não-Condicionado
RNN	Rede Neural Natural
SNC	Sistema Nervoso Central



# Resumo

O desenvolvimento de *Agentes Autônomos* que sejam capazes de sobreviver e de desenvolver tarefas em ambientes complexos, dinâmicos, imprevisíveis, desestruturados e até mesmo hostis, como o *mundo real*, é tarefa extremamente desafiadora. Por outro lado, seres vivos, sejam eles mamíferos, aves, insetos ou mesmo vermes, apresentam uma enorme capacidade de *sobrevivência* no mundo real quando comparados com qualquer Agente Autônomo já desenvolvido.

Este trabalho propõe que a *inspiração biológica* pode ser a fonte de mecanismos e soluções que, uma vez entendidos e implementados, permitam construir Agentes Autônomos com alto grau de autonomia e utilidade.

*Inteligência Computacional*, neste trabalho, é entendida como *comportamento adaptativo* que permite a *sobrevivência* do sistema no ambiente onde ele é posto para operar. Ao buscarmos uma inspiração biológica, podemos considerar que os *Algoritmos Evolucionários* fornecem o modelo computacional que simula, mesmo que de maneira simplista, os processos naturais de *reprodução, mutação e seleção* que simulam a evolução de estruturas individuais no processo de busca por indivíduos mais aptos a sobreviverem e agirem em um dado ambiente. Quanto à questão de modelagem de aspectos do comportamento dos animais, a *abordagem Conexionista* é considerada e utilizada para simular aspectos comportamentais na implementação de Agentes Autônomos.

Este trabalho apresenta a idéia de que uma série de classes de comportamentos observados nos animais podem ser implementados através de diferentes arquiteturas de *Redes Neurais Artificiais*. Também se enfatiza o fato de que estas diferentes arquiteturas podem ser obtidas através de Algoritmos Evolucionários. Procura-se comprovar a hipótese de que *comportamentos reflexivos* podem ser implementados através de arquiteturas mais simples de redes neurais *estáticas*, do tipo *direta* ou *"feedforward"*. Já comportamentos mais complexos, tais como os *comportamentos reativos*, que persistem e se desenrolam mesmo após ter cessado o estímulo sensorial excitador, necessitam de redes neurais realimentadas ou *recorrentes*, com neurônios dinâmicos, de modo a inserir dinâmica e memória no sistema.

Inicialmente, são apresentados os fatores motivadores e alguns conceitos básicos. Logo após, apresenta-se o conceito de Agente Autônomo e as principais alternativas de implementação e propõe-se um formalismo, baseado na *Teoria Geral de Sistema* para conceituação de Agentes Autônomos. A seguir, os conceitos de comportamento, aprendizado, evolução e sistema nervoso são descritos e analisados, tanto do ponto de vista biológico como dos respectivos paradigmas computacionais. Finalmente, são descritas algumas contribuições originais que têm como base os temas estudados anteriormente e que permitem uma nova alternativa para implementação de Agentes Autônomos e são apresentadas implementações concretas de exemplos simples que ilustram as idéias apresentadas neste e nos capítulos anteriores. No epílogo é apresentado um resumo final do que foi discutido e são listados pontos em aberto para trabalhos futuros.

## Palavras Chaves:

Agentes Autônomos, Redes Neurais, Algoritmos Evolucionários, Vida Artificial, Robótica.

# Abstract

## Title: Emergence of Intelligence in Autonomous Agents through Nature Inspired Models

The development of *Autonomous Agents* capable to survive and develop tasks in a complex, dynamic, unforeseeable and even hostile environment, as is the *real world*, is a very challenging research area. Life beings, as mammals, birds, insects or even worms, present a bigger *survival* capacity in the real world when compared with any Autonomous Agent ever developed.

This work proposes that *biological inspiration* can be the source of mechanisms and solutions that, once understood and implemented, allow the development of Autonomous Agents with high level of autonomy and utility.

*Computational Intelligence*, in this work, is understood as *adaptive behavior* that allows the *system survival* and operation in a given environment. Looking for a biological inspiration, it is considered that the *Evolutionary Algorithms* give the computational model that simulates, even in a simplistic fashion, the natural procedures of *reproduction, mutation and selection* that allow the evolution of individual structures searching the most capable solution for a given environment. What concerns about modeling aspects of animal behaviors, we consider and use the *Connectionist approach* as an appropriate tool to model and simulate behavioral aspects when implementing Autonomous Agents.

This work presents the idea that a series of behavior classes, observed in animals, can be implemented through different *Artificial Neural Networks* architecture's. We also emphasize that these different architectures can be obtained using Evolutionary Algorithms. We try to confirm the hypothesis that pure *reflexive behaviors* could be implemented through simple *static* neural networks architecture's like *feedforward* architectures. More complex behaviors, as reactive behaviors, that persist and develop, even after finished the sensorial exciting stimulus, require more complex neural networks architecture's, like a *recurrent* one, with dynamic neurons to insert dynamic and memory in the system.

Firstly, motivational facts and some basic concepts are presented. Just afterwards the concepts of Autonomous Agents and the main implementations strategies are presented as well as a formal conceptuation, using the *General Systems Theory* is proposed. After that, the concepts of behavior, learning, evolution and nervous system, are described and analysed, both from the biological as from the equivalent computational point-of-view. Finally, some original contributions based in the previously presented subjects and that allow a new strategy in Autonomous Agents implementation are described. Some implementations of simple examples that illustrate the ideas are presented. A summary of what had been discussed is presented and open points for future works are listed.

## **Keywords:**

Autonomous Agents, Neural Networks, Evolutionary Algorithms, Artificial Life, Robotics.

# Capítulo 1

## Introdução

### 1.1 Motivação

Quando se observa o mundo ao nosso redor, um fato extraordinário muitas vezes passa despercebido diante de nossos olhos: a sobrevivência. Uma minhoca, uma barata, um gato e mesmo os seres humanos são máquinas extraordinárias que conseguem interagir com o ambiente que os cerca, retirando deste ambiente os recursos necessários à manutenção da vida, crescendo e se reproduzindo. Para que estes seres consigam meios para sua sobrevivência, três capacidades parecem fundamentais: a percepção, a mobilidade e a inteligência. A percepção do mundo, a mobilidade e a inteligência parecem ter evoluído juntas. Na verdade, estes três conceitos estão intimamente ligados e um dificilmente poderia existir sem o outro.

Podemos entender a percepção como sendo o canal pelo qual uma entidade obtém informações sobre o que ocorre no ambiente ao seu redor. Já a mobilidade é a capacidade da entidade de se mover, explorar e alterar o ambiente que a cerca. Entre os seres vivos, os vegetais parecem possuir uma capacidade de percepção do ambiente que os cerca (temperatura, dia/noite, etc), no entanto sua mobilidade, da maneira como a definimos, é extremamente limitada, para não se dizer inexistente. Com os animais, os conceitos de percepção e mobilidade são mais evidentes e desenvolvidos. Em muitos deles pode-se mesmo identificar sistemas especializados para estas funções, tais como os nervos e os músculos.

Assim como na área da teoria da computação, a definição de função computável corresponde à existência de um algoritmo capaz de calcular o seu valor, poderia se dizer que no caso dos seres vivos, a existência de uma função com característica semelhante à função computável pode ser comprovada pela existência de um algoritmo que permite ao ser-vivo gerar um comportamento adequado e a garantir sobrevivência de sua informação genética. Ao se procurar fazer uma analogia, ainda

que superficial, com os sistemas computacionais, poder-se-ia comparar o ambiente externo com a fita de entrada de onde são lidas as instruções a serem executadas (percebidas) pelo processador (a entidade viva que age no ambiente). Assim, os vegetais poderiam ser comparados com as Máquinas Seqüenciais que processam seqüencialmente as instruções lidas da fita de entrada para produzir um resultado (a sobrevivência), enquanto que os animais poderiam ser comparados a uma Máquina de Turing, na qual a percepção seria a capacidade de ler a fita de entrada, enquanto que, a mobilidade, a sua capacidade de alterar o ambiente, poderia ser associada à capacidade da Máquina de Turing de escrever na fita de entrada (o ambiente).

Assim como, computacionalmente, uma Máquina de Turing é mais potente que uma Máquina Seqüencial, a mobilidade aumenta, em muito, as chances de sobrevivência dos seres vivos, pela modificação do ambiente, através da fuga de predadores, busca de alimentos, procura por elementos da mesma espécie, construção de ninhos e abrigos, etc.

Enquanto os conceitos de percepção e mobilidade são relativamente bem entendidos, o conceito de inteligência é muito mais aberto e vago. Fogel [66] apresenta uma discussão bastante interessante sobre este tema. Em um primeiro momento, a inteligência era geralmente associada a uma característica unicamente humana, de representação de conhecimentos e resolução de problemas, refletindo um ponto de vista altamente antropocêntrico. Mas, ainda assim, nós, humanos, não compreendemos a nós mesmos, como funciona nossa “inteligência” e nem mesmo a origem de nossos pensamentos [13]. Minsky [126] apresentou a seguinte definição de inteligência “Inteligência ... significa ... a habilidade de resolver problemas difíceis”. Esta definição, apesar de bastante geral, também é problemática. Um computador, programado para resolver um “problema difícil” como, por exemplo, as raízes de um polinômio de grau  $n$ , não pode ser considerado como possuindo inteligência.

Passou-se então a associar a idéia de inteligência com a capacidade de tomar decisões. Fogel [66] coloca que “... um organismo, ou qualquer sistema, para ser inteligente, ele deve tomar decisões. Qualquer decisão deve ser descrita como a seleção de como alocar os recursos disponíveis. ...a tomada de decisões requer um objetivo”. Note-se, no entanto, que muitas vezes o comportamento de um sistema pode não envolver a tomada de decisões, mas sim o reconhecimento de um padrão, por exemplo, um pato que ao nascer associa o primeiro objeto de grande porte que ele enxerga como sendo sua “mãe” e o segue por toda a parte. Este comportamento não envolve uma tomada de decisão, mas é um comportamento “inteligente” que foi “adquirido” geneticamente ao longo de gerações e que aumenta a chance de sobrevivência da espécie pela proteção que o ser maior (provavelmente a mãe) oferece.

Assim, a idéia de inteligência passou a ser associada com a idéia de sobrevivência. Carne [45] define: “Talvez a característica básica de um organismo inteligente seja sua capacidade de aprender a realizar várias funções em um ambiente dinâmico, tais como sobreviver e prosperar”. E Fogel [66] generaliza: “inteligência pode ser definida como a capacidade de um sistema de adaptar seu comportamento para atingir seus objetivos em uma variedade de ambientes”.

Seguindo esta linha de raciocínio, vê-se que, no caso dos seres vivos, a inteligência reside no sistema nervoso e está associado a sua complexidade, ou seja, quanto mais complexo o sistema nervoso do organismo, maior a capacidade de adaptar o seu comportamento e, portanto, maior a sua inteligência. Para o caso de sistemas artificiais, poder-se-ia associar a inteligência à capacidade do processador dos sistemas computacionais. Assim, a função primordial da inteligência é de interligar de maneira coerente a percepção com a mobilidade, ou seja, sua função é processar corretamente os dados que estão sendo lidos da fita de entrada (percepção do ambiente) e gerar os resultados de saída coerentes que, por sua vez, de alguma maneira, são escritos na fita de entrada (através da modificação do ambiente), resultando em novos dados e em novas ações que, assim, se transformam em um imenso programa que dura enquanto o processador subsistir (enquanto o ser estiver vivo).

No entanto, a capacidade do processador não é tudo, é necessário que os “programas” que ele executa, assim como os circuitos e estruturas neurais dos sistemas nervosos naturais, sejam adequados à complexidade do processador. Um supercomputador rodando um algoritmo ineficiente ou um cérebro humano com os circuitos avariados ou que não tenha a oportunidade de aprender novos comportamentos, têm seu grau de “inteligência” bastante reduzido.

Assim, a inteligência é um fator que depende tanto da capacidade do sistema nervoso como das informações que este sistema é capaz de processar ou aprender. No caso de sistemas artificiais, quanto melhor o processador e mais adequado o “software” que ele executa maior o grau de inteligência e portanto, melhor a coordenação entre percepção e mobilidade e, maiores as chances de sobrevivência da entidade. Ou seja, quanto mais inteligente a entidade, mais adequada é a ação gerada para uma determinada percepção.

A partir daí, podemos associar este conceito amplo de inteligência à algumas de suas características, tais como a adaptação, o aprendizado, a criatividade, a resolução de problemas, etc, que têm sido objeto de pesquisas de diversas áreas de estudos.

O mecanismo de sobrevivência dos animais em um ambiente é um ciclo fechado. Para poder sobreviver, um animal necessita de certa inteligência a fim de coordenar

de maneira adequada o funcionamento do seu sistema nervoso sensório-motor. Por outro lado, é através da mobilidade conferida pelo sistema sensório-motor que os animais aplicam a sua inteligência na busca pela adaptação ao ambiente [150].

A inteligência do ser humano, aliado a sua mobilidade e criatividade, o levou a desenvolver mecanismos que, uma vez dotados de inteligência e mobilidade, viessem a substituí-lo nas tarefas repetitivas, desgastantes ou perigosas. Estes mecanismos são conhecidos como Agentes ou Sistemas Autônomos.

Agentes Autônomos (AAs) são sistemas que possuem uma interação duradoura com um ambiente dinâmico externo, estando normalmente instalados fisicamente em mecanismos. Eles são dotados de rodas ou outros meios de locomoção e sensores e projetados para funcionarem por longos períodos de tempo, operando em um ciclo que envolve a aquisição de informações do ambiente e a geração de dados de saída. Na sua interação com o ambiente externo, os AAs podem influenciar a dinâmica deste ambiente, porém dificilmente podem controlá-lo completamente [90]. A operação destes sistemas envolve principalmente problemas navegacionais, tais como fugir de obstáculos, vagar no ambiente sem esbarrar em objetos, atingir uma posição determinada, etc.

Os AAs encontram numerosas aplicações nos dias de hoje. Entre elas podemos citar: o transporte de peças e ferramentas nos ambientes industriais, escavação e transporte de minérios em minas subterrâneas, exploração submarina, manutenção de equipamentos petrolíferos a grandes profundidades, exploração espacial, investigação de vulcões, construção e manutenção de estradas, vigilância, etc [31] [50] [38] [12].<sup>1</sup>

Nas aplicações descritas, os agentes operam no mundo real, ou seja, em ambientes chamados "Ambientes Abertos". Um Ambiente Aberto é aquele que não é cem por cento conhecido, podendo apresentar obstáculos e eventos imprevisíveis. A operação dos AAs nestes ambientes está associada a uma série de problemas. Em primeiro lugar é impossível descrever completamente o ambiente e os obstáculos que o agente irá encontrar; os ambientes abertos apresentam constantemente eventos e situações imprevisíveis que podem levar a replanejamentos e novas ações. Desenvolver Agentes Autônomos capazes de realizar suas tarefas operando em ambientes abertos, dinâmicos e desestruturados, de maneira autônoma, tem sido um desafio para muitos pesquisadores de áreas como Robótica e Inteligência Artificial [34] [104] [105] [31] [106] [90] [113].

Os AAs desenvolvidos por estes pesquisadores apresentam uma performance de

---

<sup>1</sup>Ao longo deste texto, quando mais de uma obra for citada, o critério de ordem adotado é o de grau de importância ou pertinência da referência com o assunto em questão.



operação em tempo-real no mundo real infinitamente inferior quando comparada à dos organismos biológicos. Assim, se os mecanismos utilizados pela Natureza foram capazes de desenvolver sistemas (os animais) que apresentam uma extraordinária performance de operação e sobrevivência, parece interessante e promissor tentar estudar e utilizar estes mecanismos naturais a fim de desenvolver modelos que, assim como os animais, funcionariam muito bem no mundo real. Nesta área, conhecida de maneira geral como Vida Artificial (“Artificial Life”), encontram-se diversos pesquisadores trabalhando em diferentes aspectos desta abordagem [8] [28] [190] [103], entre outros. A corrente de pensamento destes autores parece ver em um modesto inseto, no lugar das habilidades simbólicas de um especialista, o melhor protótipo para o que eles consideram como inteligência [32].

Brooks em [37] também apresenta um argumento bastante interessante. Ao refletir como se deu a evolução biológica sobre a Terra, vê-se que os primeiros peixes e vertebrados surgiram há 550 milhões de anos atrás enquanto que o homem chegou há apenas 2.5 milhões. Além disso, a agricultura foi inventada há 19000 anos, a escrita há menos de 5000 e o conhecimento “especialista” apenas há umas poucas centenas de anos. Isto sugeriria que a resolução de problemas, a linguagem, o conhecimento especialista e o raciocínio seriam mais simples de serem obtidos, uma vez que as capacidade de ser e de reagir estivessem disponíveis. A essência seria a habilidade de se mover em um ambiente dinâmico, interagindo com o ambiente em um grau suficiente para a manutenção da vida e da reprodução, uma vez que foi nesta parte da inteligência que a Natureza concentrou a maior parte do seu tempo.

Segundo Langton, a Vida Artificial é o campo de estudo dedicado a tentar abstrair a dinâmica fundamental dos princípios que regem os fenômenos biológicos e que procura recriar esta dinâmica em outro meio físico - os computadores - tornando-os acessíveis a novas formas de manipulação experimental e testes [103].

Porque procurar na Natureza, em especial nos fenômenos biológicos, os paradigmas para o desenvolvimento de sistemas computacionais mais eficazes e capazes de resolver os problemas propostos?

A motivação para tomar esta inspiração biológica vem da suposição que a Natureza, com seus mecanismos, buscou resolver um problema de otimização. A idéia cada vez mais aceita hoje em dia, é de que a Natureza procura um ótimo. Pesquisadores como Rashevsky [147] e Rosen [160] têm procurado mostrar como os fenômenos biológicos obedecem a leis perfeitamente expressas de forma matemática, bem como a importância dos conceitos de otimização em biologia.

Observando a Natureza, vemos que os mecanismos de evolução e seleção natural possibilitaram o surgimento de seres cada vez melhor adaptados a sobreviverem em

um determinado ambiente. A evolução é o mecanismo pelo qual se obtém a diversidade de organismos biológicos que interagem com o meio ambiente, retirando deste meio as suas condições de vida e, por sua vez, modificando-o. Como os organismos são diferentes, os mais aptos encontram condições de vida e possibilidade de reprodução maiores que outros menos aptos e, assim, seu número tende a aumentar pelo processo chamado seleção. O mecanismo de seleção pode ser considerado como uma busca constante de maior eficiência.

Pode-se constatar, também, que este processo evolutivo, na busca por elementos cada vez mais aptos, levou ao desenvolvimento de organismos com um sistema nervoso. Este sistema, conectado a toda uma gama de sensores, realiza um processamento de modo a atuar sobre um sistema motor. Este sistema nervoso, composto por uma rede de células especializadas, chamadas neurônios, mapeia de maneira coerente as informações do ambiente e do próprio agente, recebidas pelos sensores, em ações executadas pelos atuadores (sistema motor). Este mapeamento cria o que poderíamos chamar de comportamentos. Além disso, este sistema nervoso tem condições, em maior ou menor grau, de se adaptar a mudanças do ambiente, num processo que poderíamos chamar de aprendizado, de modo a melhorar a operação do agente durante o seu período de existência.

Uma vez estabelecido que a inspiração biológica pode ser a fonte de mecanismos e soluções que, uma vez implementadas, permitem alcançar um alto grau de autonomia, devemos verificar como estes paradigmas biológicos podem ser transformados em ferramentas para a construção de AAs.

Na área da Inteligência Artificial, vemos se popularizar a cada dia o conceito de Computação Evolucionária. Computação Evolucionária é o nome genérico dado à métodos computacionais inspirados na teoria da evolução. Os algoritmos usados em computação evolucionária se chamam Algoritmos Evolucionários (AEs) [19].

Existe uma gama enorme de algoritmos inspirados na teoria da evolução. Os mais conhecidos são os Algoritmos Genéticos propostos por Holland [86], certamente o mais popular deles, a Programação Evolucionária, as Estratégias Evolucionárias e a Programação Genética. Todos são inspirados nos mecanismos de evolução biológica e portanto usam mecanismos de seleção, mutação e reprodução que permitem a evolução de estruturas individuais na procura pela solução mais adequada para um dado ambiente.

No caso de AAs, é interessante notar que o “problema” que se quer resolver é o da “capacidade de sobrevivência” do agente e a sua habilidade para realizar determinada tarefa em um dado ambiente aberto. Este problema dificilmente poderia ser descrito por uma função ou um conjunto de equações. No entanto, se estamos

interessados em obter o “melhor” agente como solução para este problema, ou seja, o agente que tiver melhor aptidão para sobreviver no ambiente enquanto realiza a tarefa especificada, estamos diante de um problema de otimização, mesmo que o que se esteja otimizando seja desconhecido. Ora, historicamente, os algoritmos evolucionários e suas variações surgiram justamente como ferramenta para solução de problemas em que se busca uma solução ótima apesar de não se conhecer explicitamente o que se está otimizando.

Se os AEs são paradigmas da área de Inteligência Computacional que procuram buscar inspiração nos mecanismos naturais de evolução e seleção para a solução de problemas, é a abordagem conexionista através das Redes Neurais Artificiais (RNAs), por sua vez, que procura modelar o funcionamento do sistema nervoso, simulando o funcionamento de sua unidade básica, o neurônio com suas conexões.

Rumelhart et al. [162] em seu excelente texto introdutório, afirmam que a estratégia até hoje adotada tem sido a de desenvolver modelos matemáticos simplificados de sistemas nervosos e de estudar estes modelos a fim de entender como vários problemas computacionais podem ser resolvidos por estes modelos. Este trabalho tem atraído pesquisadores de diversas áreas: neurocientistas interessados em desenvolver modelos dos circuitos neurais encontrados em áreas específicas do cérebro de diversos animais; físicos que vêem analogias entre o comportamento dinâmico de sistemas nervosos e vários tipos de sistemas dinâmicos não-lineares muito comuns na física; engenheiros de computação interessados em fabricar “computadores neurais”; pesquisadores da área de inteligência artificial interessados em construir máquinas com a inteligência de organismos biológicos; engenheiros interessados em resolver problemas práticos; psicólogos interessados em compreender os mecanismos humanos de processamento de informações; matemáticos interessados nas características matemáticas destes sistemas neurais; filósofos interessados em entender como estes sistemas podem mudar nosso ponto de vista sobre a natureza da mente e sua relação com o cérebro; e muitos outros.

É importante ressaltar que se os mecanismos evolutivos encontrados na Natureza são relativamente bem compreendidos, os mecanismos de funcionamento do sistema nervoso nos seres vivos parecem ser bem mais complexos e muitos são ainda desconhecidos. Portanto, ainda hoje em dia, os modelos de RNAs estão muito distantes das redes neurais naturais (RNNs) e, freqüentemente, as semelhanças são mínimas.

Apesar das simplificações adotadas, pesquisas têm procurado modelar o sistema nervoso com suficiente precisão de tal modo a poder observar um comportamento emergente que sendo semelhante ao comportamento do ser vivo modelado, possa servir de apoio às hipóteses usadas na modelagem. Através de certos modelos de

RNAs é possível criar mecanismos de adaptação que simulam o processo de aprendizagem. No que diz respeito a modelagem de aspectos do comportamento dos seres vivos, também podemos considerar a abordagem conexionista como uma ferramenta apropriada para modelar e simular aspectos comportamentais na implementação de AAs.

Resumindo, pode-se dizer que ao longo deste trabalho procura se estudar como fazer emergir aspectos de inteligência em AAs através de modelos inspirados na Natureza, tais como os AEs e as RNAs.

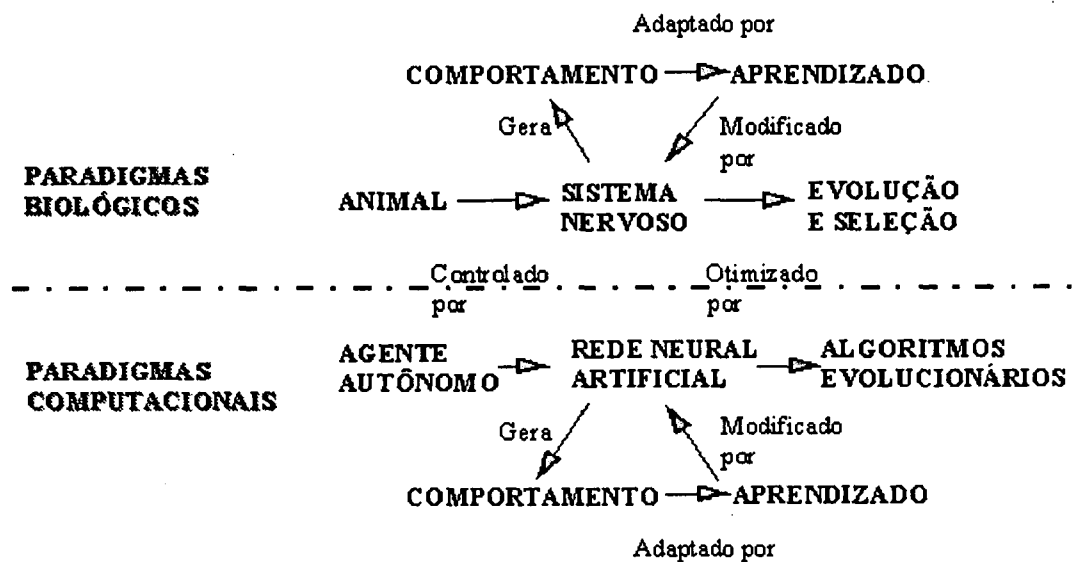


Figura 1.1: Comparação entre os principais paradigmas biológicos e computacionais envolvidos neste trabalho

## 1.2 Objetivos da Tese

Nesta tese se busca mecanismos de criação de AAs que sejam adaptáveis e que possam evoluir e ter uma boa performance em um ambiente não especificado a priori. Estes mecanismos de criação de AAs são inspirados nos mecanismos encontrados na Natureza tais como evolução e aprendizado, com a conseqüente adaptação ao meio.

Especificamente, este trabalho pretende:

- Apresentar e analisar os pontos principais da Natureza sobre os quais se buscou inspiração para o desenvolvimento de AAs;

- Apresentar e analisar os modelos computacionais utilizados para simular os paradigmas naturais;
- Apresentar e utilizar a abordagem de Teoria de Sistemas para construir uma base formal para a definição de Agente Autônomo e de outros conceitos relacionados;
- Uma vez estabelecida esta base teórica, mostrar como uma ferramenta formal bem conhecida, tal como o Autômato de Estados Finitos, pode ser utilizada para descrever um Agente Autônomo;
- Propor como os Agentes Autônomos assim descritos podem ser implementados por Redes Neurais Artificiais utilizando o mesmo arcabouço teórico;
- Introduzir e fundamentar a idéia de que uma série de classes de comportamento observadas em animais podem ser implementados através de diferentes arquiteturas de RNAs;
- Desenvolver algoritmos de aprendizado que sejam biologicamente inspirados para a implementação de comportamentos aprendidos;
- Demonstrar que as diferentes arquiteturas de RNAs podem ser obtidas através de AEs;
- Relacionar as diferentes arquiteturas obtidas através de AEs com o grau de complexidade dos comportamentos necessários para a sobrevivência do agente em diferentes ambientes;
- Ilustrar os idéias apresentadas com a implementação concreta de exemplos simples.

### 1.3 Organização do Texto

Este trabalho está dividido em 8 capítulos. O capítulo 1 descreveu o contexto geral no qual o trabalho está inserido. Também foram apresentados os objetivos desta tese.

O capítulo 2 procura apresentar e formalizar o conceito de Agentes Autônomos. Inicialmente são apresentados: o conceito básico de AA e as abordagens tradicionais já utilizadas no desenvolvimento destes sistemas, tais como programação direta, técnicas de IA simbólica, “subsumption architecture”, etc., analisando características e limitações. Em seguida, procura-se formalizar os modelos de AAs e de

Ambiente, bem como as interações entre estes dois sistemas, utilizando a abordagem de Teoria de Sistemas proposta por Barreto [15] e de Azevedo [59] e que é inspirada pelos trabalhos de Bertalanfy [186] (biologia), Zadeh & Desoer [197] (engenharia), Kalman, Arbib & Falb [93] (matemática) e Mesarovic & Takahara [123] (modelos populacionais). Finalmente apresenta-se critérios para a avaliação da performance do Agente no Ambiente.

Ao longo deste trabalho, procura-se mostrar que o comportamento dos animais, um dos fatores de inspiração por nós utilizados para o desenvolvimento de AAs, é influenciado pela hereditariedade e pela aprendizagem. Conforme Hebb, a hereditariedade, ao longo de um processo evolutivo, determina o tipo de cérebro, órgãos sensoriais e sistema motor, dos quais a capacidade de aprendizado depende inteiramente. Devido às diferenças na hereditariedade, uma pessoa pode aprender o que seria impossível a um elefante; os cães, os gatos e os macacos aprendem da mesma situação coisas diferentes, e assim por diante [81]. Deste modo, o capítulo 3 descreve as diferentes classes de comportamentos animais encontrados na Natureza, tais como comportamentos reflexos, inatos, aprendidos, etc. Mostra-se também que certos comportamentos podem ser descritos através de autômata de estados finitos.

O objetivo primordial do aprendizado em nosso trabalho é o de aprimorar o comportamento do sistema, tornando-o mais apto para operar no ambiente em que for colocado. Assim, o capítulo 4 procura fazer uma revisão no conceito de aprendizado, apresentando os diversos conceitos encontrados na literatura tanto na área da psicologia como na área de computação.

O capítulo 5 trata da questão da evolução, descrevendo inicialmente os mecanismos naturais da evolução e seleção natural, apresentando hipóteses de como este processo levou ao desenvolvimento do sistema nervoso e à evolução dos comportamentos. Finalmente, neste capítulo são apresentados os principais métodos computacionais inspirados nos mecanismos biológicos de evolução e como eles podem ser utilizados para gerar diferentes arquiteturas de RNAs.

No capítulo 6 é feito um estudo sobre as Redes Neurais. Primeiramente são apresentados os modelos biológicos de Sistema Nervoso Central, o funcionamento dos neurônios naturais e a organização interna do cérebro. Em seguida são apresentados os modelos artificiais de neurônios e de RNAs, tanto estáticas como dinâmicas, bem como os mecanismos de aprendizado utilizados. Procura-se relacionar as arquiteturas de redes neurais apresentadas com a capacidade de resolução de diferentes classes de problemas. Vemos que sistemas dinâmicos necessitam de redes neurais dinâmicas para serem modelados, o que nos leva a apresentar a hipótese de que diferentes classes de comportamentos necessitam de diferentes arquiteturas de RNAs.

Finalmente, no capítulo 7 são descritas algumas contribuições originais que têm como base os temas estudados anteriormente e que permitem uma nova alternativa para implementação de AAs. Estas contribuições envolvem: uma reflexão sobre as capacidades de generalização das RNAs em função dos exemplos apresentados durante o aprendizado e da própria estrutura da RNA; um estudo sobre como a estrutura da RNA determina suas potencialidades computacionais; uma proposta de implementação de comportamentos observados na natureza através de diferentes estruturas de RNAs, bem como mecanismos biologicamente inspirados para o aprendizado de RNAs; a descrição de um sistema baseado em Programação Evolucionária para a geração de RNAs para AAs e finalmente aborda-se como RNAs podem induzir autômatos através de exemplos de comportamentos observados.

Ao longo deste capítulo são apresentadas implementações concretas de exemplos simples que ilustram as idéias apresentadas neste e nos capítulos anteriores.

No epílogo é apresentado um resumo final do que foi discutido e são listados pontos em aberto para trabalhos futuros.

# Capítulo 2

## Agentes Autônomos

### 2.1 Introdução

Quando se começa a pesquisar e ler a bibliografia disponível sobre Agentes Autônomos, nos defrontamos com uma séria lacuna. Não se encontra uma definição clara e precisa sobre o que é um “agente” e, nem quando, ou porque, ele se torna “autônomo”.

Ao longo deste capítulo procura-se preencher pelo menos parte desta lacuna. Analisam-se os trabalhos relacionados com esta área e procura-se extrair destes, inicialmente, uma conceituação básica comum a vários autores. Em seguida, apresentam-se as diversas abordagens utilizadas por estes pesquisadores no desenvolvimento destes sistemas, tais como programação direta, técnicas de IA simbólica, “subsumption architecture”, etc., analisando características e limitações de cada abordagem. Por fim, procura-se construir uma fundamentação formal para o problema da conceituação de agentes autônomos, utilizando a abordagem de Teoria de Sistemas. O objetivo desta formalização é tornar mais claro os conceitos de Agentes Autônomos e de Ambiente, bem como as interações entre estes dois Sistemas e permitir uma base para a avaliação de desempenhos e comparação entre implementações.

Os trabalhos na área de Agentes Autônomos têm características multi-disciplinares e, dependendo da área de interesse de determinado autor, a nomenclatura encontrada e a abordagem utilizada pode variar imensamente. O significado preciso do termo “agente” é mal-definido, porém, hoje em dia encontram-se trabalhos envolvendo agentes em duas grandes áreas.

A primeira grande área de pesquisa envolvendo agentes parece estar mais relacionada com os “software agents”, um campo emergente de pesquisa que combina elementos da inteligência artificial tradicional, teoria de jogos, psicologia e programação orientada a objetos. Nesta abordagem os agentes são considerados entidades com-



putacionais baseadas na idéia de que os usuários necessitam apenas especificar um objetivo em alto-nível, ao invés de utilizar instruções explícitas, deixando as questões de como e quando agir, a cargo do agente. Aplicações utilizando esta abordagem procuram criar produtos na área de interfaces amigáveis, cartografia, auxílio ao ensino, ao diagnóstico médico, etc. Quanto a nomenclatura utilizada, vê-se que alguns autores também utilizam o termo “Agentes Inteligentes”, como se encontra na edição dedicada a este tema na revista “Communications of ACM” [4] ou ainda “softbots” [169]. Maiores informações sobre “software agents” podem ser encontradas em [4] [89] [116] [196]

A outra grande área de pesquisa envolvendo os agentes, e a que nos interessa mais especificamente, é a área da robótica, também conhecida como a dos “hardware agents” [89]. O desenvolvimento de pesquisas nesta área se deu a partir do final da década de 70 quando surgiram os primeiros robôs ditos de terceira geração, os chamados “Robôs Inteligentes” [54]. Segundo Alves, o grande diferencial destes robôs com relação aos robôs “Play-Back” de segunda geração é a capacidade de sensorar o ambiente no qual estão inseridos. O salto de qualidade para esta terceira geração de robôs estava na possibilidade do robô detectar mudanças ambientais e, através da reavaliação de seus objetivos, encontrar uma nova seqüência de ações capazes de persegui-los, sem que esta seqüência tivesse sido prevista [54]. Nesta área, além do termo “Agentes Autônomos”, vamos encontrar autores que utilizam o termo “Robôs Inteligentes” [54][12], “Sistemas Autônomos” [32][184], outros chamam de “Sistemas Veiculares Autônomos” (AGVs - Autonomous Guided Vehicles)[50] ou simplesmente “Veículos Autônomos”[31][109], outros ainda de “Organismo Artificial” [139], “Embedded Systems” [90] e “Situated Automata” [91] [38].

Da mesma maneira que a abordagem adotada, as definições e a nomenclatura variam entre os diversos autores, também os resultados apresentados utilizam métricas de avaliação bastante diversas, o que torna difícil a comparação entre os resultados de diferentes autores [90].

Ao procurar uma definição de Agente Autônomo comum a todos estes autores, chega-se a seguinte definição por senso comum:

“Um Agente Autônomo é um sistema computadorizado capaz de extrair informações do seu ambiente (eventualmente este ambiente é computacional) e, através de alguma capacidade cognitiva, mapear as informações extraídas em ações que, eventualmente, podem afetar o ambiente de modo a alcançar os objetivos para os quais foi projetado”.

Outra definição bastante clara aparece na chamada de trabalhos para a Primeira Conferência Internacional sobre Agentes Autônomos realizada na Califórnia

em fevereiro de 1997 [169].

“Agentes são sistemas computacionais que operam em ambientes dinâmicos e imprevisíveis. Eles interpretam dados obtidos pelos sensores que refletem eventos ocorridos no ambiente e executam comandos em motores que produzem efeitos no ambiente. O grau de “autonomia” de um agente está relacionado à capacidade de decidir por si só como relacionar os dados dos sensores com os comandos aos motores em seus esforços para atingir objetivos, satisfazer motivações, etc.”

Ao longo deste trabalho, sempre que se utilizar o termo Agentes Autônomos, estaremos nos referindo aos “hardware agents” que operam em um ambiente físico real, mesmo que muitas vezes em nossas experiências, utilizemos agentes e ambientes simulados em computador.

Todo AA possui algum mecanismo capaz de sentir aspectos sobre o estado do ambiente. A este mecanismo chamamos sensores. Da mesma maneira, um AA afeta o ambiente através de atuadores [90]. De modo geral, todo AA opera de uma maneira cíclica, examinando o estado do ambiente e ativando seus atuadores como especificado pelo mapeamento fornecido pela entidade cognitiva. A Figura 2.1 mostra de forma esquemática os elementos que compõem e o funcionamento de um AA.

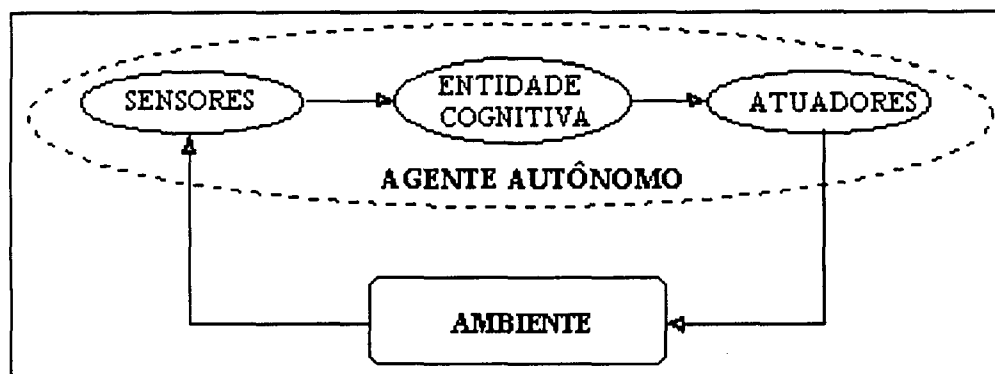


Figura 2.1: Funcionamento genérico de um Sistema Autônomo

A entidade cognitiva responsável pelo mapeamento coerente entre os dados obtidos pelos sensores em ações executadas pelos atuadores é a responsável pela autonomia do agente. Se o conceito de agente é simples de ser compreendido, o conceito de autonomia é bem mais vago. Ao se procurar um dicionário, vemos que o conceito de autonomia relaciona-se com a capacidade de governar-se por si mesmo [60]. Assim, um agente torna-se autônomo quando, quaisquer que sejam os dados obtidos do

ambiente através dos seus sensores, as ações daí decorrentes continuarão permitindo ao agente subsistir no ambiente e a continuar a sua operação. Por exemplo, um robô que se movimentasse por uma sala, batesse em uma parede e ficasse trancado, ou seja, não pudesse mais se movimentar, não é considerado um agente autônomo. Para que este robô pudesse ser considerado autônomo, o fato de bater na parede deveria desencadear um processo que primeiramente avaliasse as condições que o levaram a bater na parede de modo a que estas condições não se repetissem no futuro, e em seguida avaliasse estratégias para tirá-lo daquela condição de não operação.

É importante não confundir a noção de autonomia com a de imortalidade. Por autonomia deve se entender que todos os requisitos para a operação no ambiente estão no próprio agente, mas que eventualmente, em função das limitações nas capacidades sensoriais, motoras, e/ou cognitivas, ou mesmo de condições agressivas do ambiente, pode existir um momento em que o agente não seja mais capaz de operar no ambiente. Por isso o conceito de autonomia é relativo, não existe nenhum sistema que seja 100% autônomo. Uma maneira de medir o grau de autonomia de um agente poderia ser pelo tempo em que o agente consegue operar no ambiente mantendo um determinado desempenho de operação.

Assim, pode-se dizer que um requisito necessário para dotar um agente de autonomia é dotá-lo de inteligência. Esta inteligência deve tornar o agente capaz de se adaptar, ou seja, modificar o mapeamento entre sensações e ações de maneira a melhorar o seu desempenho de operação no ambiente [66].

A entidade cognitiva pode ser implementada de diversas maneiras, como pode-se ver em seguida. Neste trabalho, a entidade cognitiva é implementada na forma de redes neurais artificiais, por razões que serão expostas oportunamente.

Outro conceito bastante importante no estudo dos AAs é o conceito de comportamento. Podemos entender comportamento, neste caso, como sendo a resposta fornecida pelo AA para os estímulos recebidos do ambiente, ou seja, o mapeamento, efetuado pela entidade cognitiva, entre os dados recebidos pelos sensores e a saída dos atuadores [109]. Deve-se atentar para o fato de que este mapeamento pode possuir estados internos, fazendo com que a saída dos atuadores dependa potencialmente de todas as entradas passadas recebidas pelos sensores [90].

## 2.2 Alternativas de Implementação

O projeto e a implementação de AAs é tarefa extremamente difícil e complexa e sem a utilização de modelos conceituais claros e ferramentas de programação poderosas, as complexidades do mundo real podem facilmente tornar esta tarefa simplesmente

impossível de ser realizada. Como os AAs foram idealizados para operar em ambientes abertos e dinâmicos, técnicas tradicionais de controle baseadas em paradigmas matemáticos bem fundamentados, raramente podem ser utilizados [91]. Do mesmo modo, a questão da definição de uma arquitetura adequada para a implementação de AAs ainda é um problema em aberto. Pode-se dizer que pesquisas nesta área ainda estão em sua infância, e que resultados práticos somente poderão ser obtidos após numerosos sistemas terem sido projetados e testados [109].

### 2.2.1 Programação Direta

A abordagem primeiramente utilizada na construção de agentes autônomos foi a de simplesmente tentar programá-los. Segundo Kaelbling [90], controladores simples eram construídos utilizando técnicas de especificação lógica e de programação procedural.

Ao projetista do sistema eram fornecidos uma especificação do ambiente ou classe de ambientes em que o sistema deveria operar, bem como a especificação da tarefa que o sistema deveria realizar. Apesar de parecer teoricamente possível construir o sistema, uma vez que as especificações completas e corretas fossem fornecidas, estes pré-requisitos dificilmente eram satisfeitos, além disso, a medida em que os sistemas que se desejava construir se tornavam mais complexos, no caso de vários ambientes ou mesmo ambientes cujas características variam com o tempo, a tarefa de programação se tornava virtualmente impossível.

Um exemplo interessante dos problemas encontrados na utilização desta abordagem para implementação de AAs é a descrição de uma tentativa de implementação fornecida por Kaelbling [90]:

“Certa vez eu dispendi um grande esforço tentando programar um robô móvel para utilizar seus sensores ultra-sônicos de forma a se movimentar por um corredor. Eu possuía a especificação física do ambiente (era o corredor em que eu estava sentada) e especificações bastante acuradas dos fabricantes dos sensores e atuadores do robô. Teoricamente, eu possuía informações suficientes para escrever um programa de controle correto. Entretanto, as especificações sobre as capacidades do robô e sobre as propriedades do ambiente eram inviáveis de serem incorporadas diretamente em um programa de controle. Então, eu trabalhei seguindo o seguinte ciclo:

- Escrever um programa para o robô;
- Rodar o programa no robô e observá-lo bater na parede;
- Analisar o comportamento do robô e descobrir onde o programa falhou;

- Consertar o programa;
- Rodar o programa no robô e observá-lo bater na parede (desta vez por uma outra razão);

e assim sucessivamente. O resultado deste ciclo foi que eu aprendi bastante sobre a natureza das interações entre os sensores do robô e o ambiente físico. Utilizando estas informações, eu aprendi sobre o ambiente a adaptei o comportamento do robô, de modo a que ele realizasse corretamente a tarefa especificada. Uma estratégia muito mais eficiente teria sido para mim projetar um comportamento para o robô que permitisse a ele mesmo se adaptar ao ambiente em que estivesse.”

### 2.2.2 Inteligência Artificial Simbólica

Se no passado a Robótica e a Inteligência Artificial (IA) eram atividades independentes, com técnicas e paradigmas diferentes, nos últimos anos, diversos esforços têm sido desenvolvidos no sentido de integrar estas duas áreas.

A abordagem simbólica da IA defende a idéia de que a inteligência opera sobre um sistema de símbolos. A idéia implícita é que as interfaces sensoras e motoras são conjuntos de símbolos sobre os quais opera o sistema cognitivo central. Este sistema central, ou máquina de raciocínio opera sobre estes símbolos independentemente do domínio. Os símbolos, por sua vez representam entidades do mundo. Eles podem representar objetos, propriedades ou conceitos.

Esta idéia inicial dos pesquisadores de simplesmente anexar sensores e atuadores aos programas existentes de IA deu lugar a uma séria reavaliação dos algoritmos clássicos de IA à luz dos problemas envolvidos com o trabalho no mundo físico e em tempo real. Alguns problemas importantes da operação no mundo real são listados a seguir [150] [37] [39] [109]:

- A entrada para um programa de IA tem forma simbólica. A entrada para um robô é um sinal analógico, como uma imagem de vídeo bidimensional ou uma onda sonora da fala. Como então converter este sinal analógico em uma descrição simbólica correta? Certamente esta conversão deve ser independente da tarefa. No entanto, existem evidências psicofísicas de que certos aspectos da percepção, tais como a atenção, são ativos e dependentes da tarefa que se está realizando.
- Os robôs precisam de hardware especial para perceber e atuar no mundo, enquanto os programas de IA precisam apenas de computadores de propósito geral.

- Os sensores dos robôs são imprecisos, e seus atuadores têm precisão limitada. Há sempre algum grau de incerteza sobre onde exatamente o robô está localizado, como e onde estão os obstáculos e objetos em relação a ele.
- Muitos robôs precisam reagir em tempo real.
- O mundo real é imprevisível, dinâmico e incerto. Um robô não pode esperar manter uma descrição completa e correta do mundo. Isto significa que um robô deve considerar os prós e contras de criar e executar planos. Esse equilíbrio tem vários aspectos. Primeiro, um robô pode não possuir informações suficientes sobre o mundo para executar um planejamento útil. Nesse caso, ele precisa primeiro engajar-se na atividade de coletar informações. Além disso, uma vez iniciada a execução de um plano, o robô tem de monitorar continuamente os resultados de suas ações. Se os resultados forem inesperados, então ele poderá ter de efetuar um replanejamento.

Muitos pesquisadores viram nestes problemas evidências de que seria fantasioso esperar que tarefas orientadas a ação em tempo-real pudessem ser implementadas nestas máquinas utilizando técnicas clássicas de IA. Assim, já há alguns anos, estes pesquisadores estão desenvolvendo novas idéias de como os Sistemas Autônomos devam ser organizados [112].

### 2.2.3 Implementações baseadas em comportamento

Enquanto as alternativas de implementação anteriores baseadas em programação direta e IA Simbólica que utilizavam uma arquitetura de sistema puramente hierárquica, com módulos estanques de percepção, modelagem do ambiente, planejamento e execução, se mostravam ineficientes para a construção de AAs capazes de operar em tempo-real no mundo real, a partir da metade da década de 80, uma série de novas propostas foram apresentadas [34] [141] [5] [65] [70].

Um dos pesquisadores pioneiros nesta nova idéia foi Rodney Brooks [34]. Esta nova proposta veio trazer uma nova abordagem para o desenvolvimento da Inteligência Artificial. Ao invés de se basear unicamente na representação simbólica do conhecimento como base para a construção de sistemas autônomos capazes de operar no mundo real, as novas propostas buscavam a representação diretamente do mundo físico para o desenvolvimento dos chamados sistemas reativos ou comportamentais (“behavior-based”). Na sua forma mais pura, não há nenhum objetivo, plano ou modelo de mundo explícito nesses sistemas. Eles simplesmente reagem à situação que tem em mãos [150] [35] [37] [38] [141].

Ao utilizar esta nova abordagem para a construção de robôs autônomos capazes de operar em ambientes desestruturados e que podem mudar dinamicamente, Brooks utilizou uma arquitetura a que chamou de “Subsumption Architecture” [34] [37] [36] também chamada por Alves de Arquitetura de Assunção [54].

Uma excelente descrição desta arquitetura pode ser encontrada em [54] e será reproduzida a seguir:

“A abordagem baseada em comportamento conseguiu, de fato, fazer avançar a pesquisa envolvendo o paradigma de robôs de terceira geração, ou robôs inteligentes. Um controlador baseado em comportamento é dividido em camadas. Cada camada é responsável por um tipo de comportamento ou tarefa a ser executada pelo sistema global. Nesta estrutura paralela, cada camada possui um caminho completo dos sensores aos atuadores; a arquitetura de controle permite a produção de comportamento útil, antes de todo o conjunto de camadas, inicialmente projetado, estar completo; a estrutura em paralelo reduz a tendência ao congestionamento das informações; a falha de qualquer das camadas não significa, necessariamente, o colapso total do sistema. Cada camada usa apenas as informações do sensor que lhe é acoplado e a percepção de mundo que necessita para sua tarefa específica, isto é, cada camada é completa (no sentido de implementar um comportamento ou tarefa). Cada módulo, então, precisa resolver somente o problema que lhe é pertinente. Brooks construiu um robô móvel autônomo utilizando a arquitetura de “assunção” cuja primeira camada, a camada 0, evitava obstáculos. Em seguida, adicionou a camada 1 que introduz a atividade de fazer com que o robô se dirija a um determinado lugar. Independentemente, a primeira camada livra o robô dos obstáculos que se lhe interponham. A segunda monitora o progresso do robô e envia comandos atualizados aos atuadores na perseguição de seu objetivo, sem estar consciente, explicitamente, dos obstáculos que foram evitados pela camada inferior.

Cada camada é composta por uma rede de topologia fixa de máquinas de estado finito simples que são combinadas através de mecanismos chamados inibidor e supressor. Cada máquina de estado finito possui uma quantidade de estados, um ou dois registradores internos, um ou dois relógios internos e acesso às máquinas simples que podem fazer cálculos tais como soma de vetores. São ativadas através de mensagens que recebem, e uma mudança de estado ocorre quando chega uma determinada mensagem ou quando o tempo estipulado para este estado expira. Não há memória global compartilhada. As entradas de cada máquina de estado finito podem ser suprimidas e as saídas podem ser inibidas por outras máquinas” [54].

As Figuras 2.2 e 2.3, extraídas de [54] ilustram esta estrutura de camadas de comportamento.

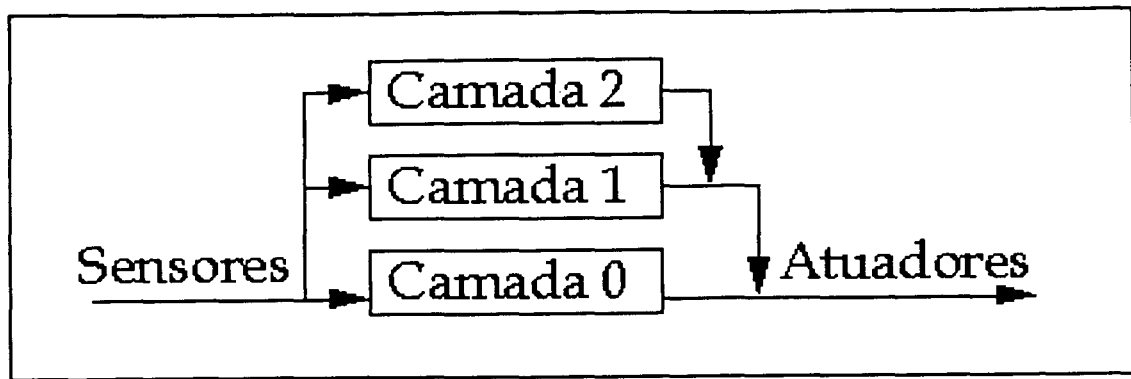


Figura 2.2: Arquitetura em camadas de comportamentos da “subsumption architecture”

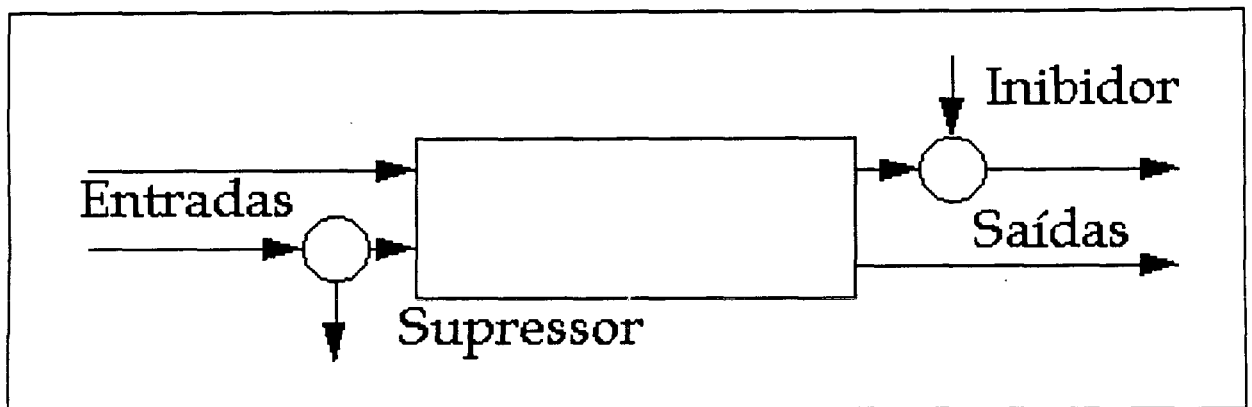


Figura 2.3: Esquema Inibidor/Supressor para uma camada de comportamento

A “subsumption architecture” faz emergir comportamentos através da interação assíncrona entre módulos de ação simples e o ambiente [90].

A modelagem simbólica do mundo real é uma tarefa difícil que a “subsumption architecture” evita. Por outro lado, não está claro se esta arquitetura alcança problemas que exijam planejamento complexo, além disso, uma série de outras limitações e dificuldades ainda persistem:

- As máquinas de estados finitas expandidas com elementos de temporização utilizadas por Brooks para implementar cada módulo de comportamento necessitam ser codificadas “manualmente” pelo programador [38];



- A interdependência entre os comportamentos implementados torna muito difícil a reconfiguração do sistema para que este se adapte a outras tarefas [77];
- A abordagem dificilmente pode ser expandida para domínios de comportamentos mais elaborados, tais como a destreza na manipulação de objetos [77];
- Na “subsumption architecture” os níveis de prioridades das atividades são determinados pela própria estrutura da arquitetura e não pode ser facilmente modificado [109];

Apesar das limitações apresentadas, os sistemas comportamentais têm apresentado resultados práticos altamente promissores. Um exemplo disto foi o fato de um robô implementado com uma arquitetura deste tipo, chamado “Autonomous Flying Vehicle-I” (AFV-I), desenvolvido por pesquisadores do Laboratório de Pesquisas Robóticas da Southern California University, ter vencido a 4ª Competição Internacional de Robôs Aéreos em 1994.

Esta competição, promovida anualmente pela Associação para Sistemas Veiculares Não-Tripulados (“Association for Unmanned Vehicle Systems”), requer robôs voadores para localizar e manipular objetos, transportando-os de um lugar para outro. Os robôs devem realizar estas tarefas sob condições adversas, sem interferência humana e dentro de um período de tempo limitado.

Conforme descrito pela equipe que desenvolveu o AFV-I, ele utilizava uma arquitetura de controle baseada em comportamentos, que particionava o problema de controle em um conjunto de módulos fracamente acoplados. Cada módulo, ou comportamento, era responsável pela execução de uma tarefa específica. Os módulos eram organizados hierarquicamente, com os de baixo-nível responsáveis por comportamentos reflexivos tais como o desvio de obstáculos e a manutenção de altura, e os comportamentos de alto-nível responsáveis por tarefas como navegação e localização de objetos [130].

#### 2.2.4 Outras alternativas

A utilização de máquinas de estados finitas expandidas com elementos de temporização foi a alternativa encontrada por Brooks para implementar cada módulo de comportamento, no entanto, inúmeras outras alternativas e técnicas têm sido empregadas para o desenvolvimento de sistemas comportamentais, tais como redes neurais, algoritmos evolucionários, reinforcement-learning e sistemas nebulosos, seja de maneira isolada seja combinadas entre si [90] [53] [63] [130] [141] [153] [38].

## Sistemas Nebulosos

A própria equipe que desenvolveu o AFV-I, já descrito anteriormente, parece ver na utilização de sistemas nebulosos uma ferramenta para superar as limitações existentes na Arquitetura Baseada em Comportamentos.

*“Increasing the complexity of a behavior-based system is difficult due to the possible coupling of behaviors. Therefore, we hope to develop methods to overcome this difficulty. We believe that integration of a fuzzy-rule system with the behavior-based controller can maintain the strengths while reducing or eliminating the weakness of the behavior-based approach.*

*Fuzzy systems have demonstrated the ability to deal with uncertainty in unstructured, real-world environments for a variety of applications, including autonomous robots. For a system as complex as an autonomous flying helicopter, determining and organizing the fuzzy rules is not trivial. Automatically generating these rules is a desirable approach. A fuzzy-logic-augmented controller can automatically tune the rules through techniques such as reinforcement learning to optimize system performance” [130].*

Outros pesquisadores trabalhando com “fuzzy” para controle de veículos ou robôs autônomos são Kramer [102], Hessburg & Tomizuka [85], Kosko & Kong [99], Sugeno [177], Fabro & Gomide [63] entre outros.

## Algoritmos Evolucionários

O próprio Brooks parece estar, hoje em dia, bastante interessado na utilização de algoritmos evolucionários, particularmente da Programação Genética, para superar algumas das limitações apresentadas pela “subsumption architecture” em desenvolver AAs baseados em comportamento [38].

Ainda segundo Brooks, antes disso, vários pesquisadores já haviam utilizado Algoritmos Genéticos para programar agentes [38]. Langton [103] também sugeriu a utilização de Programação Genética para implementação de AAs baseados em comportamento, e Koza [100] [101] tem apresentado vários trabalhos em que utiliza a Programação Genética em robôs autônomos que utilizam a “subsumption architecture”.

## Aprendizado por Reforço (Reinforcement Learning)

O problema do aprendizado do mapeamento entre os dados obtidos pelos sensores e a ação efetivada pelos atuadores quando feita por tentativa e erro, é também chamada de Aprendizado por Reforço ou “Reinforcement Learning” devido a sua

semelhança com os mecanismos de aprendizado de comportamentos encontrados em seres humanos e animais. Este método também é chamado de “aprendizado com um crítico”, em contraste com os métodos de “aprendizado com um professor” normalmente encontrados em técnicas de aprendizado supervisionado [90].

Encontramos muitos pesquisadores utilizando esta abordagem, tais como Kaelbling [90], Piggott & Sattar [141], Mahadevan & Connell [115] e Watkins [189].

No método de aprendizado por reforço, o sistema aprende as reações adequadas diretamente através da realimentação de um sinal que indica se a ação desenvolvida foi adequada ou não. No caso mais simples, o sinal seria negativo até que robô alcançasse o objetivo proposto (por exemplo, passar por uma porta). No entanto, devido ao longo período de treinamento, a utilização de aprendizado por reforço em AAs ainda é limitada. Além disso, a exploração aleatória do ambiente, envolvida no processo de aprendizado pode levar a condições de operação inseguras, que muitas vezes inviabilizam a utilização deste método [92].

## Redes Neurais

Alguns trabalhos bastante interessante tem sido encontrados utilizando Redes Neurais para a implementação de AAs, seja em mecanismos reais seja em ambientes simulados [184].

Nesta área, podemos encontrar os trabalhos de Vaario [184] onde Sistemas de Lindenmayer são utilizados para gerar Redes Neurais capazes de fazer emergir comportamentos inteligentes em agentes simulados.

Taga et al. [178] desenvolveu um modelo de Rede Neural Dinâmica para controlar a locomoção de um bípede em um ambiente desconhecido. McKenna & Zeltzer [122] desenvolveram no MIT trabalhos nesta mesma direção.

Beer et al. [28] também desenvolveram um modelo de Rede Neural, com o qual equiparam um inseto artificial. A criatura era capaz de achar alimentos em um ambiente artificial, evitar obstáculos e seguir paredes.

### 2.2.5 Arquiteturas de Quadro-Negro e Multi-Agentes

Parece que nem a Arquitetura Hierárquica com seus paradigmas simbólicos, nem a Arquitetura Baseada em Comportamento que, na sua forma mais pura, evita qualquer forma de modelagem e planejamento, são capazes de resolver de maneira isolada o problema de implementação e operação de AAs capazes de executar tarefas complexas. Outras arquiteturas, tais como a de Quadro-Negro (“Blackboard”) e de Multi-agentes têm sido propostas para serem utilizadas em domínios onde os proble-

mas são tipicamente complexos e exigem diversas fontes e técnicas de conhecimento para serem resolvidos. Cada agente especialista que compõe o sistema pode, então, ser implementado usando o paradigma mais adequado.

Sistemas de Quadro-negro utilizam módulos especialistas distribuídos que interagem através de dados compartilhados (acessados através do Quadro-negro). Cada um dos agentes possui seu próprio mecanismo de inferência e conhecimento local para executar uma tarefa específica. Os sistemas de Quadro-negro possibilitam um projeto altamente paralelizado, permitindo a depuração individual de cada módulo especialista [109] [164]. Segundo Scalabrin [164], como na arquitetura de Quadro-negro a comunicação entre os especialistas se dá apenas através dos dados compartilhados, existe a possibilidade de ocorrerem “gargalos”. Este fato também é citado por Liscano et al. [109], que afirma que vários projetos de robôs móveis que utilizam a arquitetura de Quadro-negro, modificaram suas estruturas a fim de poder suportar os requisitos de tempo-real peculiares a este tipo de aplicação.

Informações mais completas e precisas sobre a arquitetura de Quadro-negro podem ser obtidas em [136].

O JANUS, um robô manipulador de objetos, projeto desenvolvido no GMD na Alemanha e coordenado por Śmieja & Beyer [171] [29] é um exemplo da utilização da arquitetura de Quadro-Negro, como pode ser visto na Figura 2.4. Outro projeto utilizando esta mesma arquitetura é descrito por Liscano et al. em [109].

Já na arquitetura de Multi-agentes, os agentes podem ser mais simples ou complexos e se comunicarem diretamente entre si ou através de um “facilitador”, utilizando mecanismos de trocas de mensagens. Assim como no caso da arquitetura de Quadro-negro, um estudo aprofundado sobre conceitos e aplicações de Multi-agentes pode ser obtido em [145].

## 2.3 A Modelagem de Agentes Autônomos

Conforme visto anteriormente, as várias abordagens e alternativas de implementação de agentes autônomos têm refletido a área de interesse particular de cada pesquisador, assim, encontra-se na literatura diferentes nomenclaturas, definições e métricas utilizadas para os mesmos conceitos básicos.

Alguns pesquisadores também têm se preocupado em estabelecer uma base formal para a descrição dos conceitos envolvidos na área de AAs. Kaelbling [90] procura modelar a interação entre um agente e o ambiente no qual ele opera, no entanto, esta

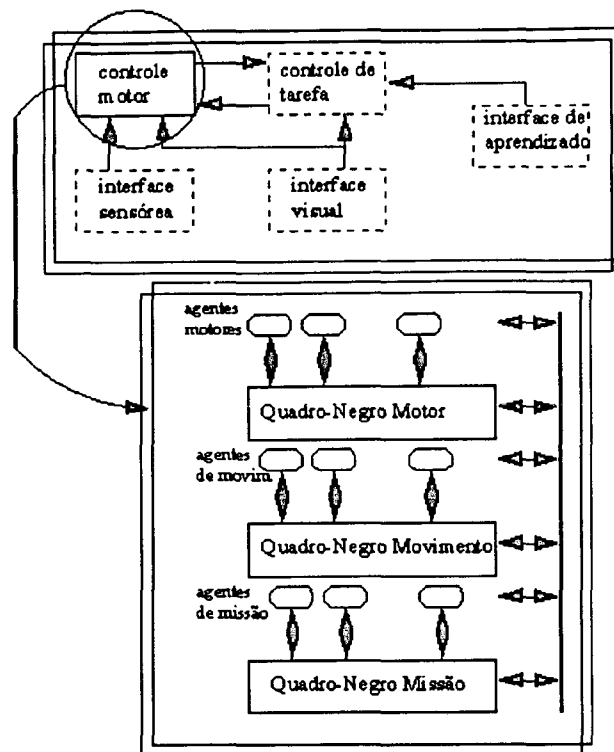


Figura 2.4: Módulo de controle motor implementado utilizando a arquitetura de quadro-negro

sua modelagem está intimamente relacionada com a utilização da técnica de Aprendizado por Reforço para implementação do AA. Para isto, ela modela o ambiente como um autômato de estados finitos determinístico ou probabilístico. Smithers [173] utiliza como base o Materialismo Eliminativo proposto por Churchland [47] para fazer uma modelagem livre de qualquer conotação ligada aos conceitos da Inteligência Artificial Simbólica e que ele chamou de “psicologia folclórica”. Smithers caracteriza um agente como sendo um sistema coerente de processos organizados em uma arquitetura e que de maneira robusta e confiável efetua mudanças específicas em seu ambiente enquanto recebe alguns estímulos do ambiente que podem ser: consequência (direta ou indireta) das suas ações; ou eventos não-relacionados que estão ocorrendo no ambiente. Da mesma maneira, o ambiente é caracterizado como um sistema de processos (não necessariamente coerente, nem organizado em uma arquitetura) que é afetado e/ou afeta um agente.

Em função destas diferentes modelagens, para Kaelbling “o comportamento faz parte da definição do agente, sendo o elemento que mapeia uma seqüência de sinais do ambiente sentidos pelo agente em ações realizadas pelo agente e que afeta o ambiente”. Já para Smithers, “o comportamento não é uma propriedade apenas

do agente, mas é caracterizado pelas dinâmicas do espaço de interação entre os processos do agente e os processos do ambiente”.

Além de Kaelbling e Smithers, outros pesquisadores, entre eles Beer [27], Kiss [97] e Steels [176] perceberam que freqüentemente os agentes apresentam um comportamento dinâmico e começaram a abordar este problema sob a perspectiva de sistemas dinâmicos [114]. Esta abordagem é bastante interessante e promissora uma vez que permite a utilização de um conjunto de definições de propósitos gerais. Do mesmo modo, a implementação de um AA pode ser alcançada utilizando-se diferentes ferramentas, contanto que estas ferramentas também possam ser descritas utilizando os mesmos conceitos de Teoria de Sistemas [159].

Seguindo este raciocínio, pretende-se neste capítulo utilizar a abordagem de Teoria de Sistemas para construir uma base formal para a conceituação de Agentes Autônomos e outros conceitos relacionados. Com esta base teórica mostra-se como uma ferramenta formal bem conhecida, como é o caso dos Autômatos de Estados Finitos podem ser utilizados para descrever um AA. Propõe-se, então, que utilizando os mesmos princípios teóricos para as Redes Neurais Artificiais, estas podem ser utilizadas para modelar os Autômatos e, por conseguinte, os Agentes Autônomos.

### 2.3.1 Conceitos de Sistemas

O texto desta seção foi extraído em grande parte de [19] e de [159] e procura fornecer um embasamento ao leitor sobre a ferramenta que será utilizada para a construção de uma fundamentação formal para os conceitos relacionados aos Agentes Autônomos. O estilo da apresentação é principalmente matemático, baseado na teoria de conjuntos, para permitir suficiente generalidade no uso dos conceitos apresentados<sup>1</sup>.

A Teoria de Sistemas foi criada com a finalidade de ser independente do domínio estudado, tal como uma teoria matemática. Assim, a seguir a teoria de sistemas é usada como apoio aos principais conceitos de Agentes, o que permite definir graus de autonomia, comportamento de um AA, aprendizado, desempenho de um AA e outros conceitos, baseados em suas propriedades funcionais em lugar de dar ênfase a detalhes ligados a um campo de aplicação particular.

A noção de sistema se desenvolveu durante os últimos anos, tendo suas raízes associadas à cibernética [193] introduzida por Norbert Wiener nos anos quarenta. Sendo a cibernética essencialmente uma ponte entre diferentes disciplinas, é natural que o conceito de sistema tenha suas raízes na biologia, na engenharia, na matemática, em modelos populacionais, na economia e, mais recentemente, em teoria

---

<sup>1</sup>O conjunto de definições apresentado a seguir foi anteriormente utilizado para conceituar modelos qualitativos e sua representação de conhecimento por Barreto & Deneyer [20].

da computação. As definições que se seguem têm esta característica de multidisciplinaridade e são inspirados principalmente em Bertalanfy [186] (biologia), Zadeh & Desoer [197] (engenharia), Kalman, Arbib & Falb [93] (autômatos finitos e otimização), Arbib & Manes (computação) [11] e Mesarovic & Takahara, [123] (modelos populacionais). Um primeiro esforço para obter estes conceitos foi feito no contexto de formalizar o problema de otimização [16], [17], e no estudo de representação de conhecimentos sobre o mundo físico [22], [21], [23], [20]. Na teoria de sistemas e, em particular, na teoria de sistemas aplicada à modelagem e formalização de Agentes Autônomos, alguns conceitos são de fundamental importância. São eles:

- o que constitui um sistema;
- orientação do sistema, ou seja, quais as entradas e quais as respostas do sistema;
- os diversos níveis de descrição de um sistema;
- reticulação;
- topologia no caso de um sistema formado por vários outros, ou seja, um sistema complexo;
- adaptabilidade que é alcançada pelo aprendizado e evolução.

Um primeiro passo na aplicação dos conceitos de sistema a um problema específico é definir o que constitui o sistema. Um segundo ponto a ser considerado é a escolha do nível de descrição desejado o que leva a uma hierarquia de conceitos que será apresentada a seguir.

### Nível Comportamental

**Sistema Geral** Desta forma a primeira definição concerne o que pode ser considerado como o menor nível de descrição possível em que somente as possíveis interações com o mundo exterior são retidas. Definiu-se o que constitui o sistema e o que constitui o mundo exterior. Este conceito é chamado de *sistema geral*:

**Definição 2.3.1** *Um sistema geral  $\Sigma_g$  é definido por um conjunto de relações entre as entidades relevantes que caracterizam as interações com o mundo exterior. Então,  $\Sigma_g \in I$  onde  $I$  é o conjunto de todas as interações.* □

Se bem que bastante geral, existem sistemas descritos neste nível.

**Exemplo 2.3.1**

Uma tabela referente a dados de fabricação de peças, com duas colunas, uma contendo os nomes das peças e outra coluna contendo uma máquina utilizada no seu processo de fabricação. Tanto pode haver mais de uma peça fabricada pela mesma máquina como uma peça pode necessitar passar por várias máquinas para ser fabricada. Como não foi especificado como se consulta a tabela, se para saber que por que máquinas uma peça deve passar para ser fabricada ou que peças uma determinada máquina produz, o sistema é não orientado.

**Sistema Orientado** Caso se deseje definir qual das duas variáveis será usada como excitação (entrada) e a outra como resposta (saída)<sup>2</sup> tem-se um *sistema orientado*.

**Definição 2.3.2** *Um sistema orientado pode ser caracterizado por uma relação entre o conjunto de entradas e saídas:*

$$\Sigma_o \subset \Omega \times \Gamma$$

onde:

$\Sigma_o$  : o sistema;

$\Omega$  : o conjunto de entradas admissíveis;

$\Gamma$  : o conjunto de saídas admissíveis.

■

**Exemplo 2.3.2**

Utilizando-se o exemplo anterior, se especificarmos que desejamos saber por que máquinas uma peça deve passar para ser fabricada passamos a ter um sistema orientado.

**Exemplo 2.3.3**

Um programa pode ser considerado como um sistema orientado, os dados sendo a entrada e as respostas as saídas.

---

<sup>2</sup>Algumas vezes faz-se distinção entre os termos *excitação* e *entrada* e também entre *resposta* e *saída* de um sistema, *excitação* e *resposta* se referindo aos sinais e *entrada* e *saída* aos dispositivos do sistema onde age a *excitação* e aparece a *resposta*. Neste trabalho não é feita distinção entre estas palavras.



**Sistema Temporal** Os dois conceitos apresentados, sistemas geral e orientado não incluem o tempo como intrínseco ao sistema. O sistema é representado por uma relação matemática. Entretanto, freqüentemente, quando lidando com sistemas que são abstrações do mundo real, o tempo é um parâmetro relevante. Incluindo o tempo em uma definição de sistema, define-se :

**Definição 2.3.3** *Um sistema orientado onde  $\Omega$  e  $\Gamma$  são funções do tipo:  $\Omega : T \rightarrow U$  e  $\Gamma : T \rightarrow Y$  onde:*

*$U$  é o conjunto dos valores da entrada;*

*$Y$  é o conjunto dos valores da saída;*

*$T$  é um conjunto ordenado com um primeiro elemento, comumente denotado por  $t_0$  (às vezes é usual considerar  $t_0$  como  $-\infty$ ) que é chamado de conjunto de tempo  $T$ .*

*é um sistema temporal representado por  $\Sigma_t$*  ■

**Nota 1:** Utilizando a notação  $A^T$  para indicar o conjunto de funções com domínio  $T$  e conjunto imagem  $A$ , um sistema temporal é representado pela relação:

$$\Sigma_t \subset Y^T \times U^T$$

**Definição 2.3.4** *Um sistema temporal cujas funções  $\Omega$  e  $\Gamma$  são funções constantes (a imagem é sempre constituída pelos mesmos elementos de  $U$  e  $Y$ , respectivamente) é dito ser um sistema estático".* ■

#### Exemplo 2.3.4

Considere-se novamente o exemplo da tabela de fabricação de peças. Agora nós desejamos saber não só por que máquinas uma peça passa para ser produzida, mas também desejamos conhecer a ordem destas máquinas no processo de fabricação da peça. Neste caso, a ordem das entradas e saídas têm um significado e tem-se um sistema temporal. Este é um exemplo de como o mesmo sistema real pode ser descrito por diferentes níveis de descrição. Dependendo do objetivo da simulação, deve ser escolhido um ou outro nível.

**Definição 2.3.5** *Um sistema causal é um sistema temporal  $\Sigma_c \subset \Gamma^T \times \Omega^T$  tal que:*

$$\forall t \in T \text{ se } \omega_1(t_0, t] = \omega_2(t_0, t] \text{ então } y_1(t_i) = y_2(t_i)$$

onde:

- $t_0$  é o primeiro elemento do conjunto tempo;
- $\omega_1(t_0, t]$  é a função entrada entre os instantes  $t_0$  e  $t$ ;
- $\omega_2(t_0, t]$  é outra função entrada entre os mesmos instantes;
- $y_1(t_i), y_2(t_i)$  são valores das funções saída  $\gamma_1$  e  $\gamma_2$  correspondentes respectivamente a  $\omega_1$  e  $\omega_2$ ;
- $t_i$  é um valor de tempo no intervalo  $(t_0, t]$ .

□

**Definição 2.3.6** Se  $T$  é um intervalo do conjunto dos reais diz-se que o sistema é um sistema contínuo no tempo. □

**Definição 2.3.7** Se  $T$  é um subconjunto dos inteiros diz-se que o sistema é um sistema discreto no tempo. □

Até agora o sistema foi descrito em relação ao mundo exterior. Diz-se que o sistema é descrito no *nível comportamental*. O nível comportamental é importante porque é neste nível de descrição que se colhe informações por meio de experiências. Entretanto, esta descrição não diz nada sobre os mecanismos necessários para produzir este comportamento e é possível que dois mecanismos inteiramente diferentes produzam o mesmo comportamento. Estes mecanismos podem ser associados com um novo conjunto  $X$ , tal que, dado um elemento do conjunto de entrada  $\Omega$  e um elemento deste novo conjunto de parâmetros  $X$ , é possível definir univocamente o elemento do conjunto de saída. A este conjunto  $X$  denomina-se de *conjunto de estados*. Isto leva à definição de um sistema através de uma função, ou seja, um *sistema funcional*. Esta descrição é dita estar no *nível de estado*.

## Nível de Estado

**Sistema Funcional: Conceito de Estado** O conceito de estado é freqüentemente usado em várias disciplinas, por exemplo, na teoria de autômatos finitos. Aqui este conceito será apresentado de forma mais geral, aplicável ao conceito de sistema como uma relação matemática. A noção de sistema no nível comportamental freqüentemente não descreve o sistema em nível suficiente de detalhe, e isto por ser uma relação podendo haver mais de uma resposta a uma mesma excitação. Freqüentemente, deseja-se ter uma única saída para uma única entrada. Em uma relação pode haver muitos valores de saída correspondentes a um valor de entrada. Isto leva a ter o sistema sob a forma de uma função, ou seja, ao conceito de *sistema funcional*.

**Definição 2.3.8** Um sistema funcional  $\Sigma_f$  é caracterizado por:

$$\Sigma_f : \Omega \times X \rightarrow \Gamma$$

onde:

$\Omega$ : conjunto das entradas admissíveis;

$\Gamma$ : conjunto das saídas admissíveis;

$X$ : conjunto dos estados.

■

### Exemplo 2.3.5

A Figura 2.5 mostra um exemplo de um sistema funcional com 5 elementos no conjunto das entradas  $\Omega$ , 7 no conjunto das saídas  $\Gamma$ , e um conjunto dos estados  $X$  constituído pelo conjunto  $\{\alpha, \beta\}$ .

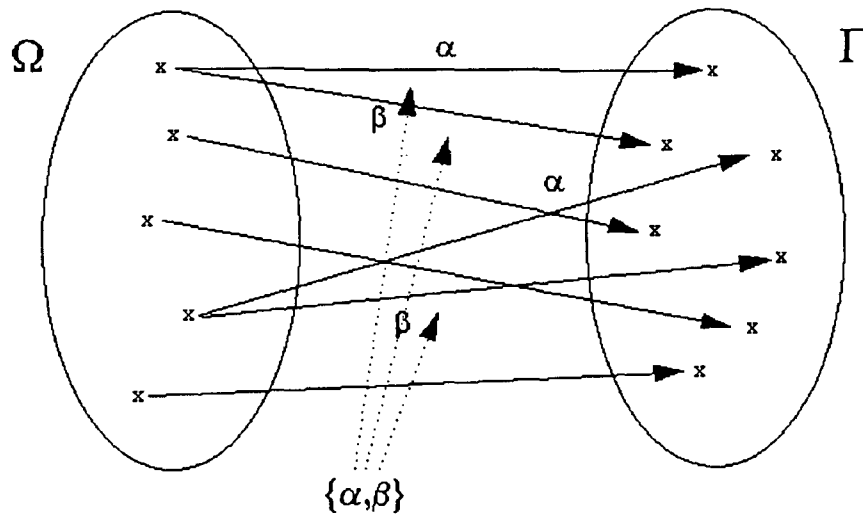


Figura 2.5: Exemplo de sistema funcional

Pode-se ver que para algumas entradas qualquer valor do conjunto  $X$  fornece a mesma saída; para outras deve ser especificado o elemento de  $X$ .

**Sistema Dinâmico** Um sistema pode também ser funcional e temporal. Neste caso o estado varia com o tempo. Este sistema se chama de *sistema dinâmico*. Em um sistema dinâmico, descreve-se um sistema como se estivesse descrevendo o mecanismo de como ele trabalha (internamente), especificando como o conjunto dos estados varia com o tempo. Tal descrição é suficiente para gerar uma descrição comportamental. Para isto basta fazer o “mecanismo” funcionar para cada entrada desejada gerando o sistema orientado correspondente. Formalmente tem-se:

**Definição 2.3.9** *Um sistema dinâmico é o objeto matemático:*

$$S = \{T, U, \Omega, Y, \Gamma, X, \Phi, \eta\}$$

onde:

$T$  é o conjunto dos tempos;

$\Omega$  é o conjunto das funções de entrada  $\omega \in \Omega = \{\omega : T \rightarrow U\}$ ;

$U$  é o conjunto dos valores da entrada;

$Y$  é o conjunto dos valores da saída;

$\Gamma$  é o conjunto das funções de saída  $\gamma \in \Gamma = \{\gamma : T \rightarrow Y\}$ ;

$X$  é o conjunto dos estados;

$\Phi$  é a função de transição dos estados:  $\Phi : T \times T \times X \times \Omega \rightarrow X$ ;

$\eta$  é a função de saída:  $\eta : T \times X \times U \rightarrow Y$ .

satisfazendo algumas condições de compatibilidade [93]. ■

### Exemplo 2.3.6

A máquina seqüencial é um sistema dinâmico. De fato, suponha o sistema definido pelas equações de transição de estado:

$$x(k+1) = F(x(k), u(k), k) \quad (2.1)$$

$$y(k) = h(x(k), u(k), k) \quad (2.2)$$

nós temos:

$k \in T = Z^+ = \{0, 1, 2, 3, 4, \dots\}$   $x(k) \in X$  conjunto de estados da máquina seqüencial, conjunto finito

$u(k) \in U$  conjunto de entradas da máquina seqüencial;

$y(k) \in Y$  conjunto de saída da máquina seqüencial;

portanto:

$(u(0), u(1), u(2), u(3), \dots) \in \Omega$  é uma seqüência de entrada da máquina seqüencial, e

$(y(0), y(1), y(2), y(3), \dots) \in \Gamma$  uma seqüência de saída.

$F$  é uma função de transição de estado particular  $\Phi$  que fornece o estado para o tempo sucessor do tempo presente

$h$  é a função de saída denotada por  $\eta$  na definição 2.3.9.

Havendo identificado os elementos da definição de um sistema dinâmico com as das equações do sistema seqüencial, verifica-se que, de fato, a máquina seqüencial é um exemplo de sistema dinâmico. ■

**Nota 2:** Informalmente, a noção de sistema dinâmico corresponde a um sistema funcional temporal cujo estado varia com o tempo dependendo do valor da entrada.

**Nota 3:** Todos os sistemas dinâmicos são sistemas temporais e funcionais.

**Nota 4:** Outro exemplo de sistema dinâmico com o qual os engenheiros estão familiarizados é o caso de um sistema dinâmico definido por um sistema de equações diferenciais, onde a função de transição de estado  $F$  é a solução do sistema.

### Tipos Particulares de Sistemas Dinâmicos

Escolhas particulares dos conjuntos envolvidos na definição de sistema dinâmico conduzem a diferentes tipos de sistemas. Então, tem-se:

**Definição 2.3.10** *Um sistema dinâmico temporal contínuo (ou sistema dinâmico contínuo no tempo) é um sistema dinâmico onde  $T$  é um subconjunto completo e compacto dos números reais* ■

**Definição 2.3.11** *Um sistema dinâmico temporal discreto (ou sistema dinâmico discreto no tempo) é um sistema dinâmico cujo conjunto dos tempos é um subconjunto dos inteiros.* ■

**Definição 2.3.12** *Um sistema invariante no tempo ou sistema estacionário é um sistema dinâmico cuja função de transição  $\Phi$  depende de um único elemento de  $T$  e a função de saída é independente de  $T$ . No caso contrário o sistema é dito variante no tempo ou não estacionário.* ■

Um tipo muito usual de sistema dinâmico em ciência da computação é o *autômato* ou *máquina de estados finitos*. Informalmente, uma *máquina de estados finitos* é um sistema dinâmico onde o conjunto dos tempos é o conjunto dos inteiros e a entrada, a saída e os estados são conjuntos finitos. Usualmente, neste caso, nós nos referimos aos valores possíveis da entrada e da saída como alfabetos de entrada e saída. Sob estas simplificações, não é essencial indicar explicitamente o conjunto dos tempos  $T$  nem explicitamente introduzir  $\Omega$  e  $\Gamma$ . Por outro lado, é usual declarar explicitamente um estado inicial correspondente ao tempo zero.

Formalmente, um autômato pode ser definido por [93]:

**Definição 2.3.13** *Um autômato (ou máquina) é descrito abstratamente como uma sêxtupla:*

$$S = \{U, Y, X, x_0, \lambda, \eta\}$$

onde:

*$U$  é um conjunto finito de entradas;*

*$Y$  é um conjunto finito de saídas;*

*$X$  é um conjunto de estados ou espaço de estado;*

*$x_0 \in X$  é o estado inicial;*

*$\lambda : U \times X \rightarrow X$  é a função de transição de estado,*

*$\eta : U \times X \rightarrow Y$  é a função saída.*

□

**Comentários:** Note-se que o tempo não aparece explicitamente na definição de autômato apresentada. Da mesma forma, tradicionalmente, não se coloca na definição a diferença entre função e valor admissível da função. Isto é uma questão de costume, podendo-se considerar que o autômato nada mais é do que um sistema dinâmico particular. Além disto, como é bem conhecido dos profissionais em simulação, no estudo de sistemas dinâmicos eles são aproximados, com a precisão que se deseje, por um autômato, o qual é implementado em computador.

**Definição 2.3.14** *Um autômato é dito finito quando o conjunto de estados é um conjunto finito.*

□

Um autômato é um sistema dinâmico invariante e discreto no tempo. Quando o espaço de estado é um conjunto finito, o autômato é chamado de autômato finito. Então, interpreta-se a sêxtupla formal acima como sendo uma descrição matemática de uma máquina à qual, se no tempo  $t_0$ , estiver no estado  $x_0$  e receber um segmento de entrada  $u$  do tempo  $t_0$  ao tempo  $t$  estará no tempo  $t$  no estado  $\lambda(x, u)$  e emitirá a saída  $\eta(x, u)$ .

**Exemplo 2.3.7**

$$G = \{U, Y, X, x_0, \lambda, \eta\}$$

onde:

$U = \{\star, \bullet\}$ ,  $Y = \{\clubsuit, \#\}$ ,  $X = \{A, B\}$  e as funções  $\lambda$  e  $\eta$  descritas a seguir. O estado inicial  $x_0$  pode ser tanto A como B.

$$\lambda : U \times X \rightarrow X$$

$$\eta : U \times X \rightarrow Y$$

os valores destas funções são dados por :

$$\lambda(\star, A) = B \quad ; \quad \eta(\star, A) = \clubsuit$$

$$\lambda(\star, B) = B \quad ; \quad \eta(\star, B) = \#$$

$$\lambda(\bullet, A) = A \quad ; \quad \eta(\bullet, A) = \#$$

$$\lambda(\bullet, B) = A \quad ; \quad \eta(\bullet, B) = \clubsuit$$

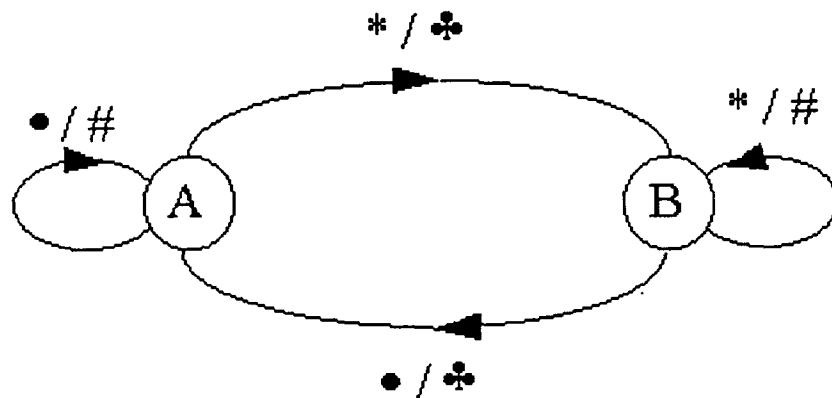


Figura 2.6: Grafo representando um autômato com dois estados

Um autômato finito é geralmente representado por um grafo orientado. Para cada estado, é associado um nó do grafo e para cada transição de estado é associado um arco. O autômato  $G$  do exemplo anterior pode ser representado pelo grafo da Figura 2.6. Deve ser observado que a representação gráfica do autômato  $G$  é muito expressiva para aqueles treinados na interpretação de tais diagramas sendo muito mais clara do que a descrição matemática apresentada previamente. Considere dois autômatos que apresentam o mesmo par de segmento de entrada e segmento de

saída. Eles são ditos ser equivalentes no nível comportamental. Considere a classe de todos os autômatos. A equivalência no nível comportamental é uma relação de equivalência. O autômato de uma classe equivalente com cardinalidade mínima do espaço de estado é dito ser a realização mínima da classe equivalente.

**Definição 2.3.15** *Um sistema dinâmico quantizado ou sistema quantizado, para abreviar, é um sistema dinâmico no qual há um isomorfismo entre os conjuntos  $U, Y, X$  e um subconjunto de inteiros.*  $\square$

Como pode ser visto, existe uma forte conexão entre um sistema quantizado e um autômato. A principal diferença é que no autômato o “conjunto dos tempos” é um subconjunto dos inteiros, e então, todas as transições de estado ocorrem em valores precisos do tempo. Em um sistema quantizado, o “conjunto dos tempos” é um subconjunto dos números reais e as transições entre estados podem ocorrer em qualquer valor real do tempo.

### Nível de Rede

**Sistema Complexo** Finalmente, em um sistema dinâmico complexo, um sistema é descrito como se estivesse sendo especificado como construí-lo pela conexão conjunta de vários sistemas elementares. Tal descrição também pode ser chamada de descrição em *nível de rede*. As caixas pretas elementares são sistemas definidos como os descritos previamente. O acoplamento entre cada sistema elementar determina uma estrutura complexa. Este tipo de sistema é também chamado de *sistema hierárquico*.

**Definição 2.3.16** *Um sistema complexo é uma rede de sistemas interconectados.*  $\square$

Da definição apresentada decorre que um sistema complexo pode ser representado por um grafo direcionado ou dígrafo, onde os vértices representam os sistemas componentes (subsistemas) e os arcos as interações entre subsistemas.  $\square$

A Figura 2.7 representa os principais conceitos dos sistemas apresentados. Sistema geral, sistema orientado, sistema temporal são conceitos ao nível comportamental. Sistema funcional e dinâmico ao nível de estado, e sistema complexo ao nível de rede. Nesta figura, os conceitos de sistema geral, orientado, temporal e funcional estão ligados por flechas indicando que os conceitos podem ser obtidos dos precedentes juntando mais estrutura matemática na definição. O sistema dinâmico é obtido quando simultaneamente o sistema é temporal e funcional. Por outro lado,



as linhas ligando os pontos representativos dos conceitos de sistema geral, orientado, funcional, temporal e dinâmico ao de sistema complexo são tracejadas, bem como o ciclo se fechando em torno do nó representativo do conceito de sistema complexo. Elas indicam, não uma hierarquia perfeita de conceitos, e sim, que o sistema complexo é constituído pelo menos por dois sistemas que podem ser de qualquer dos outros tipos, ou ainda de um outro sistema complexo.

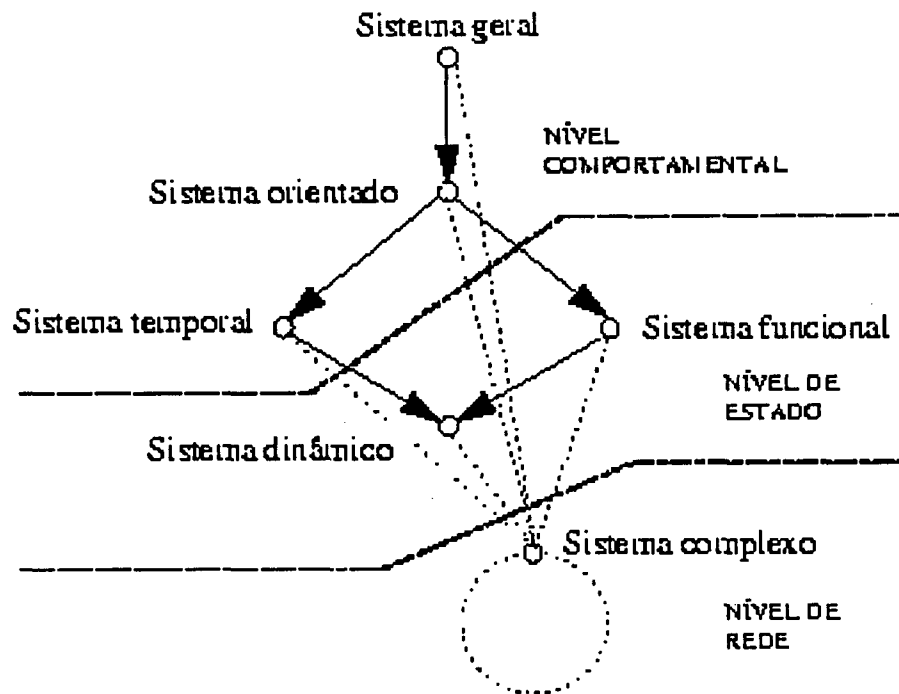


Figura 2.7: Hierarquia de tipos de sistemas

Exemplos de representações de sistemas complexos úteis nos mais diversos campos do conhecimento são:

- Os diagramas de blocos usados em sistemas de controle.
- Os “bonds graphs” de um sistema.
- Os fluxogramas usados na programação de computadores.
- Os gráficos conceituais [174] utilizados na representação de conhecimento.

Cada subsistema pode, muitas vezes, ser particionado em outros subsistemas, dependendo do propósito do estudo. A Figura 2.8 mostra um exemplo de sistema

complexo, constituído por 7 sistemas elementares, com entrada  $U$  e saída  $Y$ . Este sistema apresenta dois laços fechados criados pelo sinal indo de  $S_7$  a  $S_2$  e  $S_7$  a  $S_4$ .

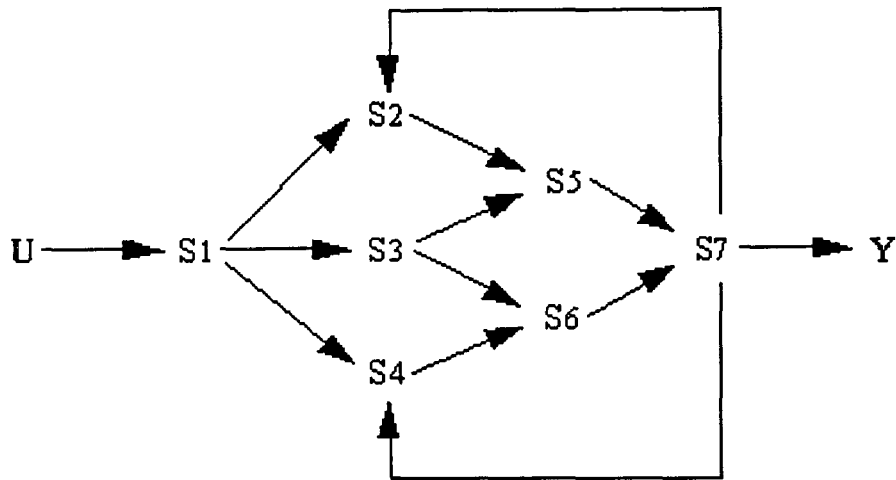


Figura 2.8: Exemplo de sistema complexo

Tal sistema é dito apresentar realimentação (ou retroação), pois o grafo correspondente apresenta ao menos um ciclo (ex:  $S_4, S_6, S_7, S_4$ ) e o seu comportamento é geralmente muito mais complexo que em um sistema sem realimentação. Uma regra concernente a diferentes descrições possíveis de um sistema é que, dada uma especificação de sistema em um certo nível, pode-se associar, quando muito, uma especificação no próximo nível superior. Então, uma descrição de estruturas significativas tem uma única descrição de estrutura de estado a qual tem uma única descrição comportamental. Por outro lado, indo mais fundo na descrição do sistema tem-se, em geral, diversas escolhas. Por exemplo, a um mesmo sistema descrito como um sistema orientado pode corresponder vários subsistemas do tipo funcional, pela definição de diferentes conjuntos de estados. A Figura 2.7 apresenta uma hierarquia dos conceitos de sistemas.”

### 2.3.2 Modelo de Agente

Conforme [59], o sucesso e a potencialidade de uma abordagem depende, primeiramente, da habilidade desta abordagem de representar, seja através de um modelo formal, seja através de uma representação abstrata, as características mais importantes do sistema que se está considerando. Em seguida, da capacidade de se manipular

esta modelagem formal do sistema, de modo a se obter novos teoremas e conjecturas, facilitando a tarefa de projeto e análise deste sistema.

Uma vez que foi apresentado, de maneira informal, um conceito básico de agente autônomo comum a vários autores e a fundamentação formal de uma ferramenta para a modelagem de sistemas, procura-se nesta seção empregar esta ferramenta para mostrar que o modelo de agente autônomo pode ser descrito como um sistema dinâmico [157] [159].

**Teorema 2.3.1** *Um Agente Autônomo pode ser modelado como um Sistema Dinâmico.*

**Prova:** Basta identificar as várias variáveis e funções com as da definição de sistema dinâmico. Assim, na definição de sistema dinâmico, (definição 2.3.9) se considera o seguinte objeto matemático:

$$S = \{T, U, \Omega, Y, \Gamma, X, \Phi, \eta\}$$

onde:

$T$  é o conjunto dos tempos;

$\Omega$  é o conjunto das funções de entrada  $\omega \in \Omega = \{\omega : T \rightarrow U\}$ ;

$U$  é o conjunto dos valores da entrada;

$Y$  é o conjunto dos valores da saída;

$\Gamma$  é o conjunto das funções de saída  $\gamma \in \Gamma = \{\gamma : T \rightarrow Y\}$ ;

$X$  é o conjunto dos estados;

$\Phi$  é a função de transição dos estados:  $\Phi : T \times T \times X \times \Omega \rightarrow X$ ;

$\eta$  é a função de saída:  $\eta : T \times X \times U \rightarrow Y$ .

Identificando com as variáveis e funções do agente tem-se:

$T$  é um conjunto que representa instantes de tempo e serve para ordenar eventos. Escolhas usuais são um intervalo do conjunto dos números reais ou um subconjunto do conjunto dos números inteiros;

$\Omega$  é o conjunto das funções de entrada provenientes do ambiente, ou seja, um sinal recebido pelos sensores do agente. Pode ser considerado como uma seqüência de vetores do espaço vetorial  $\mathcal{R}^n$ ;

$\Gamma$  é o conjunto das funções de saída, cujos valores são representativos dos valores dos atuadores do agente, ou seja, um sinal produzido pelo agente. Pode ser considerado como uma seqüência de vetores do espaço vetorial  $\mathcal{R}^m$ ;

$U$  é o conjunto dos valores de entrada. É o conjunto que representa a parte do estado do ambiente que é observada pelo agente. É um

conjunto de vetores onde cada elemento corresponde a uma entrada dos sensores;

$Y$  é o conjunto dos valores de saída. É o conjunto que representa os valores dos atuadores do agente que afetam o ambiente. É um conjunto de vetores, onde cada elemento corresponde a uma saída dos atuadores;

$X$  é o conjunto dos estados internos possíveis ao agente, ou seja, seus estados cognitivos. Ele representa a memória do agente, ou seja, o resíduo da história que afeta a sua ação atual e futura;

$\Phi$  é a função de transição dos estados:  $\Phi : T \times T \times X \times \Omega \rightarrow X$ , cujo valor é o estado  $x(t) = \Phi(x, u, t, \gamma)$  em  $X$ . É a função que define como o estado cognitivo se altera em função de estados cognitivos anteriores e dos valores dos sensores do agente;

$\eta$  é a função de saída:  $\eta : T \times X \times U \rightarrow Y$ , é a função que transforma para um dado instante o estado cognitivo e as entradas do agente nos valores de saída dos atuadores.

Isto fornece um método direto para a implementação de AAs e permite precisar vários outros conceitos relacionados. □

Conforme já visto anteriormente, para um dado sistema dinâmico corresponde um sistema temporal, além disso podemos dizer que um dado sistema temporal pode ser descrito pelo conjunto de todos os pares de entrada-saída apresentados pelo sistema, e que estes pares constituem o Comportamento Observável do sistema.

**Definição 2.3.17** *À função  $\beta$  que mapeia os valores de entrada  $U$  dos sensores e seu estado cognitivo  $X$  em valores de saída para os atuadores, chamamos de Comportamento Observável do Agente.*

*É importante ressaltar que o comportamento observável do agente pode ser alterado alterando-se a função de saída  $\eta$  ou a função de transição de estados  $\Phi$  (pois esta altera os valores de  $X$ ).* □

Uma propriedade muito importante que um AA deve possuir é a capacidade de modificar seu Comportamento Observável em função de experiências passadas, em outras palavras, a sua Capacidade de Aprendizado. Como aqui a capacidade de aprendizado está sendo formulada em termos de mudanças no nível comportamental, podemos esperar mais de uma solução ao nível de estado. Uma solução bastante intuitiva é através da modificação da função de saída  $\eta$  ou da função de transição de estados  $\Phi$ .

Considere-se, por exemplo, o diagrama de estados de um autômato que por simplicidade possui apenas quatro estados e um alfabeto de entrada de dois valores. As transições de estados estão representadas por setas na Figura 2.9.a. Suponhamos que após receber uma entrada o comportamento do autômato inicialmente representado pela Figura 2.9.a deva se alterar como consequência do aprendizado, passando a ser representado pelo autômato mostrado na Figura 2.9.b. Nota-se claramente que o aprendizado, neste caso, foi obtido através de uma alteração na função de transição de estados  $\eta$  do agente.

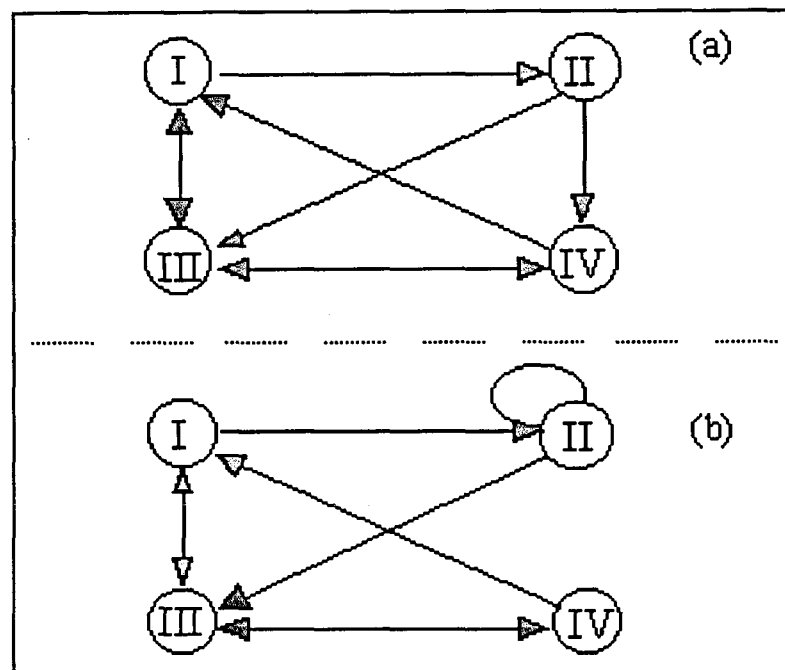


Figura 2.9: Diagrama de Estados de um AA antes e após um aprendizado

Uma segunda maneira de implementar esta mudança de comportamento pode ser através de um diagrama de estados contendo oito estados, como mostrado na Figura 2.10. O aprendizado agora ocorre se, estando no estado IV o agente receber uma entrada específica. Assim, nesta segunda abordagem, a transição do estado IV para o estado V implementa esta alteração no comportamento. Em uma implementação, a modificação da função de transição de estados não aparece de maneira explícita no diagrama de estados.

Generalizando, quando se utiliza esta segunda alternativa para implementar o aprendizado, é necessário possuir de antemão todos os autômatos possíveis e os diversos “agrupamentos” de estados. O aprendizado ocorre como consequência da

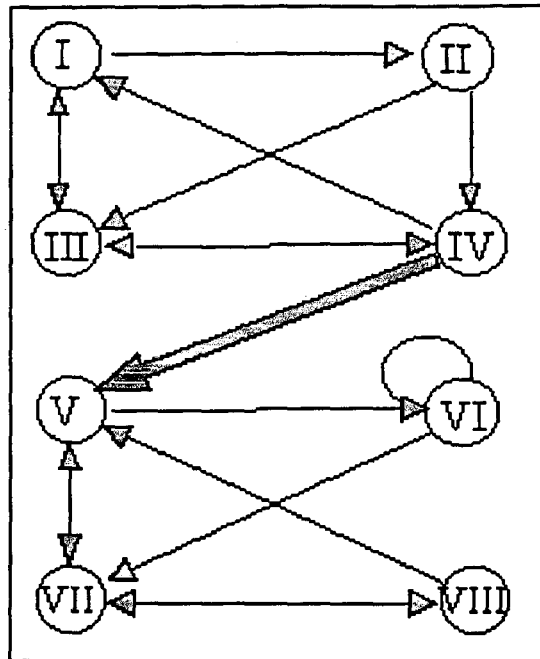


Figura 2.10: Aprendizado como uma transição “especial” de estados

transição de um “agrupamento” para outro. Se por um lado este segundo método é de certo modo simples de ser implementado, por outro lado, ele se torna inviável no momento em que precisamos conhecer previamente todos os comportamentos, o que faz o número de estado crescer drasticamente. Assim, a partir de agora, consideramos apenas a possibilidade da modificação das funções  $\eta$  e  $\Phi$  ao investigarmos o aprendizado de um agente.

**Definição 2.3.18** *Ao algoritmo que é capaz de modificar as funções  $\Phi$  e  $\eta$  de modo que o comportamento do agente se aproxime de um comportamento desejável, chamamos de Algoritmo de Aprendizado.* □

**Definição 2.3.19** *O grau de autonomia de um agente é dado pela capacidade do sistema de modificar suas funções  $\Phi$  e  $\eta$  em função dos valores anteriores de  $\Phi$ , das entradas dos sensores e um parâmetro interno  $\beta$  que representa o objetivo a ser atingido pelo agente. Este objetivo  $\beta$  pode ser um comportamento, ou seja, um conjunto de valores de estado interno  $X$  e de saída  $Y$  válidos ou desejáveis.* □

Nos capítulos seguintes mostraremos que esta formalização não pretende ser apenas uma forma estética e elegante de apresentação para um conjunto de conceitos

ainda mal-definidos. A combinação de diferentes sistemas e ferramentas, como é o caso de agentes autônomos e redes neurais artificiais sob o mesmo contexto - a Teoria de Sistemas - conduz a uma metodologia para a implementação de AAs. Propõe-se que os autômatos de estados finitos constituem uma ferramenta eficiente para a descrição de comportamentos que se deseja implementar em um agente, e que estes autômatos podem então ser modelados através de diferentes estruturas de redes neurais artificiais.

### 2.3.3 Modelo de Ambiente

O ambiente, do ponto de vista do agente, é tudo que é exterior ao agente propriamente dito, incluindo outros agentes e processos. Deste modo, pode-se dizer que o ambiente pode ser modelado como um sistema complexo composto por uma gama enorme de sistemas dinâmicos. Esta enorme gama de sistemas, conectados entre si das mais diversas maneiras e repletos de ciclos, faz com que, sob o ponto de vista prático, o ambiente seja um sistema não-determinístico e aberto, pois, na maioria das situações, é impraticável se conseguir modelar todos os sistemas que fazem parte do ambiente e como estes sistemas estão interconectados.

Para fins de estudo, é interessante que se possa trabalhar com um modelo mais simples de ambiente. Uma simplificação natural é considerar o ambiente como um sistema dinâmico, mais especificamente, podemos usar a inspiração fornecida por Kaelbling [90] e modelar o ambiente como um autômato

$$G = \{U, Y, X, \Phi, \eta\}$$

onde:

$U$  é o conjunto dos valores da entrada, neste caso, o conjunto de todas as saídas possíveis (ou ações) geradas pelo agente no ambiente;

$Y$  é o conjunto dos valores da saída, neste caso, o conjunto de todos os sinais externos que refletem o estado do ambiente e que eventualmente podem ser utilizados como sinais de entrada para o agente;

$X$  é o conjunto dos estados, é o conjunto de todos os estados possíveis para o ambiente;

$\Phi$  é a função de transição dos estados que mapeia  $\Phi : X \times U \rightarrow X$ ;

$\eta$  é a função de saída que mapeia  $\eta : X \times U \rightarrow Y$ .

Neste modelo, a inconsistência do ambiente é fundamental para exercitar as características de autonomia do agente.

Segundo Kaelbling [90], um ambiente é aparentemente inconsistente para um agente se for possível que, em duas ocasiões diferentes em que o agente receba as mesmas entradas e gere as mesmas ações, o próximo estado do ambiente, refletido pelas novas entradas sentidas pelo agente, são diferentes.

Esta inconsistência pode ser provocada por diferentes razões:

- O agente não possui capacidade de discriminar todos os diferentes estados do ambiente. O número e capacidade dos sensores do agente não permite fazer um mapeamento um-para-um entre todos os estados do ambiente.
- As entradas dos sensores podem estar contaminadas por ruído ou ainda estar parcialmente inoperantes.
- As saídas dos atuadores podem não possuir capacidade de alterar o estado do ambiente ou ainda estar parcialmente inoperantes.
- A função de transição de estados  $\Phi$  pode ser probabilística. Neste caso, o ambiente é um autômato estocástico e o novo próximo estado  $X$  da máquina de estados é determinado por uma função de distribuição de probabilidades.
- Estados não observáveis do ambiente para o agente ou para o algoritmo de aprendizado.

### 2.3.4 Interação entre Agente e Ambiente

O modo como um agente interage com o ambiente é através do seu comportamento observável. Durante o desenvolvimento de um AA, o projetista muitas vezes se depara com o problema de como descrever o comportamento observável desejado para o AA [173]. O objetivo desta seção é introduzir uma abordagem que, assim como os conceitos formais propostos anteriormente, seja livre de conotações psicológicas, imprecisas e informais, de modo a poder facilitar a tarefa de especificar os comportamentos desejados para um agente durante o processo de desenvolvimento. Este tratamento formal é extremamente atrativo, pois uma vez que o comportamento observável possa ser descrito e especificado, pode-se utilizar ferramentas matemáticas para modelamento, análise e otimização que facilitem a tarefa de implementação deste comportamento.

De fato, existem algumas propostas para esta especificação. Smithers [173] coloca que a interação entre um agente e seu ambiente forma o espaço de fase de um sistema dinâmico; neste caso, um sistema dinâmico produzido pela interação entre os processos do agente e do ambiente que formam a realidade do agente.



Segundo a abordagem proposta neste trabalho, como o comportamento observável do agente é dado pela função de saída  $\eta$ , a tarefa do projetista de um agente autônomo é especificar esta função.

Suponhamos, como exemplo, que um agente esteja sendo projetado para se mover em um ambiente. Um comportamento básico que este agente deve possuir é o de evitar obstáculos. Em outras palavras, ele não deve ficar “trancado” ou bater em algum obstáculo. Não importa quão fácil ou difícil seja implementar este comportamento, qualquer robô móvel deve possuí-lo, para que ele continue sendo um robô MÓVEL. Entretanto, “evitar obstáculos” e “não ficar trancado” são termos informais utilizados para especificar um comportamento básico de um agente.

Uma vez que definiu-se que um agente autônomo é um sistema dinâmico, pode-se utilizar um autômato, que é um tipo de sistema dinâmico, para modelar este agente. Assim, uma maneira mais conveniente de descrever este comportamento é especificar que não importa qual estado, ou o conjunto dos valores de entrada do autômato, a função de saída deve ser tal que, o próximo conjunto de valores de entrada seja diferente do anterior. A questão agora, de como gerar este autômato, é deixada a cargo do algoritmo de aprendizado.

### 2.3.5 Medidas de Desempenho

Para que se possa comparar diferentes alternativas de implementação de AAs e algoritmos de aprendizagem de comportamentos, é necessário que se estabeleça um critério que permita julgar os agentes sob consideração. Kaelbling [90] propõe três aspectos principais sob os quais se pode comparar os comportamentos ou os algoritmos de aprendizagem implementados em diferentes agentes: correção, convergência e complexidade espaço-tempo. Assim, uma vez determinado um comportamento correto para um agente em um dado domínio, poderia se medir com que grau de correção o comportamento aprendido pelo agente se aproxima do comportamento desejado, com que velocidade este comportamento se aproxima do comportamento desejado e o espaço e tempo necessários para implementar o algoritmo de aprendizagem.

Estamos então, claramente, diante de um problema de otimização. De acordo com Barreto [17], um problema de otimização pode ser considerado como o objeto matemático:

$$\{\Sigma, S, J, L\}$$

onde:

$\Sigma$ : é o sistema a ser otimizado;

$S$ : é o espaço de soluções;

$J$ : é a função que mede a qualidade de cada solução em particular;

$L$ : é o conjunto de valores da função  $J$ .

Para o caso da implementação de um comportamento, começamos com um objetivo a ser alcançado, mais especificamente a descrição do comportamento observável desejado, como mencionado na seção anterior. Restrições, como por exemplo, a quantidade e precisão dos sensores, são colocadas. Uma vez que o objetivo e as restrições estão expressos em termos matemáticos, utilizamos um processo, que pode ser, por exemplo, um algoritmo evolucionário, um algoritmo de aprendizado de rede neural artificial ou mesmo estes dois processos combinados, para gerar ou construir um modelo do agente. O agente gerado é posto para operar e o comportamento observado é comparado com o comportamento desejado especificado no objetivo, permitindo validar ou descartar o agente gerado.

# Capítulo 3

## Comportamento Animal

### 3.1 Introdução

Ao buscar na Natureza inspirações para o projeto e desenvolvimento de AAs, vemos que qualquer animal possui uma capacidade de sobrevivência e “operação”, no mundo real, muito superior à de qualquer AA já desenvolvido. A idéia então é procurar observar quais são os mecanismos comuns utilizados pelos animais na sua luta pela sobrevivência com o objetivo de empregá-los como fonte de inspiração no desenvolvimento de AAs. A este mecanismo utilizado pelos animais para se adaptar ao meio, transformando as informações obtidas do ambiente em ações que os distanciem de efeitos nocivos ou que os aproximem de situações propícias ou, ainda, que alterem o ambiente próximo aumentando as suas chances de sobrevivência, chamamos de comportamento animal [81].

O estudo sistemático do comportamento animal começou durante a década de 30, com os trabalhos de Konrad Lorenz [110] e Niko Tinbergen [181]. Lorenz e Tinbergen procuraram criar modelos que explicassem o comportamento previsível de certos animais quando submetidos a estímulos específicos. Eles chamaram todo o comportamento exibido de padrões fixos de ação. Estes comportamentos eram acionados por energia que era acumulada a medida que o tempo passava e que era liberada quando o comportamento era disparado. Em seus primeiros trabalhos, Lorenz procurava fazer uma classificação bastante rígida entre comportamentos inatos e comportamentos aprendidos. Esta questão gerou bastante polêmica na comunidade acadêmica e hoje em dia este tipo de classificação já se mostra ultrapassada, pois experiências têm demonstrado que muitos comportamentos ditos inatos e altamente estereotipados são influenciados pela experiência anterior e maturidade do organismo [187] [8] [58].

Hoje em dia, o comportamento animal é tema de estudo de naturalistas, etólogos,

ecologistas, fisiólogos, anatomistas, psicólogos, sociólogos entre outros. Cada um destes especialistas utiliza abordagens distintas para analisar a questão. Etologistas estudam o comportamento animal em seu ambiente natural, tomando como premissa a crença de que o comportamento é em grande parte uma resposta inata do animal a certos estímulos ambientais. Os psicólogos estudam o comportamento animal em ambientes controlados de laboratórios, procurando esclarecer os princípios que regem o aprendizado de comportamentos como resposta a certos estímulos. Já os fisiologistas estão interessados em descrever o comportamento animal ao nível neurológico. Material bibliográfico interessante para quem desejar se aprofundar neste tema pode ser encontrado em [61], [58], [124], [187], [110] e [181].

Entre as várias definições encontradas na literatura para comportamento animal, vamos aqui reproduzir a definição fornecida por Messenger [124] por parecer ser uma das mais completas e abrangentes.

*“Pode-se dizer que (o comportamento animal) abarca todas as atividades de procurar comida, evitar perigo, acasalar-se e mesmo criar uma prole. Todas essas atividades requerem órgãos dos sentidos que coletam informações, principalmente sobre as alterações no mundo externo, um sistema nervoso que processa estas informações, e sistemas efetores, tais como glândulas, músculos ou fotóforos que traduzem a saída (output) do sistema nervoso. O que caracteriza o comportamento tipicamente, entretanto, é o fato de manifestar-se através da atividade - ou inatividade - do sistema muscular. Uma anêmona ao fechar-se, um leão ao dar o bote, um chipanzé que sorri, uma prima-dona ao cantar: de algum modo estão se comportando ao executarem tais atos. Da mesma forma, também o cão de Sherlock Holmes não fazia coisa alguma durante a noite: freqüentemente a resposta biologicamente apropriada a um estímulo pode ser não dar resposta alguma.”*

Para a finalidade deste trabalho, queremos examinar sobre que princípios se organiza o comportamento animal e como estes princípios podem ser úteis para o desenvolvimento de AAs. O objetivo é dividir e classificar o comportamento animal em unidades significativas, analisá-lo e ver com certa minúscia como ele é dependente do funcionamento do sistema nervoso [61]. Queremos observar, também, como o comportamento pode ser influenciado pela hereditariedade e pelo aprendizado.

Outros pesquisadores têm utilizado a observação do comportamento animal como fonte de inspiração para a sua implementação em modelos artificiais. Trabalhos envolvendo esta abordagem podem ser encontrados em [28] e [8].

## 3.2 Fatores que determinam o comportamento

Ao analisar-se a série filogenética, pode-se reconhecer uma enorme variação de complexidade de comportamento, desde o ato simples, breve, estereotipado, até uma longa seqüência de atos altamente complexos e variáveis [61]. Percebe-se, então, que a hereditariedade, o sistema sensório-motor e a aprendizagem estão intimamente relacionados e exercem grande influência na geração de um comportamento. A hereditariedade determina, no animal, o sistema sensório-motor, ou seja, o tipo de cérebro, órgãos sensoriais e sistema motor. Este aparato nervoso, por sua vez, vai determinar a capacidade de aprendizado do animal, que vai possibilitar uma capacidade de sobrevivência maior e, portanto, uma maior possibilidade de gerar descendentes com as mesmas características hereditárias. Em resumo, todo comportamento, com exceção de reflexos incondicionados, é influenciado pela aprendizagem, porém, todo ele, reflexo ou não, é basicamente influenciado pela hereditariedade. Esta, por sua vez, vai determinar a estrutura e complexidade do sistema sensório-motor, dos quais o aprendizado depende inteiramente [81]. Este relacionamento entre hereditariedade, sistema nervoso e aprendizado pode ser visto de maneira esquemática na Figura 3.1.

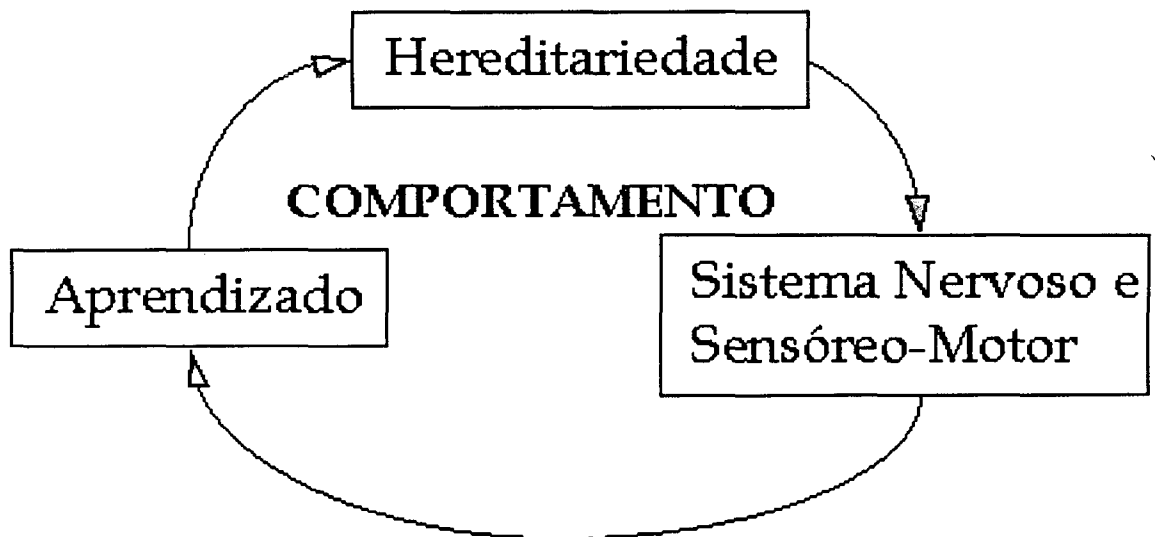


Figura 3.1: Representação esquemática do relacionamento entre hereditariedade, sistema nervoso e aprendizado na determinação do comportamento de um animal

### 3.3 Classes de comportamentos

Segundo Hebb [81], “em última análise, todo o comportamento é uma reação à estimulação ambiental, mas a relação entre estímulo e resposta varia entre o direto e o extremamente indireto. Num extremo se encontra o reflexo incondicionado, em que o estímulo tem efeito imediato; as conexões neurais entre sensor e atuador são diretas. No outro extremo, o estímulo tem o efeito de excitar circuitos corticais complexos e o efeito comportamental pode ser longamente retardado (por exemplo, quando ocorre a aprendizagem latente)” [81].

O texto a seguir, extraído de Dethier & Stellar [61] apresenta de maneira bastante clara e resumida as diversas gamas de comportamento animal e sua relação filogenética.

*“No início, em filogenia, o comportamento é em grande parte uma questão de um estímulo desencadear uma resposta, ou de um padrão de estímulos desencadear uma seqüência de respostas. Neste ponto, o comportamento é estereotipado e o organismo é em grande extensão limitado pelo estímulo. Como este tipo de comportamento é essencialmente a revelação de propriedades herdadas do sistema nervoso do organismo ou da espécie em questão, é chamado de inato. Mais tarde, o comportamento torna-se mais variável e, o que é particularmente importante para nós, torna-se passível de modificação através da experiência. As adaptações de um organismo individual podem se desenvolver unicamente na história da sua vida, através do processo de aprendizado e denominamos estes padrões de comportamento de adquiridos. Nos organismos mais simples, o que é adquirido pode ser bastante simples e ainda bastante limitado pelo estímulo. Mas, à medida que a complexidade do sistema nervoso cresce, surgem novas propriedades do comportamento. O comportamento agora não é tão limitado pelo estímulo; uma boa parte dele pode se originar dentro do organismo, com base na experiência anterior; e boa parte pode ser guiada por processos simbólicos complexos, como a linguagem no homem. Estes processos intrínsecos constituem o raciocínio no homem e encontramos os rudimentos dele em comportamento animal que serve aos fins adaptativos do organismo” [61].*

Ao analisar-se o texto pode-se perceber que existem várias classes de comportamento e que uma se funde na outra sem uma linha divisória distinta. A grosso modo, podemos classificar dois tipos principais de comportamento. Um é o comportamento estereotipado, em que a ação do organismo aparece claramente como uma resposta a um estímulo, seja ele originado do ambiente externo, ou do estado interno do animal como fome, temperatura, hormônios, etc. A outra grande classe de comportamento é aquele em que não se consegue mapear claramente o processo estímulo-resposta do comportamento, pois boa parte dele pode se originar dentro

do organismo como processos simbólicos complexos que envolvem o “pensar” e o raciocínio. Este tipo de comportamento só parece estar presente no homem e em algumas espécies de primatas.

A Figura 3.2 apresenta de forma esquemática a forma de comportamento dominante nos diferentes níveis da escala filogenética. Note-se que existe uma hierarquia das classes de comportamento, e que a medida que se evolui na escala filogenética, os comportamentos mais complexos vão se sobrepondo e de certa forma, “mascarando” os comportamentos mais primitivos. Por exemplo, se estivermos segurando um objeto muito quente em nossas mãos, a ação de nosso comportamento reflexo é a de soltar este objeto. No entanto, se este for um vaso de porcelana muito valioso, nosso comportamento racional nos levará a tentarmos continuar segurando-o.

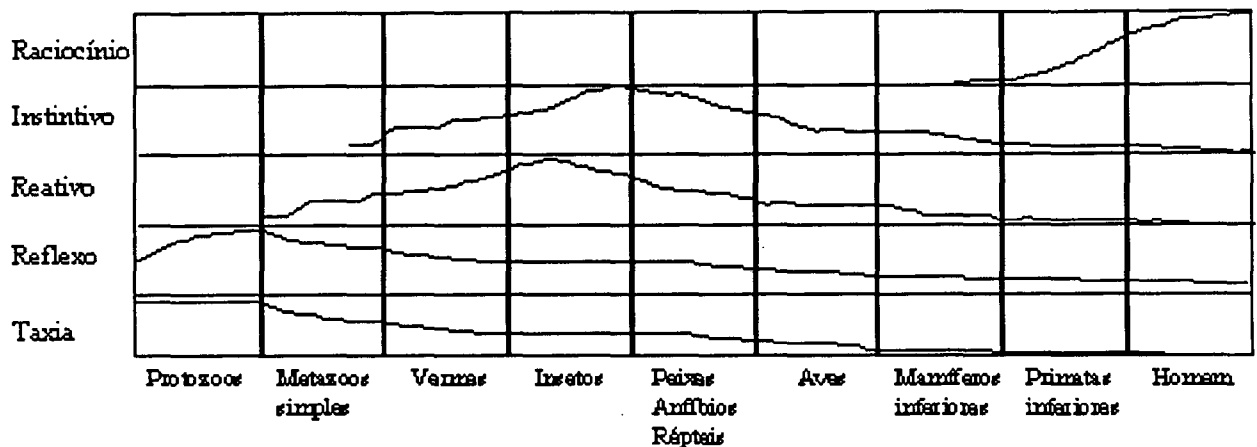


Figura 3.2: Apresentação esquemática da forma de comportamento dominante em série filogenética

### 3.3.1 Comportamentos estereotipados

Apesar de ser difícil estabelecer claramente fronteiras e linhas limítrofes entre as diferentes classes de comportamento, diversos pesquisadores têm procurado classificar os diversos tipos de comportamentos apresentados pelos animais. Os critérios para classificação podem envolver as estruturas neurais envolvidas nos comportamentos, a série filogenética, a capacidade de aprendizado do comportamento, etc. Da mesma forma, o número de classes e a nomenclatura atribuída a cada classe de comportamento varia de pesquisador para pesquisador.

As classificações de comportamentos estereotipados apresentadas a seguir foram em grande parte extraídas de Dethier & Stellar [61].

## Taxias

*“Talvez a mais simples forma de comportamento adaptativo seja a orientação de um organismo em relação a algum aspecto do seu ambiente. Um exemplo de taxia seria a orientação de manter estímulo igual de dois receptores bilateralmente simétricos ou, por movimentos alternados para a esquerda e a direita, a uniformização do estímulo em intervalos sucessivos de tempo. Por exemplo, um organismo pode se orientar em direção a uma fonte de luz de tal forma que os dois olhos recebam estímulo igual. Se a fonte for deslocada lateralmente, a orientação mudará porque um olho está recebendo agora mais iluminação que o outro. Se um olho for removido ou pintado, o organismo irá se mover continuamente em círculos, como se estivesse tentando igualar a luz nos dois olhos. Esta orientação, guiada contínua e especificamente por estímulos externos, é chamada de taxia.*

*Embora seja possível demonstrar muitos exemplos precisos de orientações relativamente fixas e estereotipadas a estímulos específicos, no reino animal nem todas as taxias ocorrem tão simplesmente na Natureza. Todos os animais mostram grande variabilidade no seu comportamento e não são completamente fixos em suas respostas de orientação. A razão disto, naturalmente, é que um organismo vivo responde a mais de um aspecto do seu ambiente e do estado interno, em qualquer momento, e pode estar fazendo várias adaptações diferentes ao mesmo tempo. Quanto mais subirmos na escala filogenética e mais modos de adaptação um animal possuir, tanto mais variável será seu comportamento, e assim, menos fixa e estereotipada a sua orientação.”*

## Reflexos

*“Muito semelhante às taxias são os reflexos, os quais são respostas relativamente estereotipadas e fixas a estímulos que se ajustam à definição de comportamento inato no sentido de serem eles a revelação de mecanismos nervosos herdados. De uma forma geral, as taxias envolvem uma orientação de todo o corpo que pode envolver várias respostas reflexas específicas. Os reflexos, como o reflexo de susto ou de endireitamento, podem envolver todo o corpo ou a maior parte dele, mas eles são, na forma típica, respostas de parte do corpo, como a flexão de perna em resposta a estímulos dolorosos, ou a constrição da pupila provocada pela luz intensa. Muitos padrões de comportamento são séries complexas de reflexos simples. A resposta reflexa é um dos modos principais de adaptação no reino animal. No curso da evolução contudo, os reflexos tornaram-se aspectos menos proeminentes do comportamento, pois eles se tornaram mais variáveis e cada vez mais sujeitos a influências modificadoras dos mecanismos nervosos superiores e são obscurecidos pelos outros modos*



*de adaptação.”*

Uma característica que parece ser marcante tanto aos comportamentos reflexos quanto às taxias se refere ao fato de que a intensidade e a duração das respostas dos organismos são uma função direta da intensidade e duração do estímulo que disparou o comportamento. Também é importante notar que o tempo decorrido entre a ocorrência do estímulo e o aparecimento da resposta é mínimo.

### Comportamento Reativo

Uma classe intermediária entre os comportamentos puramente reflexivos e os comportamentos instintivos é a classe de comportamento que chamamos de reativo ou seqüencial e que foi chamado por Beer [28] de padrão fixo de ação. Esta classe de comportamento é formada por uma série de comportamentos estereotipados como resposta a um dado estímulo. O estímulo que dispara o comportamento é geralmente mais complexo e específico que o necessário para disparar um comportamento reflexivo. A resposta, por sua vez, envolve uma seqüência temporal de ações que se desenrolam até o seu final, mesmo que o estímulo disparador não esteja mais presente. As respostas que compõem o comportamento reativo podem estar relacionadas de um modo intrincado, no qual cada resposta componente é disparada pelo final da ação precedente ou por algum estímulo ou sinal proveniente do ambiente e que é alcançado como resultado da ação precedente. Se qualquer dos sinais disparadores de uma ação da seqüência for inibido, todas as ações seguintes não serão disparadas, mesmo que os sinais seguintes sejam apresentados.

Um exemplo típico desta classe de comportamento pode ser vista no exemplo apresentado por Anderson & Donath [8] e que foi extraído de [118] descrevendo o comportamento apresentado por certa espécie de vespa.

#### Exemplo 3.3.1

*“A fêmea de certa espécie de vespa emerge de sua pupa subterrânea na primavera. Seus pais já estão mortos desde o verão anterior. Ela deve se acasalar com um vespa macho e então realizar toda uma série complexa de ações relacionadas com a escavação de um buraco para o vespeiro, construindo células dentro dele, caçando e matando presas como lagartas, estocando as presas capturadas nas células do vespeiro, colocando ovos nas células e então as fechando. Todas estas ações devem ser completadas dentro do espaço de poucas semanas, depois das quais a vespa morre.”*

Neste exemplo, o primeiro estímulo disparador para a seqüência de comportamentos é a presença do vespa macho, que habilita o comportamento de acasalamento. O resultado do acasalamento dispara a ação de construção do vespeiro, cujo resultado, por sua vez dispara a ação de caça e assim por diante. Se qualquer dos resultados disparadores não puder ser atingido, todas as ações seguintes da seqüência serão inibidas [8].

### Comportamento Instintivo ou Motivado

*“Os padrões de comportamento inato mais complexos e mais fascinantes são sem dúvida os instintos. Podemos começar com o ponto de vista dos etólogos, pois eles têm oferecido as concepções mais completas e mais gerais. Eles estabelecem dois pontos importantes que distinguem comportamento instintivo de tacias e de reflexos. Primeiro, acentuam que o comportamento instintivo freqüentemente depende de alguma condição especial do ambiente interno do organismo. Por exemplo, muitos aspectos do comportamento reprodutivo dependem da presença dos hormônios sexuais, com o resultado que em um extremo não há resposta positiva a um estímulo sexual forte, na ausência de hormônios, e no outro, quando a concentração de hormônio é alta, é necessário apenas estímulo mínimo para despertar um padrão completo de comportamento sexual. Em alguns casos, o estado interno pode ser tão forte que conduz quase diretamente ao comportamento, sem qualquer estímulo provocador mensurável; esta é a chamada reação no vácuo. O segundo ponto é que os estímulos servem apenas para desencadear o comportamento instintivo e não são sempre necessários para guiá-lo através do padrão total. O ganso cinzento, por exemplo, recupera um ovo que rolou fora do ninho, empurrando-o entre suas pernas com o lado inferior do bico e pode continuar estes movimentos diligentes de empurrar até “o fim”, mesmo que o ovo tenha rolado fora do seu alcance.*

*Cada um dos atos instintivos é desencadeado pelos efeitos combinados dos estímulos externos, hormônios e influências nervosas centrais excitatórias.*

*No lado “behaviorista”, muitos psicólogos americanos acentuam os aspectos motivacionais do instinto e iniciam com a concepção de que muitos padrões do comportamento instintivo podem ser analisados como um impulso dirigido para uma finalidade que, quando alcançada, resulta em redução do impulso, ou saciedade. Comportamento motivado, então, é um impulso que conduz a comportamento dirigido para a meta e a saciedade. Ele pode ser medido pela intensidade ou ritmo do comportamento consumatório, como no comer, beber e acasalar, ou pelo ritmo ou intensidade de trabalho que o animal realiza para atingir a própria meta, uma pequena fração dela, ou ainda outra meta aprendida.*

*No lado neurofisiológico, comportamento motivado ou comportamento instintivo têm sido investigados, descobrindo-se as partes do sistema nervoso que estão envolvidas e os efeitos nelas produzidos pelos estímulos externos e mudanças no ambiente interno. O hipotálamo tem sido o foco de atenção no trabalho com mamíferos, porque foi verificado que ele contém mecanismos excitatórios, cujas ações contribuem para o surgimento do comportamento motivado e mecanismos inibitórios, cujas ações contribuem para a redução do comportamento motivado. Por exemplo, no caso da alimentação, os pesquisadores descobriram que destruição das regiões ventromediana do hipotálamo, em ambos os lados, resulta em um grande aumento do comer, até o ponto em que um rato, um gato, ou um macaco poderiam dobrar ou triplicar o peso do seu corpo. Por outro lado, a destruição bilateral das regiões laterais do hipotálamo levam um animal a não comer até morrer de fome.*

*Mecanismos hipotalâmicos relativamente específicos semelhantes têm sido encontrados para a sede, o comportamento sexual, o comportamento emocional, o sono e o comportamento maternal, e estes achados conduziram à noção de que “o impulso é baseado na atividade de um mecanismo hipotalâmico excitatório e que a saciedade é baseada na atividade de um mecanismo hipotalâmico inibitório”. Acredita-se também que a ação de ambos estes mecanismos é controlada por estímulo sensorial relevante, mudanças no ambiente interno e influências do córtex cerebral.”*

### 3.3.2 Comportamento racional

À medida em que vamos evoluindo na escala filogenética vê-se que começam a aparecer novos comportamentos que cada vez menos estão relacionados de maneira clara e perfeitamente previsível a estímulos externos ao organismo. Com a crescente complexidade do sistema nervoso, o comportamento, bem como os mecanismos de aprendizado (que serão tratados no próximo capítulo), vão apresentando novas propriedades. Na classe de comportamento que chamamos de racional, boa parte do processo estímulo-resposta se origina dentro do próprio organismo, não mais como estímulos hormonais, como acontece no comportamento instintivo, mas como processos simbólicos complexos e, ainda hoje, não bem explicados sob o prisma neurológico, e que chamamos de pensamento ou raciocínio.

Segundo Dethier & Stellar, “raciocínio é a capacidade de resolver problemas complexos, com alguma coisa além de simples modificações por tentativa-e-erro, hábito ou estímulo-resposta”. No ser humano este comportamento se manifesta claramente na sua capacidade de desenvolver conceitos, comportar-se de acordo com princípios gerais e reutilizar os conhecimentos adquiridos de experiências anteriores em novas organizações, de maneira independente da forma física particular que um

problema assuma, ou dos elementos sensoriais ou motores específicos envolvidos na situação [61].

Diversas experiências têm sido feitas objetivando demonstrar com que grau diferentes animais são capazes de apresentar um comportamento racional. Uma das experiências mais antigas e simples é a chamada *problema de contorno*. Nesta experiência, um animal é estimulado por um alimento que ele pode ver e cheirar, mas que devido a existência de alguma barreira ou outro elemento ele não pode alcançá-lo diretamente. Para apanhar o alimento, o animal deve primeiro afastar-se dele, contornar a barreira e então fazer o percurso direto. A capacidade de um comportamento racional possibilitaria ao animal resolver o problema na sua primeira exposição a ele, sem ter de aprendê-lo por tentativa-e-erro ou através de soluções "ocasionais". De todos os animais testados em problemas de contorno, apenas macacos e chimpanzés mostraram qualquer grau de sucesso com a primeira exposição à situação, embora outras espécies de animais tenham aprendido a desempenhar contornos depois de terem fracassado nas suas primeiras poucas tentativas [61].

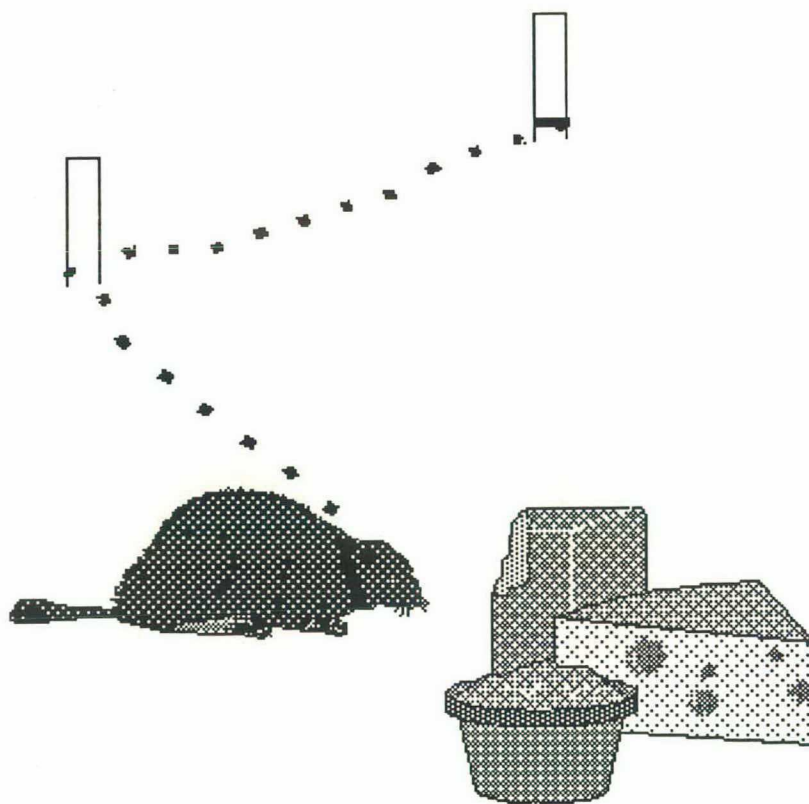


Figura 3.3: Um problema de contorno no qual o animal deve primeiramente se afastar do alimento a fim de atingi-lo

Outras formas de testar a capacidade de resolver problemas exigem que os animais aprendam a desempenhar uma tarefa de acordo com um princípio geral, muito distante dos estímulos específicos disponíveis em que a resposta é feita. Uma das experiências mais simples é a da *reação condicional*. Neste experimento existe uma seqüência de condições que devem ser satisfeitas para que a recompensa seja obtida. Em um experimento típico, um animal poderá ter de aprender a escolher a porta esquerda, entre duas portas de cor cinza, se ela fosse precedida por uma única porta preta, ou a porta direita se ela fosse precedida por uma única porta branca. Ratos podem aprender gradualmente este tipo de resposta, porém macacos aprendem reações condicionais mais facilmente e são mais versáteis e estáveis nas suas soluções [61].

Um experimento similar é o da habituação reversa. Neste tipo de experimento, tão logo um animal aprenda a escolher entre duas alternativas aquela considerada correta, e que portanto fornece uma recompensa, esta é transferida para a outra alternativa. A medida que esta troca continue, a função de aprendizado é determinada pelo tempo decorrido até que o animal perceba a troca. Macacos e ratos apresentam a capacidade de aprendizado deste problema, porém peixes não. Tartarugas mostram uma capacidade intermediária, pois conseguem aprender progressivamente o problema quando o fator discriminante é espacial (alto e baixo), mas não quando é visual (por exemplo, vermelho e verde, círculos e quadrados) [187].

Por fim, um experimento que demonstra a existência de uma capacidade de processamento simbólico é o problema da *reação retardada*. Neste experimento duas taças idênticas são apresentadas a um animal com fome, e é permitido que ele veja o alimento sendo colocado debaixo de uma das taças. Após um certo intervalo de tempo, o animal é libertado para fazer a sua escolha deslocando uma das taças. Com o odor controlado, o animal não tem nenhuma senha discriminativa no momento de escolher entre as duas taças a não ser a sua memória. Para que o animal consiga memorizar qual a taça contém o alimento durante o período de retardo, ele deve ser capaz de algum processo simbólico que ele pode utilizar para representar a senha discriminativa perdida na ocasião da escolha. Certos mamíferos, como cães, gatos e ratos resolvem este problema orientando-se na direção da taça com alimento durante o período de retardo, e então seguindo o rumo em direção à taça correta. Se eles interromperem a orientação durante o retardo ou se o experimentador os desorientar, eles fracassarão. Os primatas além de serem capazes de retardos muito mais prolongados, ainda podem escolher a taça correta, mesmo após removidos da situação de teste e devolvidos após o retardo [61].

### 3.4 Autômatos de Estados Finitos como ferramenta para descrever comportamentos

Na literatura específica sobre comportamento animal, os comportamentos apresentados são descritos de maneira discursiva. Isto é bastante compreensível, neste caso, pois na verdade, em um animal real agindo no seu ambiente natural, o que ocorre é uma grande variedade de estímulos ambientais e reações do organismo agindo simultaneamente, de modo que nem sempre fica muito claro quais comportamentos básicos estão presentes, nem qual é o dominante a cada instante de tempo. Mesmo em um ambiente controlado de laboratório, é impossível restringir o comportamento do animal apenas aquele que se deseja estudar ou descrever. Principalmente nos animais superiores, diversos processos mentais estão se desenrolando simultaneamente, podendo inclusive ocorrer de processos diferentes agirem sobre o mesmo conjunto de atuadores. O próprio processo de aprendizado introduz alterações, algumas vezes sutis, em um comportamento que se esteja tentando descrever.

No entanto, ao utilizar o comportamento animal como fonte de inspiração para a construção de AAs, torna-se necessário utilizar uma ferramenta que possibilite descrever de maneira precisa os comportamentos que se deseja transferir para um AA, mesmo à custa de simplificações que se façam necessárias para a descrição formal do comportamento ou simplificações introduzidas por eventuais limitações da própria ferramenta.

Uma ferramenta para descrição de comportamentos para AAs foi proposta por Brooks para a implementação da sua “subsumption architecture” [34] [35] [37]. Brooks utiliza Máquinas de Estado Finitas Expandidas (máquinas de estado finitas convencionais conectadas a um conjunto de registradores, temporizadores e alarmes) agrupadas de maneira a descrever um comportamento, através do que foi chamado de “linguagem do comportamento” (*behavior language*). A “linguagem do comportamento” agrupa vários processos (cada um deles normalmente implementados por uma máquina de estados finitos expandida) no que é chamado um *comportamento*. Na implementação de Brooks pode haver passagem, bloqueio e inibição de mensagens tanto entre processos de um mesmo comportamento, como entre comportamentos. Um programa escrito nesta linguagem forma um agrupamento de regras escritas com um subconjunto Lisp e que são executadas em paralelo e em tempo-real [36].

A finalidade da linguagem proposta por Brooks parece estar exclusivamente voltada para a implementação de robôs que utilizam a sua “subsumption architecture”, sem nenhuma referência a uma possível capacidade de sua utilização para a representação de comportamentos observados em animais reais. No entanto, a utilização

de redes de máquinas de estados finitos parece ser uma boa ferramenta para a descrição de comportamentos estereotipados observados em animais e que se deseje implementar em AAs.

Conforme o que já foi visto através da definição 2.3.13, se um Agente Autônomo pode ser modelado como um Sistema Dinâmico, então a utilização de Autômatos de Estados Finitos (AEFs) se mostra uma ferramenta ideal para descrever um AA que se deseje implementar, uma vez que o AEF pode ser também considerado como um sistema dinâmico discreto no tempo com conjuntos finitos de entrada e saída, permanecendo, portanto, no mesmo nível hierárquico de descrição de sistemas. Este sistema responde a um número finito de diferentes estímulos (o conjunto de entrada ou alfabeto de símbolos) e possui um repertório finito de comportamentos (o alfabeto de saída), que depende das entradas apresentadas e do estado do autômato.

Note-se que, neste caso, a descrição ou modelagem do AA é feita descrevendo-se o comportamento desejado do AA. Este comportamento pode ser composto por vários comportamentos mais primitivos e envolve o conhecimento prévio dos estados cognitivos do agente, não devendo ser confundido com o conceito de Comportamento Observável descrito na seção 2.3.2, definição 2.3.17.

Convém neste ponto ressaltar que existem duas formas de representação para os AEFs, as chamadas Máquinas de Mealy e Máquinas de Moore. Na Máquina de Mealy a saída é associada com cada transição de estados, enquanto na Máquina de Moore a saída está associada com o estado em que o autômato se encontra. Não é difícil demonstrar que as duas formas de representação são equivalentes e que pode-se transformar um autômato representado por uma Máquina de Mealy em outro representado por uma Máquina de Moore e vice-versa.

Deste modo, de agora em diante, usaremos para descrição de comportamentos de agentes a forma que for mais conveniente para o fácil entendimento do comportamento que se deseja descrever.

Comportamentos em que os valores dos atuadores dependam apenas do valor dos sensores, como as taxias e os reflexos, podem ser descritos por autômatos na forma de uma Máquina de Mealy de apenas um estado como mostrado no exemplo 3.4.1. Quando os valores dos atuadores dependerem, não apenas dos valores dos sensores mas também de alguma espécie de 'memória', similar aos estados internos que refletem a história passada do agente, como nos casos dos comportamentos reativos e instintivos, autômatos com vários estados devem ser utilizados para descrever estes agentes.

Outra característica bastante interessante desta ferramenta é que ela permite a descrição dos comportamentos em vários níveis de detalhamento. Em um nível mais

elevado, cada estado do autômato pode representar um comportamento geral de determinada ação de um comportamento reativo, como por exemplo, caçar, acasalar, comer, etc. Cada um destes comportamentos (estados do AEF) pode ser detalhado por outra máquina de estados finitos expandida, por exemplo, comer pode ser decomposto nas ações de morder, mastigar, engolir, etc. Da mesma forma, a ação de morder poderia ser decomposta em vários estados e assim sucessivamente até o nível de detalhamento desejado. A utilização de redes de AEFs permite que vários estados estejam ativos ao mesmo tempo.

Os exemplos a seguir mostram a descrição de vários comportamentos estereotipados encontrados na literatura, utilizando redes de autômatos de estados finitos. Estes exemplos ilustram várias classes de comportamentos e diferentes níveis de detalhamento.

#### Exemplo 3.4.1 Comportamento de taxa de mover-se em direção ao alimento

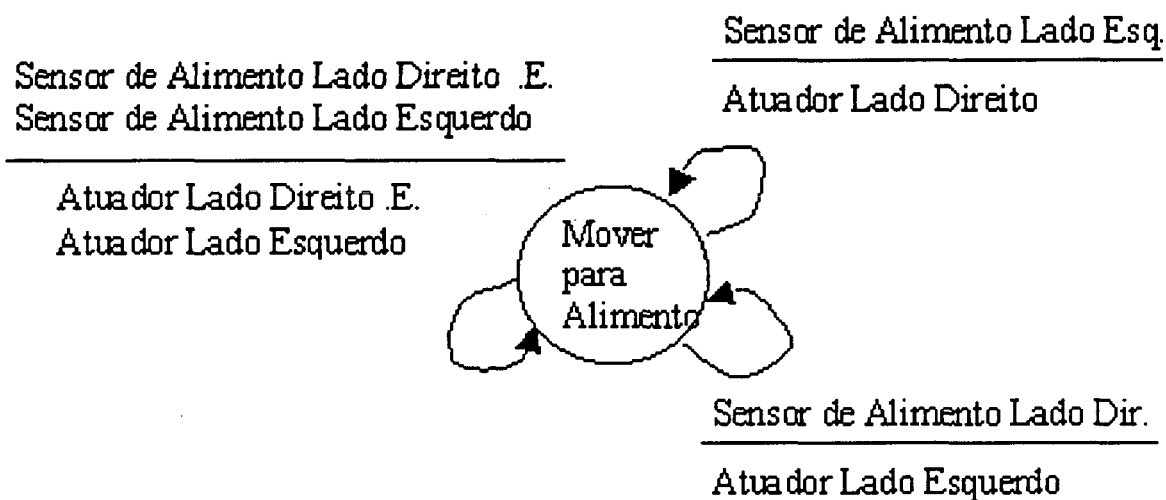


Figura 3.4: Máquina de Mealy representando o comportamento de mover-se em direção ao alimento

#### Exemplo 3.4.2 Comportamento reativo apresentado pela vespa.

#### Exemplo 3.4.3 Barata que procura a sombra e foge de ataque.



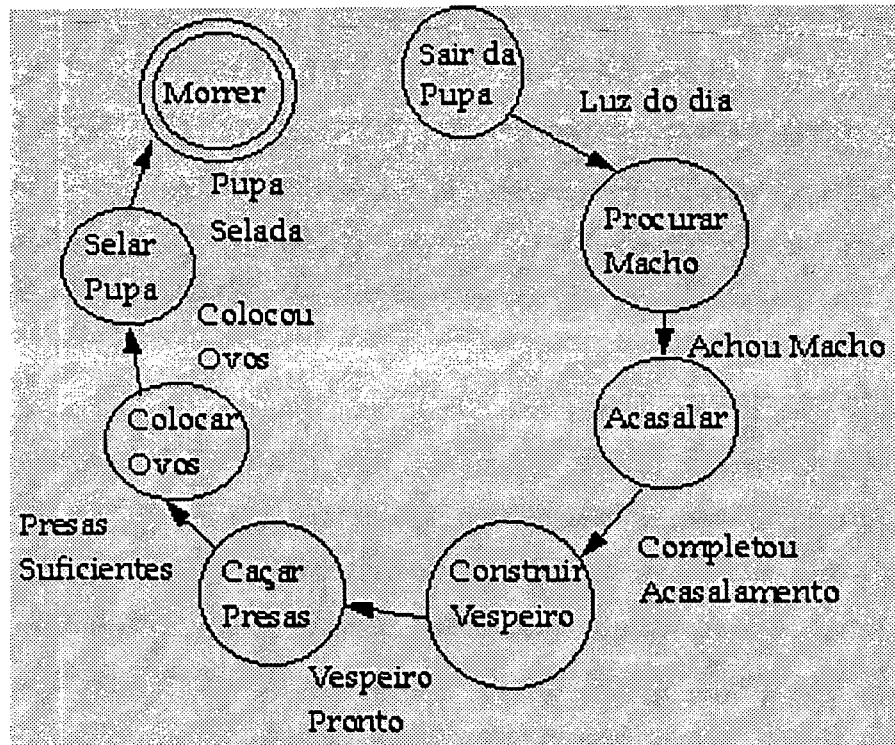


Figura 3.5: Autômato de Estados Finitos que descreve de modo geral o comportamento da vespa descrito no Exemplo 3.3.1

*Neste exemplo podemos ver como é possível descrever dois comportamentos básicos agindo sobre o mesmo grupo de atuadores. A transição entre um comportamento e outro ocorre quando um evento específico (no caso a presença ou ausência de um elemento 'perigoso') é detectado pelos sensores.*

#### Exemplo 3.4.4 Acasalamento

*Descrição de acasalamento movido por comportamento instintivo. Se o nível hormonal ultrapassar um determinado limiar, o comportamento é disparado, mesmo na ausência de um estímulo externo.*

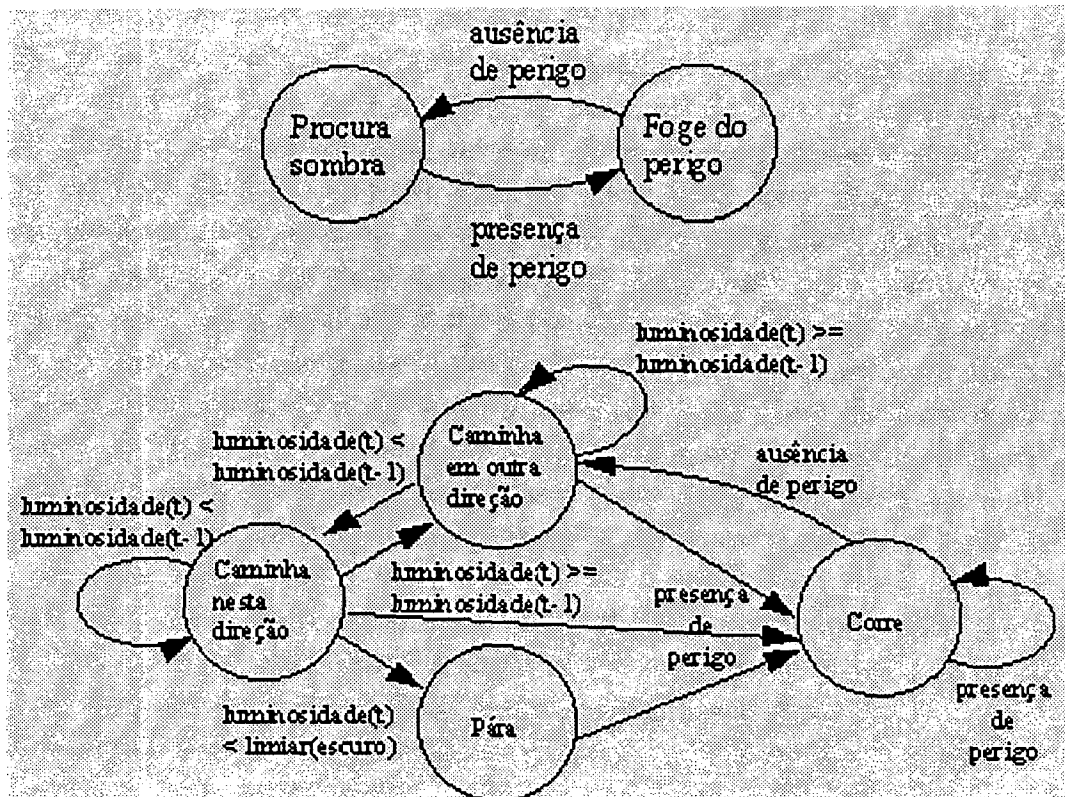


Figura 3.6: Dois Autômatos de Estados Finitos descrevendo em diferentes níveis de detalhamento o comportamento de uma barata que procura por um ambiente escuro e ao mesmo tempo foge de perigo

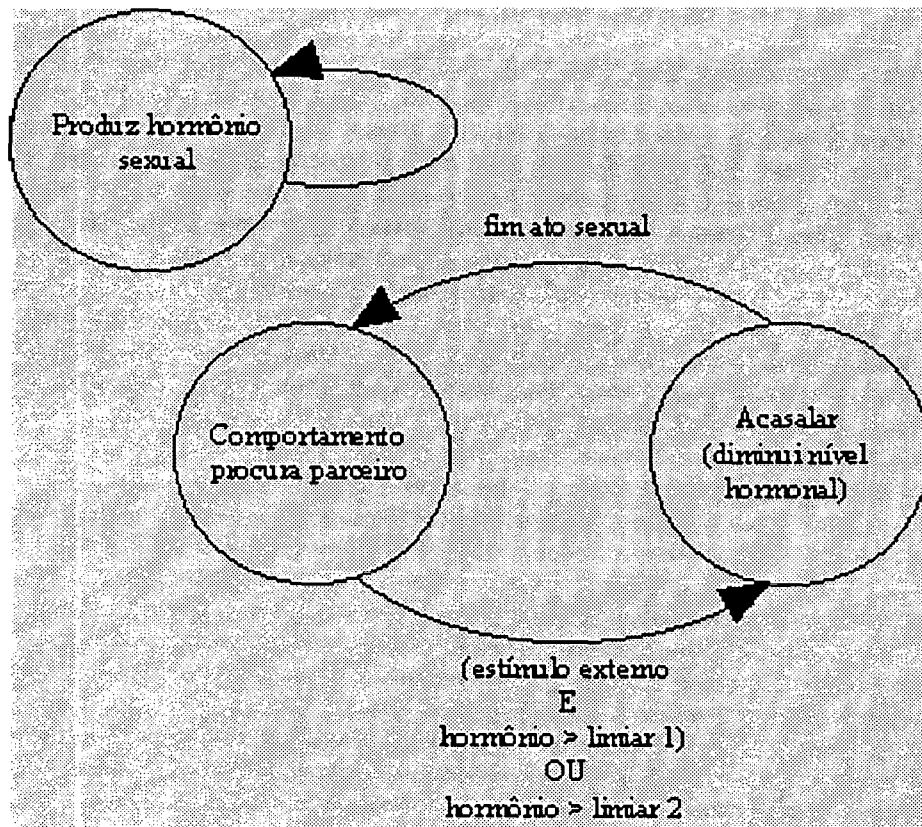


Figura 3.7: Autômato de Estados Finitos que descreve um acasalamento movido por comportamento instintivo

# Capítulo 4

## Aprendizado

### 4.1 Introdução

Trabalhos recentes na área de AAs procuram construir dispositivos e dotá-los de um conjunto de comportamentos que os tornem aptos a realizarem as tarefas a que foram projetados. Assim, viu-se no capítulo anterior as diferentes classes de comportamento encontradas na Natureza. No entanto, dotar um AA de um conjunto fixo de comportamentos rígidos não é suficiente para garantir a sua operação em um ambiente dinâmico como o mundo real. É necessário dotá-lo também da capacidade de selecionar, alterar e aprimorar seus comportamentos, tornando-o mais apto para operar no ambiente em que for colocado. A esta capacidade chamamos de “Aprendizado”.

O objetivo deste capítulo é procurar fazer uma revisão do conceito de aprendizado tanto do ponto de vista psicológico como do ponto de vista da Inteligência Artificial (IA). Este estudo do processo de aprendizado tem duas razões fundamentais. O primeiro é tentar entender o processo e, através da implementação de modelos, investigar como o processo funciona em animais e seres humanos. A segunda razão é procurar dotar os AAs com a capacidade de aprender. Desde os seus primórdios, tem sido objetivo da IA desenvolver sistemas que possam ser ensinados ao invés de programados [84].

Assim como o termo “Inteligência”, o termo “Aprendizado” é bastante difícil de ser definido de maneira clara e precisa. Muitas vezes, o próprio termo inteligência é confundido com aprendizado [66]. Dependendo da área específica de estudo, vamos encontrar diferentes definições, classificações, paradigmas e métodos de estudo.

Faz-se também uma breve especulação sobre as mudanças que se supõe que ocorram no sistema nervoso central por ocasião do aprendizado. Analisando-se as diferentes abordagens, pode-se selecionar que aspectos de cada uma são mais

interessantes para implementação em AAs. Finalmente, nos capítulos seguintes procura-se utilizar os diferentes tipos de aprendizado encontrados na Natureza para o aprendizado de comportamentos em AAs e como eles podem ser implementados em Redes Neurais Artificiais.

## 4.2 O que é Aprendizado?

Muitos pesquisadores têm procurado apresentar a conceituação de aprendizado. De acordo com o American Heritage Dictionary, aprendizado é “o ato de aprender, de ganhar conhecimento, compreensão ou domínio através da experiência ou estudo”. No entanto, muitas pessoas não aceitam com facilidade esta definição, pelo fato dos termos vagos que ela contém, tais como “conhecimento”, “compreensão” e “domínio”. Nos últimos anos muitos psicólogos têm aceito a definição de aprendizado “como um processo relacionado com mudanças de comportamento” [84] [81] [61] [124]. Uma das definições mais populares é a proposta por Kimble [96] que define aprendizado como sendo uma “mudança relativamente duradoura no comportamento presente e potencial, e que ocorre como resultado da estimulação sensorial (inclusive da estimulação do reforçamento)”.

Insistindo que a mudança de comportamento deve ser relativamente duradoura, eliminamos automaticamente as mudanças transitórias resultantes de adaptação sensorial, fadiga e flutuações [61].

Assim, chamamos de aprendizado ao processo pelo qual um animal muda adaptativamente seu comportamento, como resultado da experiência individual que adquire no decorrer da sua vida. E, aquilo que é aprendido, chamamos de memória, ou traço de memória ou engrama [124].

Se o aprendizado é o processo capaz de causar mudanças de comportamento observados nos animais, a maioria dos teóricos da área de aprendizado concordam também que o processo de aprendizado não pode ser estudado de maneira direta. A natureza do aprendizado só pode ser inferida através de observações nas modificações comportamentais. Skinner é um dos poucos pesquisadores que se opõe a esta idéia. Skinner, como um “behaviorista” puro acredita que o aprendizado é a própria mudança comportamental e que nenhum processo de inferência adicional é necessário [84].

## 4.3 Tipos de Aprendizado

Se a conceituação do termo aprendizado já é tarefa espinhosa, muito mais difícil é a análise dos diferentes tipos de aprendizado. Dependendo da área específica de estudo de determinado autor, os tipos de aprendizado apresentados variam enormemente.

Pode-se, a grosso modo, identificar duas correntes principais. A primeira envolve as pesquisas nas áreas de psicologia, etologia e educação, e a segunda envolve principalmente pesquisas na área de IA.

Os pesquisadores da primeira área estão preocupados em estudar o aprendizado como um processo de modificação de comportamento, seja de animais ou de seres humanos. Nesta abordagem, fundamentalmente, todo o processo de aprendizado é um processo de condicionamento, seja ele clássico ou instrumental. No entanto, ao nos limitarmos apenas nestes processos de condicionamento, excluimos grande parte do processo de aprendizado utilizado pelos seres humanos.

Já o conceito de aprendizado, quando apresentado sob a ótica da IA clássica ou simbólica, envolve primordialmente a formulação e refinamento de conceitos, incluindo a generalização ou especialização de um modelo conceitual através de manipulações a nível simbólico. Esta abordagem está de acordo com os princípios da IA simbólica, que desde os seus primórdios procura simular em computadores o comportamento inteligente apresentado por seres humanos. No entanto, como os seres humanos são os únicos animais capazes de articular regras e de raciocinar usando símbolos [99], esta abordagem também é, de certo modo, restrita.

Ao longo desta seção serão apresentados de maneira sucinta os diferentes tipos de aprendizado apresentados pelas duas principais correntes de pensamento. Ao final, procuraremos fazer uma síntese a fim de adaptar estas conceitos aos objetivos deste trabalho, qual seja, o de dotar AAs da capacidade de aprendizado.

### 4.3.1 Aprendizado em Sistemas de IA Clássica

Do ponto de vista da IA clássica, técnicas de aprendizado, ou mais especificamente, processo de aprendizado pela máquina pode ser encarado como:

*“O desenvolvimento de sistemas capazes de aprender envolve o desenvolvimento ou identificação e implementação de algoritmos (algoritmos de aprendizado) que habilitem outros algoritmos (algoritmos de IA) a melhorar sua performance ou se adaptar. [44]*

Existem muitas abordagens de aprendizado pela máquina, sendo que muitas delas foram desenvolvidas motivadas pela necessidade de aquisição de conhecimentos por sistemas especialistas. Algumas destas abordagens são descritas a seguir: [165] [49]

### **Aprendizado por Memorização**

É a situação mais simples de aprendizado, na qual o ambiente, um especialista por exemplo, fornece o conhecimento em uma forma que pode ser utilizada diretamente pelo sistema. O sistema de aprendizado não necessita fazer qualquer processamento para entender ou interpretar a informação fornecida pelo ambiente. Tudo que ele necessita é memorizar a informação para poder recuperá-la posteriormente. É o equivalente à memorização de fatos e ações, como por exemplo, quando se “decora a tabuada” ao aprendermos a multiplicação. Neste contexto, o oposto de aprendizado é esquecimento.

### **Aprendizado por Instrução**

Nesta abordagem, o módulo de inferência do sistema é modificado através do fornecimento de informações de controle, tais como heurísticas. Em um exemplo simples, o aprendizado se dá basicamente pela criação de regras adicionais obtidas através de instruções do tipo “SE . . . ENTÃO ...”. Deve-se considerar também a possibilidade de serem fornecidas instruções que entrem em conflito com informações já existentes na base de conhecimentos.

### **Aprendizado por Exemplos**

Uma forma de ensinar um sistema a realizar uma tarefa é apresentar ao sistema exemplos de como ele deve se comportar. O sistema deve então generalizar estes exemplos para achar regras de nível mais alto que possam ser aplicadas na operação da máquina de inferência. Os exemplos podem ser vistos como sendo peças de um conhecimento específico e que, no entanto, não pode ser utilizada diretamente pela máquina de inferência. Eles, então, devem ser transformados em regras a fim de poderem ser utilizados eficientemente pelo sistema. O aprendizado por exemplos é uma das formas mais populares de aprendizado utilizado em sistemas conexionistas e será visto em detalhes adiante.

### **Aprendizado por Exploração**

É a abordagem em que o aprendizado se dá através da utilização de mecanismos de busca pelo sistema. Esta é a abordagem de aprendizado em que a Natureza fornece as melhores fontes de inspiração. Nesta abordagem estão incluídos os Algoritmos Evolucionários (AEs) e o Aprendizado por Reforço (“Reinforcement Learning”).

### **Aprendizado por Analogia**

Podemos ver que, uma vez que determinado problema tenha sido resolvido, o mesmo processo de solução pode ser utilizado em outras situações similares. Assim, caso um sistema tenha acesso a uma base de conhecimentos para a realização de uma tarefa similar ou relacionada, ele pode melhorar sua própria performance através do reconhecimento de analogias e da transferência de informações relevantes desta primeira base de conhecimentos.

### **4.3.2 Aprendizado em Sistemas Naturais**

Neste ponto, considera-se o sistema classificatório usualmente aceito por etologista e psicólogos. Apesar de dividido em classes de aprendizado bem distintas, é muito difícil distinguir exatamente quando uma e não outra forma de aprendizado está ocorrendo, ou mesmo isolar uma da outra. Deve-se supor que na Natureza estas ocorrem de maneira contínua e que o mecanismo biológico que ocorre no sistema nervoso central por trás do aprendizado deve possuir uma base comum.

#### **Impressão**

É uma forma altamente especializada e limitada de aprendizado. A impressão pode ser vista muito claramente em aves, durante o período inicial da sua vida, após a eclosão dos ovos. Em uma ave jovem, ela consiste simplesmente em aprender a seguir o primeiro objeto grande e móvel que a ave vê e ouve, em um modo reminiscente da tendência natural das aves de seguirem sua mãe. Por exemplo, se um pato nasce na presença de uma caixa verde grande, contendo um despertador em funcionamento, ele seguirá o movimento da caixa ao longo de um fio que a transporte. Depois de certa exposição à caixa, o pato a seguirá mais do que a sua própria mãe ou a outras aves. Esse aprendizado deve depender de alguma condição especial do sistema nervoso que prevalece apenas cedo no desenvolvimento; pois, se a ave não for impressionada logo após a eclosão do ovo, ela dificilmente o será mais tarde. [61].

#### **Habituação**

Talvez o tipo mais simples de aprendizado visto durante a vida toda seja a habituação. Neste tipo de aprendizado, através de exposições repetidas a um estímulo, um animal diminui gradualmente a sua resposta natural, até que ela possa desaparecer inteiramente. Assim, um pássaro pode aprender a não levantar vôo ao avistar folhas levadas pelo vento vindo em sua direção. Esta mudança é considerada uma forma de aprendizado, pois a diminuição da resposta cresce com exposições diárias ao



estímulo e dura através de longos períodos de tempo sem estimulação. Em um certo sentido, a habituação representa a eliminação de respostas que não tenham “significado” na vida do animal. A maioria do resto do aprendizado relaciona-se com a intensificação das respostas que são significativas, assim, elas podem ser despertadas mais prontamente e com frequência ou probabilidade aumentadas [61] [187].

### Condicionamento Clássico

Outra forma simples de aprendizado é o condicionamento clássico, assim chamado porque foi descoberto por I. P. Pavlov, o pai do condicionamento. Em sua experiência clássica, Pavlov prendeu levemente um cão com um cabresto, soprou pó de carne repetidamente na sua boca e registrou precisamente a quantidade de saliva produzida. Então ele associou o som de uma campainha com o pó de carne, repetindo este procedimento muitas vezes, a intervalos sucessivos. A campainha não provocou salivação no início, naturalmente; mas, após pareamentos repetidos com carne, isto aconteceu. Ao descrever esta experiência, Pavlov chamou a salivação sob a influência da campainha de reflexo condicionado (RC); a campainha, de estímulo condicionado (EC); a salivação sob a influência da carne, de reflexo não-condicionado (RNC); e a própria carne de estímulo não-condicionado (ENC). A mesma experiência foi repetida muitas vezes com animais diferentes e com muitos estímulos e respostas diferentes. Tipicamente, o RC é semelhante ao RNC, mas nunca completamente idêntico a ele. Assim, a melhor forma para descrever o condicionamento clássico seria como um processo no qual um estímulo previamente neutro (EC=campainha) é capacitado a despertar uma resposta (RC=salivação) que não é nunca despertada antes do treinamento [61]. A Figura 4.1 mostra o processo de condicionamento clássico em um conjunto de três neurônios idealizados.

### Condicionamento Instrumental

Ao publicar um de seus mais importantes trabalhos, *The Behavior of Organisms*, em 1938, Skinner utilizou como base para suas teorias um tipo de aprendizado que ele chamou de Condicionamento Operante, também chamado por Hebb [81] de Condicionamento do Tipo R e por Thorpe [180] de Condicionamento Tipo II ou Condicionamento Instrumental, em oposição ao condicionamento clássico utilizado por Pavlov e que Hebb chamou de Condicionamento Tipo E e Thorpe de Condicionamento Tipo I. No condicionamento Tipo II a resposta condicionada é uma ação

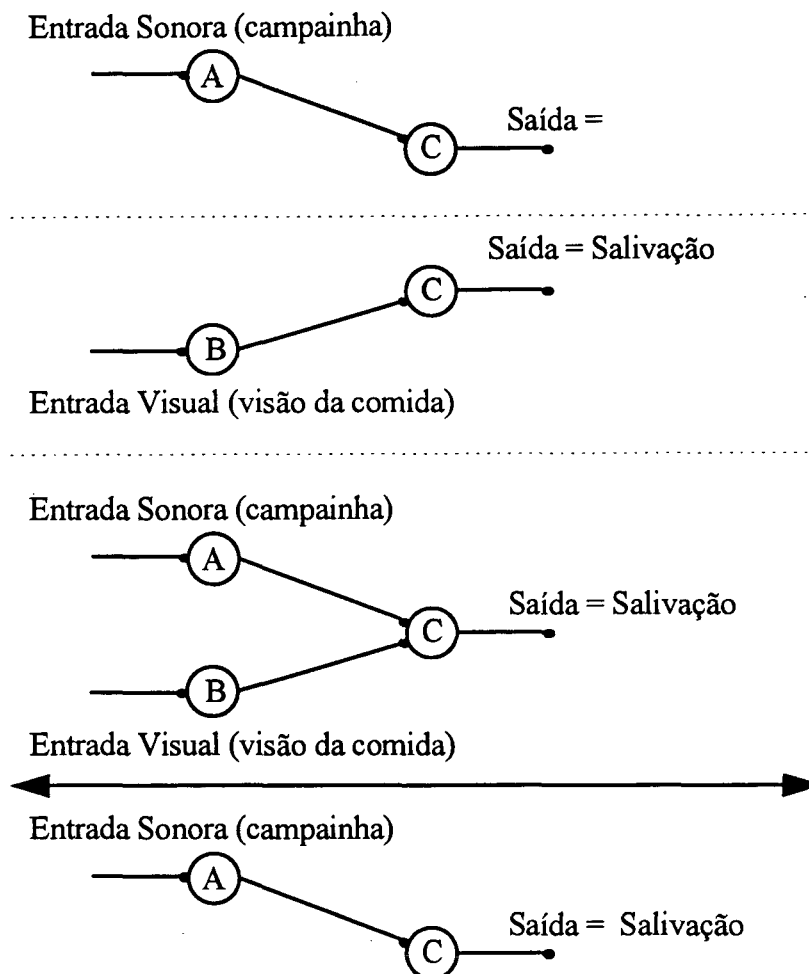


Figura 4.1: Exemplo de uma seqüência de condicionamento clássico

voluntária do animal e faz parte do seu repertório normal de comportamentos, além disso, tanto o estímulo quanto a resposta devem preceder a recompensa [187].

Assim, no condicionamento instrumental, o animal inicia timidamente, emitindo uma variedade de respostas que estão no seu repertório natural. O treinador pode então selecionar uma resposta para reforçar: abaixar uma alavanca, ficar em pé nas patas traseiras, virar para a direita e não para a esquerda ao fim de um corredor, e assim por diante. Devido ao reforço, esta resposta é emitida com maior frequência e outras respostas são eliminadas ou tornam-se hábito [61].

A Figura 4.2 mostra de maneira informal o ambiente de condicionamento instrumental onde a resposta a reforçar é o pressionar de uma alavanca.

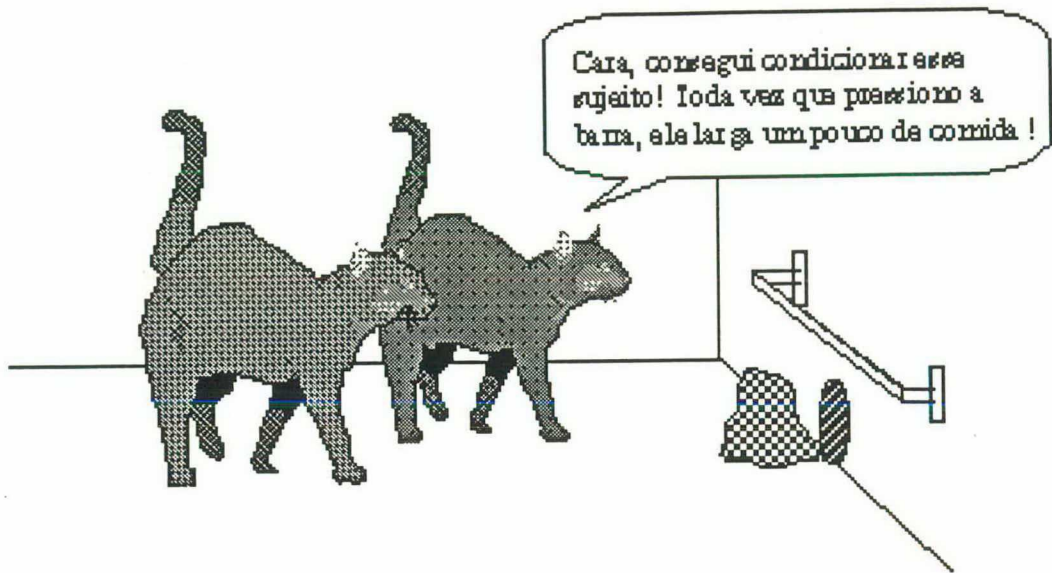


Figura 4.2: Gatos na caixa de Skinner; um deles a ponto de calcar a barra

### Aprendizado por tentativa-e-erro

Muitos pesquisadores concordam com o fato de que nem o condicionamento clássico nem o instrumental ocorrem de forma isolada na Natureza. Thorpe [180] considera que em situações reais, ambas as formas de aprendizado ocorrem simultaneamente, sendo que o condicionamento instrumental parece ser predominante, num processo que ele chamou de aprendizado por tentativa-e-erro. Outros pesquisadores consideram o aprendizado por tentativa-e-erro apenas uma situação mais complexa do condicionamento instrumental [187].

Para Dethier [61], por exemplo, o condicionamento instrumental pode ser complicado simplesmente pelo aumento da complexidade da situação do estímulo ou das possibilidades de resposta. Isto pode ser feito dando-se a um animal uma escolha de estímulos para responder. Por exemplo, o animal pode ser confrontado com duas portas, uma clara e outra escura. O alimento está disponível atrás da porta clara; a porta escura está trancada e o animal precisa aprender a se aproximar, ou saltar para a porta clara, esteja ela situada à direita ou à esquerda.

Ainda mais complicado é o labirinto de escolha múltipla, no qual pode-se exigir que o animal faça uma cadeia de discriminações, a fim de traçar seu caminho através do labirinto até o alimento no final, como pode ser visto na Figura 4.3, extraída de [61].

Neste tipo de situação, o animal está mais obviamente envolvido em um processo

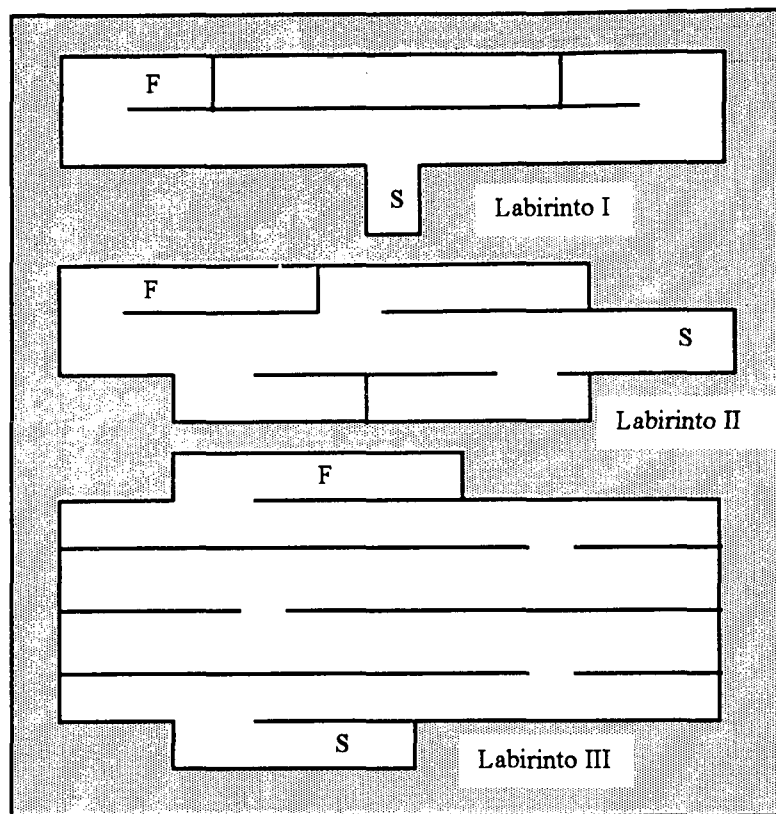


Figura 4.3: Três labirintos de dificuldade crescente. S é o local de partida e F o local de alimento

de aprendizado por tentativa-e-erro, no qual respostas corretas são encorajadas por recompensa e respostas erradas desencorajadas pela não-recompensa ou, em alguns casos, por punição [61].

#### 4.4 Aprendizado e Sistema Nervoso Central

Na busca por inspirações biológicas para a implementação de AAs, uma possível abordagem para a solução do problema do aprendizado seria, ao invés de estudá-lo de maneira indireta através da observação do comportamento do indivíduo, estudá-lo diretamente no local em que ele ocorre, ou seja, no Sistema Nervoso Central (SNC). Vê-se, no entanto, que ainda hoje em dia, este processo está ainda no terreno das hipóteses e que muita pesquisa ainda necessita ser feita.

Quando o aprendizado ocorre, ele teoricamente provoca mudanças no SNC. Devem existir então, registros relativamente permanentes do que foi aprendido. Estes registros, que chamamos de memória ou engrama, podem ser, eventualmente, “cha-

mados” de maneira consciente quando necessita-se deles para executar uma tarefa, ou ainda simplesmente “aparecerem” quando alguma experiência dispara “incoscientemente” a busca por estes registros de memória.

A procura pelos engramas deve envolver, então, uma abordagem celular. No entanto não está claro ainda se e como o aprendizado altera a morfologia dos neurônios. Permanecem questões do tipo: Os neurônios se parecem diferentes após o aprendizado? Novas sinapses ou conexões são formadas? O processo de aprendizado altera a permeabilidade das membranas dos neurônios? Os neuro-transmissores se alteram?

Enquanto não se conhece exatamente como o cérebro funciona, várias hipóteses têm sido propostas. Wallace [187] apresenta uma hipótese extraída de Manning [117] que descreve como um engrama seria formado:

“Primeiramente os órgãos sensoriais registram os eventos que ocorrem no ambiente. Em seguida, as informações são encaminhadas para partes específicas do córtex e, ao mesmo tempo, para os sistemas reticular e límbico. Caso os eventos possuam alguma importância, o sistema reticular ativa certos centros nervosos e o engrama começa a se formar no córtex. O sistema límbico registra a “significância” dos eventos registrando seus resultados - eles eventualmente reduziram a sede, a necessidade sexual, a fome? Deve-se ressaltar que os centros de prazer estão associados com o hipotálamo e o sistema límbico, no entanto, estimulações de diferentes áreas deste mesmo sistema podem produzir comportamento de ódio ou rancor. O sistema límbico, então, passa sinais para o hipotálamo que, como sabemos, possui a função de regulação das necessidades corporais. O resultado de todo este processo é a formação de engramas no córtex cerebral, que então, reage como o que poderia ser chamado genericamente de canais motores “de abordagem” ou “de movimento”. Deste modo, o engrama opera influenciando como o animal se comportará na próxima ocasião em que for apresentado para estes eventos ambientais em particular.”

Outras teorias propõem que a memória se constitui de circuitos reverberantes que, uma vez ativados através da experiência sensorial, permaneceriam por longos períodos de tempo. Críticas a esta teoria argumentam que os circuitos reverberantes necessitariam de muita energia, o que inviabilizaria a sua manutenção por períodos muito longos de tempo. Uma hipótese é a de que estes circuitos funcionariam como uma memória de curta duração, enquanto que a memória de longa duração deve ser uma estrutura estável, porém difusa, do SNC. Existem ainda autores que sustentam que poderiam ocorrer alterações morfológicas nas células nervosas associadas com o aprendizado, como por exemplo, alterações no número de microsomias (pequenos corpos encontrados no interior dos neurônios); alterações no tamanho e formato dos dendritos apicais; e, ainda, alterações nas sinapses propriamente ditas.

Hipóteses bioquímicas também foram levantadas, baseadas em alguns experimentos com planárias, alguns pesquisadores sugeriram que o ARN (Ácido Ribonuclêico) poderia ser uma espécie de “memória química”. Estes experimentos, no entanto, foram bastante criticados e nenhuma conclusão definitiva foi estabelecida [187].

Neste ponto, chega-se a conclusão que ainda não existe uma evidência definitiva que explique como o aprendizado altera o SNC. Assim, ao longo deste trabalho, ao utilizarmos diferentes algoritmos de aprendizado para RNAs, não podemos afirmar especificamente o grau de inspiração biológica do algoritmo.

## 4.5 Aprendizado no Contexto deste Trabalho

Para os objetivos deste trabalho, pode-se adotar em grande parte a definição de aprendizado derivada da etologia, e dizer que o aprendizado é um processo que procura adaptar ou otimizar o comportamento observável de um AA, tornando-o mais apto a operar no ambiente em que for colocado.

O tipo ou tipos de aprendizado utilizados para se alcançar esta melhoria de comportamento podem ser os mais diversos possíveis e dependem em grande parte de como é implementada a entidade cognitiva responsável pela geração do comportamento do AA.

Como já foi dito anteriormente, a implementação de AAs utilizando as técnicas de IA simbólica com seus respectivos tipos de aprendizado não têm obtido sucesso para operação no mundo real.

Ao procurar uma arquitetura mais conveniente para a implementação de AAs capazes de operar no mundo real, a abordagem conexionista, através de modelos de Redes Neurais Artificiais (RNAs), deve ser considerada como uma alternativa a ser pesquisada, entre outros motivos, pela sua grande inspiração biológica. Neste caso, então, deve-se estudar os tipos de aprendizado adequados e compatíveis com este modelo.

Mais adiante neste trabalho, no capítulo referente às RNAs, aborda-se com detalhes os algoritmos de aprendizado empregados nos diferentes modelos de redes. Pode-se adiantar, no entanto, que o aprendizado por exemplos é um dos tipos de aprendizado mais empregados. Outras formas de aprendizado empregadas neste tipo de sistema são o condicionamento clássico, o condicionamento instrumental, e o aprendizado por tentativa-e-erro.

# Capítulo 5

## Evolução

### 5.1 Introdução

Como já foi mencionado anteriormente, o desenvolvimento de AAs objetiva a construção de sistemas, que assim como os animais, sejam capazes de interagir e operar em um ambiente dinâmico externo. Ao observar-se a Natureza, tenta-se entender como ela resolveu o problema, que mecanismos foram utilizados para criar criaturas cada vez mais complexas e mais adaptadas a sobreviverem em um determinado ambiente, que conjunto de comportamentos estas criaturas possuem, como eles foram criados, como são modificados e como estes comportamentos são passados para os descendentes.

Estes mecanismos são a *Evolução* e a *Seleção Natural*. A *Evolução* é o mecanismo pelo qual se obtém a diversidade de organismos biológicos que interagem com o meio ambiente. O meio ambiente é uma arena finita, conseqüentemente, existe uma competição pelos recursos disponíveis. A *Seleção Natural* então, é o mecanismo que entra em ação quando os organismos criados tendem a se reproduzir e ocupar todo o espaço de recursos disponíveis. Aquelas criaturas que apresentam os comportamentos menos eficientes ou que não são capazes de alterar seu comportamento caso o ambiente se modifique, tendem a ser eliminadas nesta luta pela sobrevivência. Foi assim que, através de gerações, os organismos aumentaram a sua chance de sobrevivência através do aumento do seu repertório de comportamentos e da sua capacidade de adaptação.

A Natureza deve possuir também um mecanismo pelo qual as criaturas mais aptas sejam capazes de transmitir aos seus descendentes os comportamentos exibidos por elas. Este mecanismo é a transmissão de um código genético através da reprodução.

Se a inteligência é a capacidade de interligar de maneira coerente a percepção

com a mobilidade, e se a Natureza, através dos processos descritos acima, foi capaz de fazer emergir esta capacidade, então, pode-se especular que, se formos capazes de simular estes processos em um computador, poderemos criar AAs inteligentes, capazes de se comportarem de maneira adequada para atingir seus objetivos no ambiente em que forem colocados para operar.

Assim, este capítulo trata da questão da evolução, descrevendo inicialmente os mecanismos naturais da evolução e seleção natural, apresentando hipóteses de como este processo levou ao desenvolvimento do sistema nervoso e a evolução dos comportamentos. Finalmente, neste capítulo são apresentados os principais métodos computacionais inspirados nos mecanismos biológicos de evolução e como eles podem ser utilizados para gerar diferentes arquiteturas de Redes Neurais Artificiais.

## 5.2 Bases Biológicas da Evolução

### 5.2.1 Origens da Vida

É bastante difícil falar das origens da vida. Esta sempre foi uma curiosidade do homem que, continuamente, vem apresentando diferentes teorias e dogmas. Barreto [19] apresenta uma lista de referência onde estes diferentes pontos de vista podem ser encontrados. Neste trabalho, apresenta-se as idéias de Churchland [47], por uma questão de convicção pessoal e por esta se adequar a linha de raciocínio evolucionista do restante deste capítulo.

Próximo da superfície do oceano, entre 3 e 4 bilhões de anos atrás, os processos de reações químicas, ativadas pela luz solar e pelas descargas elétricas atmosféricas, produziram algumas estruturas moleculares que podiam se auto-replicar. Estas moléculas complexas eram capazes de retirar material de outras moléculas e elementos químicos do ambiente que as cercavam e catalizar seqüências de reações que produziam cópias exatas delas próprias. Entre estas moléculas auto-replicas, a vantagem competitiva ia para aquelas cuja estrutura molecular induzia não apenas à sua própria duplicação, mas que possuíssem uma estrutura que as protegesse do “ataque químico” do meio exterior e que possuíssem mecanismos para formar moléculas necessárias a sua duplicação através de manipulação química das moléculas do ambiente incapazes de serem utilizadas diretamente.

A célula é o resultado final desta competição entre moléculas auto-replicas. Ela possui uma membrana externa que protege as complicadas estruturas internas, e processos metabólicos complexos que transformam o material externo através das estruturas internas. No centro desta intrincada estrutura está cuidadosamente codificada uma molécula de ADN (Ácido Desoxi-Ribonucléico), o diretor da atividade



celular e o vencedor da competição descrita acima. Com a emergência da célula, temos o conceito padrão de vida: um sistema consumidor de energia que se auto-mantém e se auto-duplica [47].

## 5.2.2 Evolução Biológica

Uma revolução no pensamento biológico e até mesmo na filosofia humana começou quanto Charles Darwin e Alfred Russel apresentaram suas evidências para a teoria da evolução perante a “Linnean Society of London” em primeiro de julho de 1858. A teoria completa de Darwin, repleta de argumentos e observações pessoais foi publicada em 24 de Novembro de 1859 no seu livro “On the Origin of Species” [56] [119] e que hoje em dia pode ser obtido eletronicamente através da Internet [83]. A teoria Darwiniana clássica da evolução, combinada com o selecionismo de Weismann e a genética de Mendel, se tornaram universalmente aceitos como argumentos do paradigma conhecido como neo-Darwinismo [66].

O neo-Darwinismo defende a idéia de que a história de todas as criaturas vivas pode ser explicada através de alguns processos estatísticos atuando sobre populações e espécies. Estes processos são: *reprodução*, *mutação*, *competição* e *seleção*. *Reprodução* é uma propriedade óbvia de todas as formas de vida. Do mesmo modo, a *mutação* está garantida em qualquer sistema que continuamente se reproduza em um universo de entropia crescente. *Competição* e *Seleção* se tornam conseqüências inevitáveis para qualquer população que se expanda em um ambiente de recursos finitos. Evolução então é o resultado da ação destes processos estocásticos fundamentais a medida em que eles atuam sobre as populações, geração após geração [66].

Supondo-se válida a teoria da evolução biológica, é interessante ressaltar que a evolução em si é um processo sem um objetivo específico. Em outras palavras, não existe evidência de que a finalidade da evolução seja produzir a humanidade ou qualquer outra espécie. No entanto, uma tendência geral da evolução é a geração de organismos com grau crescente de complexidade [66]. A evolução é o resultado da diversidade de organismos biológicos que interagem com o meio ambiente, retirando deste meio as suas necessidades de vida e, por sua vez, modificando-o. Como os organismos são diferentes, alguns encontram condições de vida e possibilidade de reprodução maiores que os outros e, assim, seu número tende a aumentar pelo processo chamado seleção.

O mecanismo de seleção pode ser considerado como uma busca constante de maior eficiência.

Os mecanismos básicos pelos quais a seleção atua permitindo a evolução são:

parcimônia, diversidade e morte [19].

- *Parcimônia* significa que o organismo tende a se organizar de modo a minimizar o esforço para viver.
- *Diversidade* significa que a Natureza deve possuir um mecanismo de exploração de possibilidades na busca pela geração de organismos mais parcimoniosos. Este mecanismo é o da variação de fenótipos através de alterações nos genótipos dos organismos. Estas alterações se dão através de reproduções sexuais e mutações.
- *Morte* para dar lugar a organismos mais evoluídos - sem morte não há evolução. Na Natureza, morte não é o fim do ser, mas tal como uma regra de produção, é a substituição de um elemento por seu descendente, mantendo vivo o elo da informação genética.

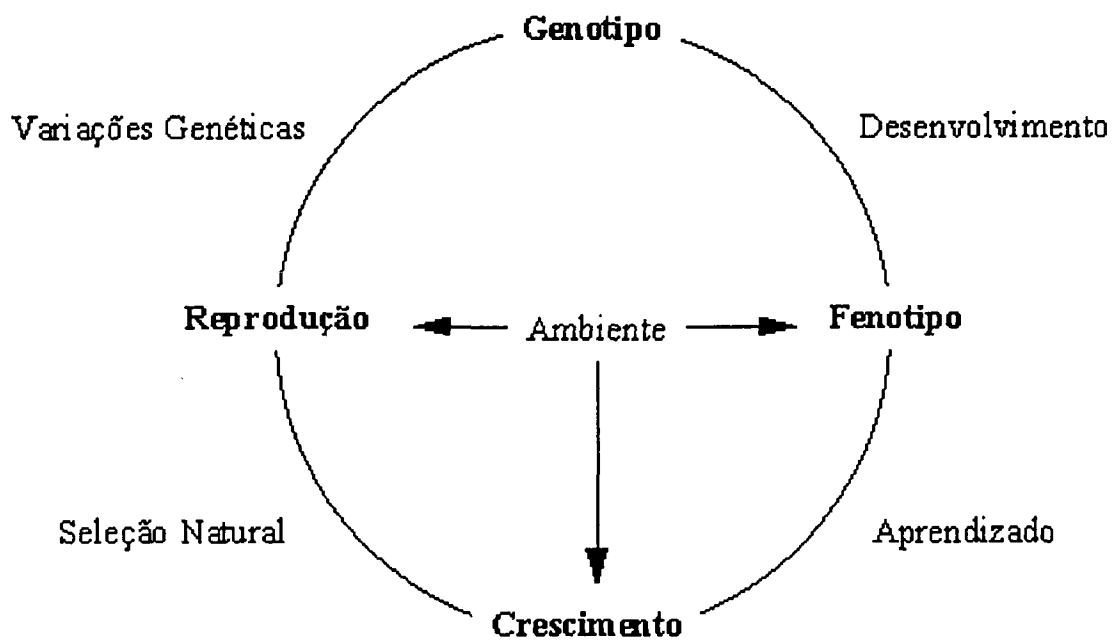


Figura 5.1: Ciclo de vida do sistema nervoso apresentando os mecanismos de desenvolvimento, aprendizado, seleção natural e variação genética

O processo, como um todo, é extremamente complexo e envolve muitos fatores. A aptidão de um organismo de se adaptar ao ambiente não é sempre fixa, pois na sua luta pela sobrevivência, o organismo interage com o próprio ambiente alterando-o.

Ao alterá-lo, altera-se também o grau de aptidão do organismo. Como o elemento de principal interação do organismo com o ambiente é o seu comportamento, e este por sua vez é determinado tanto por fatores genéticos como pelo grau de inteligência do organismo que lhe permite o aprendizado de novos comportamentos, podemos dizer que, a capacidade de aprendizado também é um fator que influencia o processo da evolução. Este efeito é chamado de efeito Baldwin [3] [7] e também foi defendido por pesquisadores como Schrödinger [166] e Piaget [140] dos quais Vaario [184] extraiu alguns argumentos:

- *“Individuals which - by chance or intelligence - change their behavior accordingly will be more favoured, and thus selected. This change of behavior is not transmitted to the next generation by the genom, not by direct inheritance, but that does not mean that it is not transmitted”. [166]*
- *“...evolutionary transformation of adaptive significance (not, therefore, just any mutation) are closely bound up with new patterns of behavior.” [140]*

### 5.2.3 Fundamentos de Genética

Toda informação genética é armazenada em genes que estão contidos em uma longa cadeia na forma de uma dupla-escada helicoidal, chamado cromossomo. Este é constituído por ADN (Ácido Desóxi-Ribonucléico), do qual cada célula em um organismo possui uma cópia (apesar de que existem formas de vida que contém apenas ARN). Esta informação genética é codificada com quatro diferentes bases: adenina, guanina, timina e citosina. Uma outra cadeia, chamada ARN (Ácido Ribonucléico), possui uma cópia invertida de parte das informações contidas no ADN. O ARN é constituído das mesmas bases que o ADN, com exceção da timina que é substituída pela uracila. Cada conjunto de 3 bases no ARN forma um código para a produção de 1 de um total de 20 amino-ácidos, ou um marcador. O ARN é traduzido em proteínas através de uma estrutura celular chamada ribossomo, que lê o ARN e conecta os amino-ácidos na ordem por ele codificada. Os marcadores indicam onde iniciar e onde terminar de ler a cadeia de ARN.

Os amino-ácidos são os blocos constituintes de todas as proteínas. A maioria das proteínas são enzimas que catalizam diferentes reações químicas nas células. Cada proteína é constituída por um número bastante grande de amino-ácidos reunidos em uma ordem determinada. É esta ordem que determina a sua forma e é através da forma que a proteína age. Para cada proteína, das quais existem cerca de 30.000 no ser humano, a ordenação dos amino-ácidos está codificada no ADN, onde cada

proteína é codificada por um gene. Deste modo, é o ADN que determina que tipo de proteínas são construídas e, em consequência, como cada célula vai operar [30].

### 5.2.4 Genótipo e Fenótipo

O código genético não descreve qual será a forma final de um organismo, mas sim, codifica um certo número de regras que, quando seguidas, resultarão na forma final. Ou seja, ao invés de um diagrama ou uma planta de como será o organismo, os genes dos cromossomos equivalem a uma receita e esta receita é chamada de *genótipo*. O processo de crescimento e manutenção de um organismo resulta na “execução” desta receita pelas células e na reprodução destas. Assim, o material genético vai produzir um *fenótipo*, resultado do genótipo interagindo com o ambiente. Isto significa que não existe uma correspondência de um para um entre o que está codificado no gene e o que será o organismo final.

Assim, num processo evolucionário, não são os organismos finais, mas as receitas que os construíram é que são combinadas para evolução. Apenas as regras (proteínas) que tem a capacidade de funcionarem conjuntamente para a produção de um novo organismo irão sobreviver.

Este processo evolucionário não leva apenas a mistura de receitas, mas faz com que certas receitas sejam executadas mais vezes, ou seja, receitas que deram certo podem ser levadas a serem repetidas mais vezes durante o processo de crescimento de novos organismos.

Provavelmente foi isto que ocorreu no caso do cérebro. O cérebro humano é o resultado de um processo evolucionário durante o qual o cérebro dos nossos ancestrais foi se tornando maior. Se o cérebro humano for comparado com o cérebro de um gato ou de um macaco, a maior diferença está no tamanho, que no caso dos humanos é significativamente maior, em particular os hemisférios cerebrais, que são responsáveis pelas funções cognitivas. A idéia então, é que, o processo evolucionário fez com que a receita que determina como um cérebro deve ser feito, seja executada mais vezes no processo embrionário de um homem, do que de um gato [30].

## 5.3 Evolução do Sistema Nervoso

Um dos fatores fundamentais para a evolução dos seres vivos foi o surgimento de estruturas especializadas no mapeamento dos sinais recebidos pelos sistemas sensoriais em ações para o sistema motor. Estas estruturas resultaram no que conhecemos hoje como sistema nervoso. Foi a evolução do sistema nervoso que permitiu o surgimento de seres mais aptos na luta pela sobrevivência. Todas as etapas deste processo

de desenvolvimento e especialização ainda não são perfeitamente conhecidas, porém Churchland [47] apresenta um episódio hipotético extremamente didático e revelador e que parece extremamente apropriado de ser aqui reproduzido.

*“Para apreciar quão facilmente um sistema nervoso pode vir a determinar aptidões de sobrevivência de toda uma espécie, consideremos uma criatura parecida com um caramujo que habitasse o fundo dos oceanos primitivos, há muitos milhões de anos. Esta espécie deveria sair da sua concha para se alimentar e recolher-se a ela apenas para descansar ou quando algum corpo externo fizesse contato direto com ela, como por exemplo, quando um predador atacasse. Muitas destas criaturas eram devoradas por predadores, pois apesar do comportamento reflexo de se recolher à sua concha, muitas eram mortas no primeiro ataque. Deste modo, as populações de diferentes espécies eram estáveis, estando em equilíbrio com a população de predadores.*

*Enquanto isto acontecia, todos os caramujos desta espécie possuíam uma faixa com células sensíveis à luz no topo de suas cabeças. Não existe aí nenhum fato notável. Muitos tipos de células são sensíveis à luz em diferentes graus. A presença destas células era uma característica completamente indiferente à espécie pois não tinham nenhuma função. Suponhamos agora que, devido a uma mutação no código genético de um determinado caramujo, células nervosas ligando a superfície da pele aos músculos que o faziam recolher-se à sua concha tenham crescido exageradamente, eventualmente conectando as células sensíveis à luz com estes músculos. Assim, mudanças bruscas na iluminação fariam com que o caramujo se recolhesse imediatamente à sua concha.*

*Esta nova característica deste caramujo poderia não ter nenhuma importância em vários ambientes. No entanto, nas condições ambientais existentes para aqueles caramujos, mudanças bruscas de iluminação normalmente eram causadas por predadores nadando diretamente acima das suas cabeças. Nosso caramujo mutante possui agora um sistema de alerta antecipado, que permite que ele se recolha à sua concha quando um predador se aproxima para atacá-lo. Suas chances de sobrevivência e, conseqüentemente, de se reproduzir são muito maiores do que a dos companheiros não atingidos pela mutação. Uma vez que esta alteração é fruto de uma alteração genética, muitos dos seus descendentes herdarão esta característica e, portanto, também serão maiores suas chances de sobreviver e se reproduzir. Claramente, com o tempo, esta característica passará a dominar a população de caramujos. De pequenos e fortuítos eventos é que são feitas as grandes mudanças.*

*Evoluções futuras podem ser facilmente previstas. Se por mutações genéticas a superfície sensível à luz se curvar sobre uma cavidade hemisférica, suas porções*

*seletivamente iluminadas fornecerão informação direcional a respeito das fontes e oclusões de luz; informações estas que podem acionar respostas motoras direcionais. Em uma criatura móvel, como um peixe, por exemplo, esta característica traz ainda mais vantagens, seja como caça, seja como caçador. Num próximo passo, a cavidade hemisférica pode se transformar em uma cavidade esférica com as células sensíveis à luz no fundo, e um pequeno buraco para o mundo exterior, permitindo que as imagens então se formem na superfície sensível. Eis então que “miraculosamente” um olho foi criado.” [47]*

Muito da história evolutiva dos sistemas nervosos pode ser deduzida analisando-se como os sistemas nervosos se apresentam dentro da árvore filogenética. Dethier & Stellar [61] apresentam vários tipos de sistemas nervosos e os comportamentos suportados por estes diferentes sistemas, desde os sistemas nervosos extremamente simples como o dos celenterados (hydras, medusas, e anêmonas-do-mar), passando por sistemas radiais mais complexos, como o dos equinodermos (holotúrias, ouriços-do-mar, estrelas-do-mar e ofúrios) e chegando aos sistemas nervosos bilaterais, com descrições minuciosas dos sistemas nervosos e comportamentos apresentados por vermes, moluscos, artrópodos e, finalmente, o sistema nervoso dos vertebrados, sistema que também é apresentado por Churchland [47].

Mesmo nos sistemas nervosos dos organismo mais simples já existem neurônio e sinapses completamente desenvolvidos. As mudanças evolutivas que ocorreram, envolveram primeiramente novos agrupamentos e disposições destes neurônios. Estes novos agrupamentos e disposições permitiram maior complexidade de integração. Quase desde o início, duas tendências principais se evidenciaram no sistema nervoso em evolução: uma no sentido de uma divisão de trabalho, na qual neurônios diferentes, ou suas partes, tornaram-se especializados para funções diferentes; a outra no sentido de segregação, na qual unidades semelhantes agrupam-se [30].

Assim, supõe-se que, inicialmente, a recepção nervosa era feita por terminações nervosas indiferenciadas ou por células individuais, porém com a especialização, formaram-se células especializadas para respostas seletivas a mudanças de ambiente, os receptores e o aperfeiçoamento destas em grupos associados formaram os órgãos sensoriais. Hoje em dia, órgãos dos sentidos bastante elaborados são encontrados até mesmo nos cnidários [124].

Do mesmo modo, o desenvolvimento de sistemas especializados de respostas formaram as células musculares e camadas musculares organizadas e a canalização de condução da excitação deu origem a redes nervosas e tratos nervosos. Esta especialização de função surgiu simultaneamente com a tendência de separação espacial. Os sistemas sensoriais obviamente tendem a permanecer próximos à superfície do

animal, os de resposta situam-se em níveis mais profundos: conseqüentemente, as porções sensoriais e motoras do sistema nervoso tornaram-se localizadas.

Elementos condutores, que começaram como redes nervosas situadas perto da superfície, em um processo de evolução aprofundaram-se dentro do animal e se tornaram os elos de conexão entre receptores e efetores. Nos organismos mais simples, as conexões têm sido relativamente simples e diretas. No entanto, o número de neurônios intervindo entre receptor e efector aumenta nos organismos mais complexos. Estes interneurônios não são, em geral, localizados, mas sim espalhados e constituem os tratos fibrosos. Estes, que representam uma condensação da rede nervosa, são ainda formados por uma mistura de fibras e seus corpos celulares.

Estes desenvolvimentos permitiram aos animais fazerem avaliações mais refinadas das mudanças de ambiente, movimentos mais diversificados e respostas mais rápidas, dirigidas e coordenadas. Apesar disso, em termos de comportamentos suportados pelo sistema nervoso, os celenterados e equinodermos permanecem como animais com respostas limitadas, baixo nível de coordenação e ausência de controle central [61].

À medida que os animais evoluíram, houve uma tendência crescente em reservar os feixes de fibras para elementos condutores e reunir os corpos celulares em massas localizadas (gânglios). Os feixes ou cordões, sendo sistemas de condução direta, transmitem mais rapidamente que uma rede.

Como o sistema nervoso condensou-se no curso da evolução, a partir de uma rede difusa, ele não permaneceu funcionalmente semelhante na sua totalidade. O aumento da importância dos gânglios em controlar as áreas do corpo que eles inervam exigiu finalmente que eles mesmos estivessem sob algum controle superior; de outra forma, o animal teria de agir como uma república de partes, e não como uma unidade. O papel de centro-mestre de controle recaiu sobre os gânglios anteriores, especialmente os supra-esofágicos (cerebrais) [61].

Este processo continuou nos vertebrados. Os vertebrados primitivos possuíam um gânglio central alongado que percorria o sentido da espinha. Este gânglio se conectava com outras partes do corpo por dois conjuntos de fibras distintas física e funcionalmente. As fibras somatosensórias traziam informações sobre as atividades dos músculos e estímulos tácteis para o cordão central. Este, por sua vez servia para coordenar os diversos músculos do corpo uns com os outros de modo a produzir um movimento natatório coerente e ainda para coordenar este movimento com as circunstâncias percebidas pelo animal, tais como a fuga de predadores ou a procura por alimento.

Mais tarde, este cordão espinhal adquiriu um alongamento na extremidade an-

terior com três protuberâncias onde a população e densidade de células nervosas atingiram um outro nível. Este encéfalo primitivo pode ser dividido no encéfalo anterior (proencéfalo), o encéfalo médio (mesencéfalo) e o encéfalo posterior (rombencéfalo). Estas três estruturas ainda hoje podem ser observadas no desenvolvimento embrionário de todos os vertebrados [61].

A partir daí, passou a haver uma especialização de função de cada uma destas estruturas. As redes nervosas do encéfalo anterior se especializaram no processamento de estímulos olfativos; o encéfalo médio processava as informações visuais e auditivas; e o encéfalo posterior se especializou em uma sofisticada coordenação da atividade motora. O cérebro dos peixes atuais ainda permanecem neste estágio, com o encéfalo médio sendo a estrutura dominante.

Nos vertebrados superiores, tais como os répteis e os anfíbios, observa-se que o encéfalo anterior é dominante na anatomia cerebral, assumindo a tarefa de fazer o processamento principal de todas as modalidades dos sentidos.

Os mamíferos primitivos apresentam não só uma maior complexidade e especialização do encéfalo anterior, mas também o surgimento de duas novas estruturas: os hemisférios cerebrais e o cerebelo. Os hemisférios cerebrais, passaram a conter um grande número de áreas especializadas, incluindo controles superiores de aprendizado e disparo de comportamentos; enquanto o cerebelo garantia uma melhor coordenação dos movimentos corporais.

Nos mamíferos típicos, estas novas estruturas, apesar de proeminentes não eram grandes se comparadas com as outras estruturas cerebrais. Nos primatas, no entanto, elas se tornaram as estruturas dominantes, sendo que no ser humano se tornaram enormes. Hoje em dia, é difícil visualizar o encéfalo primitivo debaixo do “guardachuva” formado pelos hemisférios cerebrais, do mesmo modo, o cerebelo também é claramente maior do que o de outros primatas [47].

## 5.4 Evolução do Comportamento

### 5.4.1 Sistema Nervoso e Comportamento

Tomando-se a evolução do sistema nervoso na árvore filogenética como referência de como evolui o sistema nervoso ao longo da história dos seres vivos sobre a Terra, fica claro que a evolução do comportamento está baseada principalmente no aumento da complexidade do Sistema Nervoso sem, no entanto, deixar de lado a importância dos sistemas sensórios e motores.

Baseado na teoria evolucionária, podemos imaginar uma criatura extremamente primitiva, com um sistema nervoso sensório-motor muito simples. O único requi-



sito para a sobrevivência desta criatura seria a capacidade de sentir a presença de alimento e de se mover em direção a ele em um ambiente livre de predadores e obstáculos. É fácil verificar que este tipo de comportamento, puramente reflexivo pode ser implementado através da ligação direta entre dois neurônios ou grupos de neurônios, os aferentes (sensoriais) e os eferentes (motores). A medida que o ambiente foi se tornando mais complexo e hostil, a luta pela sobrevivência passou a exigir criaturas mais aptas, com um repertório maior e mais elaborado de comportamentos. Reflexos mais complexos, necessitariam então de neurônios intermediários. Exemplos podem ser encontrados ainda hoje em dia em animais inferiores. O repertório comportamental de um poliqueta sedentário praticamente se limita a encolhimentos bruscos, uma resposta mediada por fibras nervosas gigantes. Nas anêmonas-do-mar, a retração do tentáculo, a contração da coluna e, em algumas formas, a natação, constituem todo o comportamento observável do animal [124].

Na luta pela sobrevivência, houve um momento em que a capacidade de adaptação ao ambiente não pode mais ser assegurada por um repertório de comportamentos puramente reflexivos, implementados através de redes nervosas diretas. Sobre as redes nervosas diretas, o processo evolucionário desenvolveu novas topologias de redes nervosas, incluindo ciclos entre os neurônios. Esta nova arquitetura de controle possibilitou o surgimento de outras modalidades de comportamento, aumentando a chance de sobrevivência [152].

Este ponto-de-vista é compartilhado também por outros pesquisadores, entre eles podemos citar Vaario [184]:

*“Our approach is to view the evolution process as a tool for growing gradually the complexity of systems i.e. the repertoire of behaviors. When complexity of systems, especilly the behavior of them increases, the surviving probability also increases.”*

### 5.4.2 Genética e Comportamento

A descoberta de que certos padrões de comportamento são herdados, foi uma contribuição importante para o estudo da evolução. Comportamentos determinados geneticamente estão sujeitos às pressões da seleção natural.

Lorenz argumentava que as similaridades observadas nos padrões de comportamento poderiam ser utilizadas para reconstruir-se a história evolucionária [172].

Como o comportamento é o resultado da interação de órgãos sensoriais, sistema nervoso, músculos e outras partes do corpo do animal, supõe-se que uma variação em alguns destes ítems alterará também o comportamento. Exemplos de como os fatores genéticos de reprodução e mutação afetam o comportamento observável são fartamente encontrados na Natureza.

A membrana celular do unicelulado *Paramecium*, por exemplo, é afetada pela presença de genes mutantes. Animais com membranas celulares anormais terão, por sua vez, comportamento estranho. O mutante “Pawn”, por exemplo, não consegue se locomover para trás porque sua membrana celular não permite a passagem de íons de cálcio de maneira normal.

Em grilos machos (*Teleogryllus*), que atraem sua fêmea pelo canto (produzido ao raspar rapidamente uma asa na outra), uma alteração no padrão dos impulsos nervosos dirigidos aos músculos das asas muda completamente a natureza do canto. Alguns genes no macho são responsáveis por afetar o número e o padrão dos impulsos nervosos que vão para as asas. Cruzar duas espécies que têm cantos diferentes resulta em machos que têm cantos intermediários entre as duas. O padrão dos impulsos nervosos para as asas também é intermediário. Assim, os genes afetam as células nervosas, que afetam os músculos das asas, que afetam o canto produzido que, por sua vez, afeta a atratividade do machos pelas fêmeas. A rota dos genes ao comportamento é longa mas pode ser traçada, pelo menos neste caso [58].

No entanto, o código genético isoladamente não é suficiente para caracterizar um comportamento. É importante ressaltar que nem sempre o fato do genótipo predispor a um determinado comportamento, o comportamento propriamente dito, ou seja, o fenótipo resultante daquele genótipo irá aparecer. O ambiente também é um elemento importante para a emergência de um comportamento [184], pois os fatores ambientais adequados para o disparo do comportamento devem estar presentes, além do mais, características do comportamento ainda podem ser alteradas como função do aprendizado.

## 5.5 Computação Evolucionária

Computação evolucionária é o nome genérico dado à métodos computacionais inspirados na teoria da evolução. Os algoritmos usados em computação evolucionária se chamam algoritmos evolucionários. Esta seção aborda de maneira introdutória os principais algoritmos evolucionários. Informações mais completas e outras referências podem ser encontradas em [83] [19] [66] [10] entre outros.

Estes algoritmos incluem os Algoritmos Genéticos propostos por Holland [86], a Programação Evolucionária, apresentada por Fogel, Owens & Walsh [67], as Estratégias Evolucionárias introduzidas por Rechenberg e Schwefel [148] [168] e outros. Estes algoritmos possuem em comum a característica de realizar uma investigação em paralelo de várias posições de um espaço de busca através da manipulação de uma população de soluções possíveis para um dado problema. Como eles são inspira-

dos nos mecanismos da evolução biológica, utilizam mecanismos de seleção, mutação e reprodução neste processo de busca de soluções [19] [10] [149].

Os algoritmos evolucionários possuem como característica marcante o fato de separar a descrição formal do sistema objeto do problema característico deste sistema que se deseja resolver. Diversas alternativas de solução do problema característico são transformadas em números reais através de uma função de aptidão que então são manipulados pelo solucionador do problema, sem a necessidade de um conhecimento explícito de como o sistema objeto funciona. Por esta razão, os algoritmos evolucionários são extremamente vantajosos como método de otimização, principalmente nos casos em que não se tem um conhecimento explícito da função que descreve o funcionamento do sistema que se deseja otimizar.

Os dados dos algoritmos evolucionários consistem em uma população de objetos abstratos (por exemplo, estruturas simbólicas, redes neurais, palavras de um alfabeto, variáveis de um problema de otimização, programas de computador, etc). Estes formam os objetos ou “indivíduos” sobre o qual operarão os operadores genéticos. Em um primeiro momento, através de um mecanismo randômico, introduz-se uma grande diversidade na população inicial, com vários “indivíduos” diferentes, o que corresponderia as primeiras tentativas de solução do problema. Depois disto, estes objetos são manipulados por operadores inspirados nas regras de seleção. Cada indivíduo da população tem um certo grau de aptidão, o qual depende do ambiente. Os indivíduos de maior aptidão podem se reproduzir mais facilmente. Em seguida, atuam outros operadores inspirados na evolução biológica, os operadores genéticos que agem como operadores de busca na solução de um determinado problema. Os mais comuns são recombinação e mutação. Desta forma a população tem a tendência de, após várias gerações, ter elementos com altos valores de aptidão [19].

Os primeiros trabalhos envolvendo algoritmos evolucionários foram em um primeiro momento duramente criticados, pois muitos pesquisadores os comparavam com métodos de busca aleatória [108]. No entanto é importante ressaltar que, apesar dos algoritmos evolucionários envolverem um certo grau de aleatoriedade, não se trata de uma busca aleatória [66].

De maneira genérica, um algoritmo evolucionário pode ser descrito, sem entrar em detalhes, pelo seguinte algoritmo.

- Inicialize uma população de indivíduos, usualmente de modo aleatório.
- Calcule o valor da aptidão dos indivíduos da população.
- Verifique um critério de parada e enquanto este critério não for satisfeito faça:
  - Avance o tempo.

- Forme descendentes usando alguma forma de perturbação.
- Calcule aptidão.
- Enumere sobreviventes selecionados em função da aptidão.

Nas seções seguintes expõe-se, com detalhes, os algoritmos evolucionários mais conhecidos.

### 5.5.1 Algoritmos Genéticos

Algoritmos Genéticos (AG) constituem um paradigma de aprendizado pela máquina inspirado em um dos mecanismos básicos da evolução na Natureza, chamado seleção dura.

Neste trabalho apresenta-se, de maneira introdutória, os conceitos e características envolvidas nas pesquisas sobre AG. Para o leitor interessado em aprofundar seus conhecimentos nesta área sugere-se os trabalhos de Holland [86] e Goldberg [75]. Outras fontes interessantes são ainda [175], [149], [57] e [125]. Tanomaru [179] apresenta um excelente resumo em português, voltado para aplicações de otimização.

Holland [86] propôs os AGs em 1975 como programas de computador inspirados nos processos evolucionários encontrados na Natureza. AGs manipulam uma população de soluções potenciais para um problema de otimização ou de busca. Estes programas operam especificamente em representações codificadas das soluções, equivalentes ao material genético dos indivíduos na Natureza, e não, diretamente, nas soluções propriamente ditas. O AG de Holland codifica as soluções como cadeias de bits de um alfabeto binário. Assim como na Natureza, a seleção é o mecanismo básico que faz com que as melhores soluções sobrevivam. Cada solução é associada com um valor de aptidão que reflete quão boa ela é quando comparada com outras soluções da população. Quanto maior o valor de aptidão de um indivíduo, maiores as suas chances de sobreviver e de se reproduzir, aumentando sua representatividade nas gerações subseqüentes. A recombinação do material genético no AG é simulada através de um mecanismo de cruzamento (“crossover”) que troca pedaços de cadeias de bits entre soluções. Outra operação chamada mutação, provoca alterações esporádicas e aleatórias de bits da cadeia. A mutação também tem uma analogia direta com o que ocorre na Natureza e tem como finalidade a regeneração de material genético perdido e a exploração de novas possibilidades de solução.

Desde que Holland propôs os AGs, inúmeras contribuições têm surgido sugerindo alterações na idéia original. No entanto, visando introduzir o leitor no assunto, a abordagem seguirá o modelo de AG proposto por Holland, o qual é denominado atualmente de Algoritmo Genético Simples (AGS).

## 5.5.2 Algoritmo Genético Simples

Essencial ao funcionamento do AGS é uma população inicial de soluções candidatas. Cada indivíduo da população é codificado como um vetor  $x$ , chamado cromossomo cujos elementos são descritos como genes. Holland [86] propôs que os cromossomos deveriam ser representados por uma cadeia (“string”) de bits. Cada cadeia de 0’s e 1’s é a versão codificada de uma solução para o sistema que se deseja otimizar. Utilizando operadores genéticos - cruzamento e mutação - o algoritmo cria a geração subsequente através das cadeias da população atual. Este ciclo gerador de novas cadeias é repetido até que determinado critério para término seja alcançado (por exemplo, até que um certo número de gerações seja processado) [175].

Os seguintes componentes fazem parte de um AG:

- uma população de cadeias binárias.
- parâmetros de controle.
- uma função de aptidão.
- operadores genéticos (cruzamento e mutação).
- um mecanismo de seleção.
- um mecanismo que codifique as soluções como cadeias binárias.

### Mecanismo de Codificação

Holland [86] sugeriu que se deveria codificar de forma binária as variáveis do problema que se desejasse otimizar [66]. No entanto, o mecanismo de codificação depende da natureza das variáveis do problema. Por exemplo, quando utilizado para determinar o fluxo ótimo em um problema de transporte, as variáveis (vazão nos diferentes condutos) assumem valores contínuos, enquanto que as variáveis em um problema do tipo caixeiro-viajante são quantidades binárias representando a inclusão ou exclusão de um percurso em um circuito Hamiltoniano. De qualquer modo, em cada caso, o mecanismo de codificação deve mapear cada solução a uma única cadeia binária [175].

### Função de Aptidão

É a função objetivo, a função a ser otimizada que possibilita o mecanismo para avaliar cada cadeia. No entanto sua gama de valores varia de problema para problema. De modo a manter a uniformidade sobre vários domínios de problemas, normalmente

se usa uma função de aptidão que normalize a função objetivo para o intervalo conveniente de 0 a 1. O valor normalizado da função objetivo é a aptidão da cadeia binária que o mecanismo de seleção utiliza para avaliar as cadeias da população.

### Seleção

A seleção tem por objetivo modelar o processo que ocorre na Natureza em que os elementos mais aptos têm maiores chances de sobreviver e participar do processo de geração da próxima geração. No AGS, uma solução mais apta gera uma grande descendência aumentando então a chance de sobrevivência na geração subsequente. No esquema de seleção proporcional, para uma solução com um valor de aptidão  $f_i$ , são alocados  $f_i/\bar{f}$  descendentes, onde  $\bar{f}$  é o valor de aptidão média da população. Assim, para uma solução com valor de aptidão maior que o valor de aptidão médio da população são alocados mais de um descendente, enquanto um indivíduo com valor de aptidão menor que a média da população é alocado menos de um descendente. Como na prática não há como gerar números fracionários de descendentes, os métodos de alocação normalmente transformam estes valores em medidas de probabilidade de geração de descendentes.

O AGS utiliza o esquema de seleção pelo método da roleta ponderada para implementar a seleção proporcional. A cada indivíduo da população é alocado um setor de uma roleta. O ângulo deste setor é dado pela fórmula  $2\pi f_i/\bar{f}$ . Um determinado indivíduo gerará um descendente se um número gerado aleatoriamente entre 0 e  $2\pi$  cair no setor correspondente ao indivíduo. O algoritmo seleciona indivíduos deste modo até que tenha sido gerada toda a população da próxima geração. Exemplo de uma roleta aparece na Figura 5.2 usando uma determinada população de indivíduos.

O esquema da seleção pela roleta ponderada pode gerar erros no sentido do número de descendentes gerados para um determinado indivíduo não ser aquele esperado. Estes valores só se aproximam para populações muito grandes.

### Cruzamento (“Crossover”)

Após a seleção, tem lugar a etapa de cruzamento. Nesta etapa, pares de indivíduos da população são escolhidos aleatoriamente para se cruzarem. O AGS utiliza a abordagem mais simples - o cruzamento em um único ponto. Assumindo que  $l$  seja o tamanho da cadeia binária, um ponto de cruzamento é escolhido de maneira aleatória no intervalo de 1 até  $l - 1$ . As porções das duas cadeias binárias dos dois indivíduos

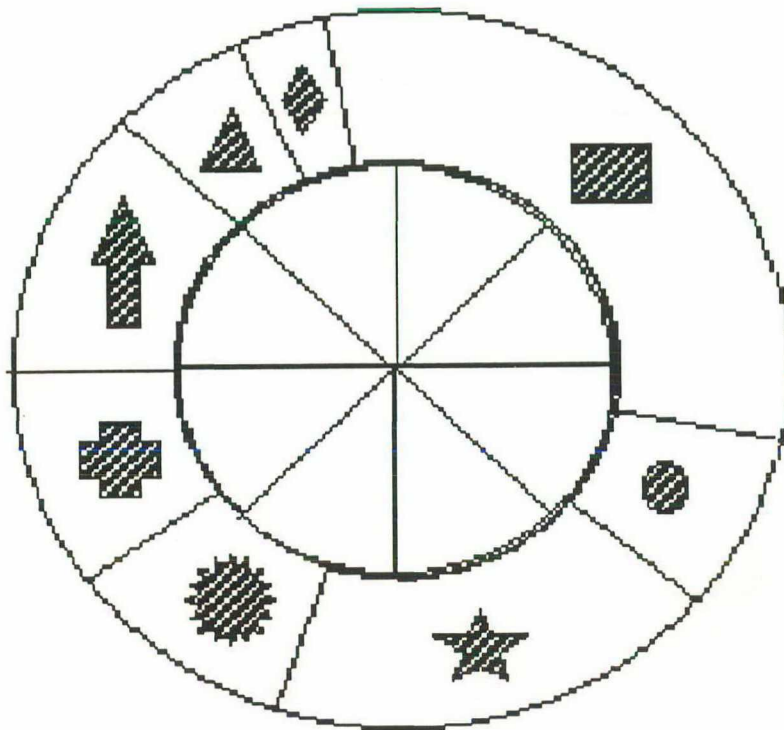


Figura 5.2: Exemplo de roleta para seleção

entre este ponto de cruzamento são trocadas para formar duas novas cadeias. No entanto, nem sempre o cruzamento ocorre. Uma vez escolhidos um par de indivíduos para realizar o cruzamento, este só ocorre se um número escolhido aleatoriamente entre 0 e 1 for maior que  $p_c$ , chamado de taxa de cruzamento (“crossover rate”). Em grandes populações,  $p_c$  fornece a fração dos indivíduos que serão realmente cruzados. O processo de cruzamento é ilustrado conceitualmente na Figura 5.3.

### Mutação

Após o cruzamento, os indivíduos estão sujeitos à mutação. A mutação de um bit significa a inversão do seu valor. Da mesma forma que  $p_c$  controla a probabilidade de um cruzamento ocorrer, outro parâmetro  $p_m$  (a taxa de mutação) fornece a probabilidade de um bit ser invertido. Os bits de uma cadeia são mutados de uma maneira independente, ou seja, o fato de um bit de uma cadeia ter sofrido uma mutação não impede que outros bits desta mesma cadeia também o sejam. O AGS trata a mutação como um operador genético secundário com o objetivo de restaurar material genético perdido. Por exemplo, suponha que todas as cadeias de uma população tenham convergido para 0 em determinado bit e uma solução ótima deveria



Figura 5.3: Processo conceitual da operação de cruzamento

possuir um 1 naquela posição. Então o cruzamento não seria capaz de gerar o 1 naquela posição, enquanto a mutação o seria. O processo de mutação é ilustrado conceitualmente na Figura 5.4.

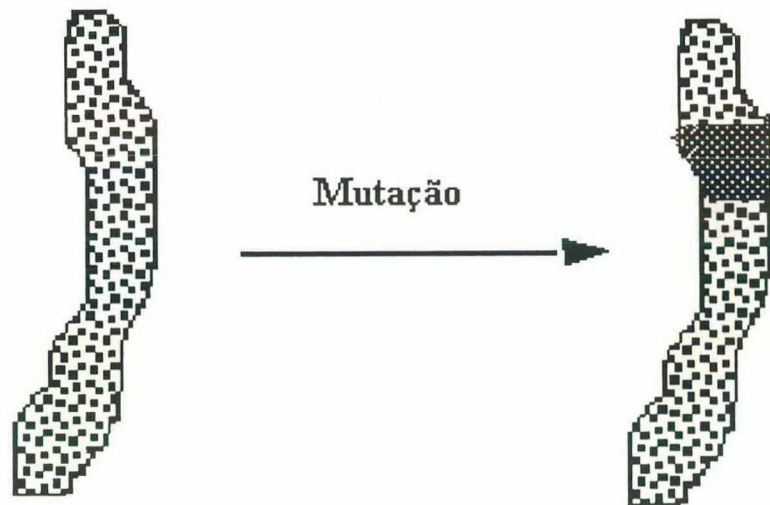


Figura 5.4: Processo conceitual da operação de mutação

### Exemplo de Funcionamento do AGS

A Figura 5.5 mostra um ciclo de funcionamento do AGS sobre uma população inicial (P1) de quatro cadeias de 10 bits cada (indivíduos). Para este exemplo específico, a



função objetivo fornece o número de bits 1 em uma cadeia. Neste caso, o valor pode variar de 0 a 10. Para normalizar a função objetivo a valores entre 0 e 1, a função de aptidão realiza uma operação de “divisão por 10” no número de bits 1 de cada indivíduo.

<b>População P1:</b>	
<b>String (indivíduos)</b>	<b>Valor de Aptidão</b>
0000011100	0.3
1000011111	0.6
0110101011	0.6
1111111011	0.9
<b>População P2: Após a Seleção</b>	
<b>String (indivíduos)</b>	<b>Valor de Aptidão</b>
1000011111	0.6
0110101011	0.6
1111111011	0.9
1111111011	0.9
<b>População P3: Após o Cruzamento</b>	
<b>String (indivíduos)</b>	<b>Valor de Aptidão</b>
10000 11011	0.5
0110101011	0.6
1111111011	0.9
11111 11111	1.0
<b>População P4: Após a Mutação</b>	
<b>String (indivíduos)</b>	<b>Valor de Aptidão</b>
1000011011	0.5
0110101011	0.6
1111111011	0.9
<b>0</b> 1111111111	1.0

Figura 5.5: Exemplo de um ciclo do Algoritmo Genético Simples

Assim, na população P1 os quatro indivíduos possuem valores de aptidão de 0.3, 0.6, 0.6 e 0.9. Teoricamente, o mecanismo de seleção proporcional deveria alocar 0.5, 1, 1 e 1.5 descendentes para cada indivíduo. No entanto, neste caso, a alocação final de descendentes é 0, 1, 1 e 2. Na figura a população P2 representa este conjunto selecionado de descendentes.

A seguir, as quatro cadeias são pareadas randomicamente para cruzamento. As cadeias 1 e 4 formam um par e as cadeias 2 e 3 formam outro. Com uma taxa de cruzamento de 0.5, apenas o par das cadeias 1 e 4 é escolhido para cruzamento, enquanto o outro par é deixado intacto. O ponto de cruzamento cai entre o quinto e o sexto bits das cadeias e os bits de 1 a 5 das cadeias dos dois indivíduos são trocados. A população P3 representa o conjunto de indivíduos após o cruzamento.

A operação de mutação na população P3 pode ser vista na população P4 no bit 10 do indivíduo 4. Apenas um bit, de um total de 40 foram mutados, representando uma taxa de mutação de 0.025. A população P4 representa a próxima geração.

Este exemplo tem apenas caráter ilustrativo. Tipicamente o AGS usa uma população de 30 a 200 indivíduos, taxas de cruzamento de 0.5 a 1 e taxas de mutação de 0.001 até 0.05. Estes parâmetros - o tamanho da população, a taxa de cruzamento e a taxa de mutação são conhecidos como parâmetros de controle do AGS e devem ser especificados antes da sua execução.

Para terminar a execução, deve-se especificar um critério de parada. Este pode ser um determinado número de gerações, um determinado valor de aptidão atingido por algum indivíduo ou até que todos os indivíduos da população tenham atingido um certo grau de homogeneidade [175].

### Alterações do AGS

Durante a década passada, várias pesquisas foram desenvolvidas no sentido de entender e melhorar o desempenho do AGS proposto por Holland. Foram sugeridas alterações nos algoritmos de seleção a fim de reduzir os erros de amostragem. Mecanismos de cruzamento tais como “dois-pontos”, “multiponto” e “uniforme” foram propostos como melhorias à técnica tradicional de cruzamento em um único ponto. Códigos cinza (“gray codes”) e codificação dinâmica foram propostos para solucionar alguns problemas associados com a codificação de números reais fracionários. Finalmente, partindo da política tradicional de valores fixos para os parâmetros de controle, técnicas adaptativas variam dinamicamente alguns dos parâmetros de controle. Trabalhos recentes tem focado o desenvolvimento de algoritmos genéticos distribuídos e algoritmos genéticos paralelos [175].

### 5.5.3 Programação Evolucionária

A Programação Evolucionária (PE) foi concebida por Lawrence J. Fogel em 1960, sendo portanto anterior aos AGs. É uma estratégia de otimização que utiliza um modelo evolutivo em um nível de abstração mais elevado do que os algoritmos genéticos

e as estratégias evolucionárias. Esta abordagem enfatiza principalmente a relação comportamental entre os pais e seus descendentes, inspirada na observação da Natureza em que os comportamentos dos filhos geralmente são semelhantes aos dos pais com pequenas variações [19] [10].

A programação evolucionária foi popularizada com o livro de Fogel, Owens & Walsh [67]. Neste livro, um autômato de estados finitos evolui e consegue prever seqüências de símbolos gerados por um processo de Markov e uma série temporal não estacionária. A motivação para a escolha deste problema foi a suposição de que predição é uma característica do comportamento inteligente [19].

### Algoritmo de Programação Evolucionária

O funcionamento da PE é feito essencialmente em quatro etapas que são iteradas.

1. Escolha uma população inicial aleatoriamente. O número de elementos desta população inicial influencia a velocidade com que se encontrará a solução do problema mas, no momento, não existe critério que permita, de modo seguro, saber qual este número, só que deve ser no mínimo 1;
2. Cada elemento da população se reproduz dando um novo elemento e produzindo uma nova população. Neste ponto, os elementos da população sofrem a ação do operador mutação, cujo valor pode variar;
3. Calcula-se a aptidão de cada elemento da população resultante;
4. Faz-se a população se reproduzir, com número de descendentes um função monotonicamente crescente da aptidão. Nesta etapa não se obriga a população total a continuar constante, nem que todos os elementos se reproduzam, mas se determina, também (freqüentemente de modo aleatório), o número de elementos da nova população.

#### 5.5.4 Estratégias Evolucionárias

Em 1963, dois estudantes da Universidade Técnica de Berlim, Ingo Rechenberg e Hans-Paul Schwefel, iniciaram uma estreita colaboração no túnel de vento da Universidade, buscando a determinação de formas ótimas de corpos sujeitos a um fluxo de ar.

Rechenberg teve a idéia de tentar modificações aleatórias da forma, tal como ocorre nas mutações nos seres vivos. Ele deu ao método o nome de Estratégia Evolucionária (EE). Um terceiro estudante, Peter Bienert, uniu-se aos dois primeiros

e iniciou a construção de um sistema para fazer automaticamente o trabalho usando inspiração biológica e implementando os operadores de mutação e seleção [19].

Nesta técnica, a ênfase não está na obtenção de uma estrutura com um alto grau de aptidão, como ocorre com os AGs, mas sim na obtenção de um comportamento com uma boa avaliação pela função de aptidão. Esta mudança de enfoque, da busca de estruturas para a busca por comportamentos pode ser ao mesmo tempo benéfica e problemática. Através da busca de otimizações de comportamentos, as heurísticas de codificação e reprodução devem ser capazes de criar objetos que sejam comportamentalmente similares aos seus pais, porém não necessariamente similares ao nível estrutural.

Com a ênfase focada na modelagem da evolução ao nível de um indivíduo, as EEs consideram um indivíduo como sendo composto de um conjunto de comportamentos, cada um deles sendo uma característica. A interação entre estas características é tipicamente desconhecida. Como resultado, sistemas que implementam EEs usam cadeias de tamanho fixo e com valores no conjunto dos reais. Cada posição representa um valor para determinado comportamento.

Como as EEs modelam a evolução ao nível do indivíduo, não existe no algoritmo a oportunidade de dois indivíduos serem recombinaados e criarem um descendente, tal como ocorre na etapa de cruzamento dos AGs.

O operador de mutação das EEs aplica um ruído Gaussiano a cada comportamento componente. A média é o valor atual do comportamento. A variância do ruído é determinada normalmente pela “regra 1/5” como descrita por Angeline [10] a seguir:

*A regra 1/5 - A razão de mutações bem sucedidas em relação a todas as mutações deve ser de 1/5. Se ela for maior que 1/5 então decremente a variância de uma constante  $c_d$ , se ela for menor que 1/5 então incremente a variância por uma constante  $c_i$ ;*

onde uma mutação é bem sucedida se o indivíduo resultante possuir um valor de aptidão melhor que o seu antecessor. Quando esta regra diminui a variância, a busca fica mais restrita, com o filho possuindo um comportamento mais próximo do seu pai. Quando a variância aumenta, este algoritmo dá chance de o filho ter um comportamento mais distante do seu pai [10].

Os indivíduos que possuem um valor de aptidão maior se tornarão os pais para a próxima geração. Este processo continua até que se chegue a um valor de aptidão aceitável ou um certo número de gerações seja examinada [66].

### 5.5.5 Diferenças entre os Algoritmos Genéticos e outros Algoritmos Evolucionários

Fogel [66] coloca que, apesar do conceito de evolução ser um conceito universalmente aceito, existem basicamente dois pontos de vista diametralmente opostos sobre a que nível opera a evolução, o que leva a filosofias diferentes de implementações de algoritmos que simulam a evolução. Para alguns autores, como Davis [57], “evolução é um processo que opera nos cromossomos e não nos indivíduos que eles codificam”. Já Mayr [120] sugere que este reducionismo não é satisfatório para explicar a evolução, para ele, “evolução é uma mudança na adaptação e na diversidade de populações de organismos”.

Os AGs seguem a filosofia sugerida por Davis e simulam a evolução em uma população de cadeias de símbolos (geralmente retirados do alfabeto  $[0,1]$ ) que seriam os cromossomos. Os bits que compõem estas cadeias seriam análogos aos genes. Sobre estes elementos, então, são realizadas as operações normalmente encontradas nos sistemas genéticos naturais (cruzamentos, inversões, mutações, etc). Em cada cromossomo da população age, então, uma função de aptidão que avalia cada indivíduo em relação a uma função objetivo através da transformação da cadeia codificada (conversão de um genótipo em um fenótipo).

Por outro lado, se for dada ênfase na adaptação e diversificação de uma população, os algoritmos que simulam a evolução, neste caso AEs e EEs, continuam operando sobre uma população de cadeias codificadas, no entanto, os elementos destas cadeias representam tipicamente traços de comportamento, sejam de indivíduos ou sejam de populações inteiras (espécies), sem uma relação de como estes traços de comportamento são estruturados. Mutações são aplicadas a cada cadeia de modo que uma estreita relação comportamental seja mantida entre cada antecessor e seu descendente. Os critérios de seleção são utilizados para avaliar o comportamento resultante de cadeia no ambiente específico e com isto determinar quais membros da população devem ser mantidos nas gerações futuras.

Apesar de inspiradas no processo de evolução biológica, nenhuma das estratégias aqui apresentadas é completa. Na verdade, quando comparadas com o processo genético/evolucionário que ocorre na Natureza, os algoritmos simulados em computador são extremamente simplificados. Barreto [19] apresenta uma série que diferenças entre o que ocorre no processo de evolução biológica e o que é simulado nos algoritmos evolucionários, aos quais podemos acrescentar:

- O fato dos algoritmos genéticos não simularem os fenômenos de pleiotropia (um gene pode afetar várias características do fenótipo) e poligenia (determinada

característica do fenótipo pode ser determinada por vários genes).

- Na Natureza, dependendo do ambiente, determinado fenótipo pode nunca se manifestar, apesar do gene para esta característica estar presente. Por exemplo, no caso de fenilcetonúricos que são sensíveis a alimentação contém fenilalanina e que pode causar retardamento mental nos seres humanos. Se o indivíduo cujo genótipo indica fenilcetonúria, nunca for exposto a alimentos contendo fenilalanina, o fenótipo resultante (traço de comportamento) que é o retardamento mental nunca aparecerá. Esta característica raramente é levada em conta nos trabalhos envolvendo algoritmos evolucionários.
- Na Natureza, os organismos evoluem do mais simples para o mais complexo ao longo do tempo. Por exemplo, ao longo da evolução apareceram primeiro sistemas nervosos mais simples que foram se tornando mais complexos ao longo do processo evolutivo. Muitos algoritmos evolucionários não se preocupam com este aspecto, preconizando inclusive que as populações iniciais normalmente procurem se distribuir uniformemente no espaço de busca.

Finalizando, conclui-se que o tipo de algoritmo evolutivo que se vai implementar ou utilizar e o grau de precisão desta implementação em reproduzir aspectos que ocorrem na evolução biológica, está fortemente relacionado com o problema que se deseja resolver. É o caso da utilização de algoritmos evolucionários para geração de redes neurais artificiais e cujas alternativas serão abordadas com mais detalhes na próxima seção.

## 5.6 Computação Evolucionária na Geração de Redes Neurais

A geração de uma rede neural artificial na sua forma completa envolve a definição tanto da topologia e quantidade de neurônios como dos valores de pesos das conexões adequados [10] [9] [25]. Metodologias para a geração de estruturas e parâmetros de redes neurais têm sido intensamente pesquisadas durante a última década. Uma destas metodologias envolve a utilização da computação evolucionária para a geração de redes neurais.

A maioria dos trabalhos em que algoritmos evolucionários são usados tratam do problema de treinamento, ou seja, da geração dos pesos das conexões da rede [55] [33]. Nestes trabalhos, dada uma determinada estrutura para a rede, um algoritmo evolucionário é utilizado para achar quais os pesos que resultam nos menores

valores de erro, ao invés de algoritmos de treinamento convencionais, como o “back-propagation”. Por exemplo, Montana [129] mostrou como é possível treinar redes neurais artificiais usando AGs. Muhlenbein [133] abordou a dinâmica da evolução combinada com aprendizado, dando os primeiros passos para a compreensão dos dois mecanismos agindo simultaneamente [132]. Prado [146] e Porto [144] usando respectivamente AGs e AEs, exploram o treinamento de redes neurais diretas [19] [69].

Trabalhos em que algoritmos evolucionários são utilizados para a escolha da topologia da rede neural artificial melhor adaptada à solução de um problema são mais raros. Nestes trabalhos, geralmente os genes codificam a topologia da rede, especificando que conexões estão presentes. Os pesos são então determinados por outros métodos. Como exemplo de pesquisas nesta área pode-se citar os trabalhos de Garcia [69], Vico & Sandoval [185] e Boers & Kuiper [30].

Uma terceira alternativa é utilizar os algoritmos evolucionários tanto para a geração da topologia como dos pesos das conexões, alguns trabalhos nesta área foram publicados por Karunanithi, Das & Whitley [94] e Angeline [10] [9].

A seguir descreve-se mais detalhadamente os trabalhos de Boers & Kuiper e de Angeline, por conterem informações e técnicas extremamente interessantes relacionadas com a geração de redes neurais utilizando algoritmos evolucionários.

O trabalho de Boers & Kuiper é extremamente interessante na medida em que eles são mais fiéis às inspirações biológicas para a geração de redes neurais. No seu trabalho eles utilizam AGs para a geração de redes neurais, no entanto, ao invés dos genes codificarem a presença ou não de conexões entre os neurônios de uma rede, estes codificam uma “receita” contendo regras de produção que serão transformadas na especificação estrutural da rede.

Para eles, uma rede neural pode ser encarada como um agregado de nós e arcos, ou seja, um grafo. Deste modo, o que seria necessário para codificar a estruturas redes neurais seria um método para geração de grafos. Existem várias linguagens formais para descrição e geração de grafos. Boers & Kuiper [30] buscaram na biologia uma linguagem para descrever e gerar as estruturas de redes neurais, os Sistemas de Lindenmayer. Para descrever a forma e o crescimento de vegetais, o biologista Aristid Lindenmayer desenvolveu uma construção matemática, chamada ‘L-systems’ e a utilizou para descrever o desenvolvimento de organismos multicelulares. Maiores detalhes sobre os Sistemas de Lindenmayer, incluindo extensas formalizações matemáticas podem ser encontradas em [107], [195], [184] e [30].

Na sua pesquisa eles tentam combinar três métodos com forte inspiração biológica:

- Algoritmos Genéticos;

- Sistemas de Lindenmayer;
- Redes Neurais.

Seu objetivo é desenvolver uma metodologia de projeto que procurasse de modo automático arquiteturas modulares ótimas para redes neurais artificiais. Esta metodologia pode ser resumida nos seguintes passos:

1. Um AG gera uma cadeia de bits, esta cadeia corresponde ao cromossomo de um membro da população. A busca do AG é direcionada para indivíduos que possuam alto grau de aptidão, resultante da função de avaliação realizada na etapa 3.
2. Um Sistema de Lindenmayer implementa o crescimento da rede neural resultante da “receita” codificada no cromossomo. O cromossomo é decodificado e transformado em um conjunto de regras de produção. Estas são aplicadas em um axioma por um certo número de iterações, e a lista resultante é transformada na especificação estrutural da rede.
3. Um simulador de rede neural treina a rede resultante para o problema específico. O erro resultante é então transformado na medida de aptidão: um baixo erro representa uma grande aptidão. Esta aptidão é retornada para o AG [30].

Já o trabalho de Angeline [10] [9] faz uma análise comparativa da utilização de algoritmos genéticos, programação evolucionária e estratégias evolucionárias para a geração de redes neurais artificiais.

Inicialmente ele descarta a utilização de EEs por estas se basearem em cadeias de parâmetros de tamanho fixo.

*“Because the interest here is the acquisition of both structure and function of the recurrent neural network, evolution strategies and their reliance on fixed-length real-valued parameter strings seems an obvious mismatch.”* [10]

Quanto aos AGs, Angeline coloca que devido à sua característica de possuir dois espaços de representação distintos - o espaço de recombinações onde estão os cromossomos e onde atuam os operadores genéticos e o espaço de avaliação onde é criado o fenótipo correspondente a um cromossomo e onde atua a função de aptidão - o AG fica muito dependente da função de interpretação que mapeia o genótipo no fenótipo.

*“The dual representation of genetic algorithms is an important feature for searching in certain environments. For instance, when it is unclear how to search*



*the evaluation space directly, and when there exists an interpretation function such that searching the space of bit strings by crossover leads to good points in evaluation space, then the dual representation is ideal. It is unclear, however, that there exists an interpretation function that makes a dual representation beneficial for evolving connectionist networks. Clearly, the choice of interpretation function introduces a strong bias into the search, typically by excluding many potentially interesting and useful networks (another example of forcing the task into an architecture). Moreover, the benefits of having a dual representation hinge on crossover being an appropriate evolutionary operator for the task for some particular interpretation function; otherwise, the need to translate between an dual representations is an unnecessary complication. [10]*

Finalmente, Angeline apresenta as vantagens da utilização da programação evolucionária sobre os algoritmos genéticos para a obtenção de rede neurais artificiais, tanto a nível de estrutura quanto a nível dos valores das conexões. Ele alega que ao manipular diretamente uma representação da rede (seus nós, suas conexões e os valores destas conexões), a PE evita a necessidade de uma dupla representação e de uma função de interpretação. Uma segunda vantagem advém da manutenção da representação distribuída das redes descendentes, já que o operador de cruzamento não é utilizado.

*“Evolutionary programming provides distinct advantages over genetic algorithms when evolving networks. First, EP manipulates the network representation directly, thus obviating the need for a dual representation and the associated interpretation function. Second, by avoiding crossover between networks in creating offspring, the individuality of each network’s distributed representation is respected. For these reasons, evolutionary programming provides a more appropriate framework for simultaneous structural and parametric learning in recurrent networks.” [10]*

Angeline, então, propõe um algoritmo chamado GNARL (GeNeralized Acquisition of Recurrent Links), que é um algoritmo evolucionário baseado em programação evolucionária que através de mutações, constrói não-monotonicamente, redes neurais recorrentes capazes de resolverem uma dada tarefa.

Após analisar as duas alternativas descritas anteriormente, optou-se neste trabalho pela utilização da programação evolucionária como ferramenta para a geração de redes neurais artificiais capazes de dotar agentes autônomos com repertórios de comportamentos adequados à operação em ambientes com diferentes graus de complexidade. Assim, implementou-se um sistema chamado EVAG (*Evolutionary Agents Generator*) que possui como características principais: a capacidade de geração de diversas topologias de redes neurais, não só diretas, como também recorrentes; as re-

des possuem um grau crescente de complexidade, tanto na quantidade de neurônios como na estrutura gerada. Deste modo pretende-se simular a hipótese de evolução dos sistemas nervosos e de comportamentos observada na série filogenética [24].

Uma descrição detalhada deste sistema, os experimentos realizados, bem como os resultados obtidos são apresentados no capítulo 7.

# Capítulo 6

## Redes Neurais

### 6.1 Introdução

Atualmente, pesquisas na área de arquiteturas de computadores procuram desenvolver estruturas altamente paralelizadas a fim de otimizar os tempos envolvidos nos ciclos de busca e execução de instruções. Entre as arquiteturas mais avançadas pode-se citar as arquiteturas hiper-cúbicas, máquinas sistólicas, “data-flow”, etc. Apesar do alto grau de paralelismo presente nestas arquiteturas, todas elas se baseiam no princípio da execução de instruções para a realização do processamento desejado, adotando uma abordagem algorítmica para a solução de problemas [135]. Utilizando a nomenclatura proposta por Barreto [18] e desenvolvida por de Azevedo [59], qualquer computador que funcione baseado na execução de instruções para realização do processamento pode ser chamado de Computador Baseado em Instruções - CBI (“Instruction Based Computer” -IBC). A abordagem algorítmica utilizada nestes computadores para a solução de problemas pode ser extremamente eficiente sempre que se conheça exatamente a seqüência de instruções a serem executadas para a resolução do problema.

Por outro lado, apesar de toda a tecnologia desenvolvida em software e hardware para os computadores baseados em instruções, existe uma série de problemas que os seres vivos, e os seres humanos em particular, parecem resolver de maneira inata e que mesmo os computadores digitais mais poderosos ainda se mostram ineptos ou ineficientes para resolver. Entre estas tarefas podemos citar o processamento de imagens, o reconhecimento da fala, a recuperação de informações de maneira associativa, a filtragem adaptativa de sinais, o aprendizado de novos fatos e idéias, etc. Mesmos problemas mais corriqueiros, tais como a simples locomoção autônoma no mundo real é tarefa ainda desafiadora para ser implementada de maneira algorítmica em CBIs.

Se o sistema nervoso dos seres vivos parece ser adequado para resolver os problemas não algorítmicos e se desejamos desenvolver uma abordagem capaz de resolver esta classe de problemas, ou seja, o desenvolvimento de AAs capazes de aprender a operar em ambientes abertos, parece natural que esta abordagem procure se inspirar no funcionamento do sistema nervoso para a solução dos problemas. Como a unidade básica de funcionamento do sistema nervoso parece ser uma estrutura celular chamada neurônio e o comportamento inteligente parece emergir da interconexão entre estas células, esta nova abordagem é chamada de abordagem neural ou conexionista. Os computadores desenvolvidos utilizando esta abordagem são chamadas de Computadores Baseados em Redes Neurais - CRN ("Neural Networks-Based Computers" - NNC). Ainda segundo de Azevedo [59], é importante diferenciar Redes Neurais Artificiais (RNAs) de Computadores Baseados em Redes Neurais (CRN). A grosso modo, podemos dizer que uma RNA está para um CRN assim como o processador está para um CBI, ou seja, em termos de computador, é necessário considerar ainda todas as interfaces necessárias para se comunicar com o mundo exterior. Além disso, um CRN não é composto por apenas uma RNA, mas sim por uma série de Neurônios Artificiais (NA) que podem possuir diferentes funções de ativação e de saída e que podem, ainda, ser interconectados segundo diferentes topologias

Assim, ao longo deste capítulo procura-se, em primeiro lugar, apresentar os modelos biológicos do neurônio, seu funcionamento e a organização interna do cérebro. Em seguida são apresentados os modelos artificiais de neurônios e de RNAs, tanto estáticas como dinâmicas, bem como os mecanismos de aprendizado utilizados. Procura-se relacionar as arquiteturas de redes neurais apresentadas com a capacidade de resolução de diferentes classes de problemas. Vemos que sistemas dinâmicos necessitam de redes neurais dinâmicas para serem modelados, o que nos leva a apresentar a hipótese de que diferentes classes de comportamentos necessitam de diferentes arquiteturas de RNAs.

## 6.2 O Neurônio Biológico

Para que possamos entender melhor o funcionamento da abordagem conexionista, é necessário entender o funcionamento e a evolução do sistema nervoso dos seres vivos, e como, a partir do funcionamento deste sistema, parece emergir uma consciência inteligente.

Existe uma grande variedade de literatura disponível a respeito do funcionamento do neurônio e do sistema nervoso biológico. O leitor interessado em se aprofundar no tema pode consultar bons livros de neurofisiologia tais como [41] ou [78]. Além

disso, também uma série de artigos e livros que tratam das Redes Neurais Artificiais, possuem capítulos introdutórios descrevendo o funcionamento do neurônio e cérebro naturais, tais como [135], [30], [68], [170], [98], entre outros. Apesar disto, ainda não conhecemos e talvez nunca venhamos a compreender exatamente como a inteligência emerge desta estrutura biológica.

*“An interesting question is whether the human brain is capable to understand its own functioning. Göedel’s theorem suggests that there may be ‘ideas’ which can not be understood by the brain, assuming the brain can be described as a formal system...”[30]*

No livro de Churchland [47] a descrição e funcionamento do neurônio é colocada de forma bastante didática, e grande parte do conteúdo aqui apresentado foi extraído de lá.

Como já foi mencionado no capítulo anterior, com o surgimento dos primeiros organismos multicelulares, a aproximadamente 1 bilhão de anos atrás, a tendência observada foi no sentido da especialização de funções, pois o resultado desta forma de organização é um sistema muito mais apto do que cada um de seus componentes (“o todo é maior do que a soma das partes”) e portanto com mais chances de reproduzir e sobreviver do que os seus concorrentes unicelulares. Surgiram então organismos cujas células passaram a ter funções especializadas tais como digestão de alimentos, transporte de nutrientes para outras células, contração e alongamento para produzir movimento, sensibilidade a fatores ambientais, etc.

O funcionamento sincronizado e coordenado de uma série de órgãos que compõem os animais e mesmo a emergência de comportamentos, requer uma comunicação entre as células. Especificamente no caso de comportamentos, o sistema muscular requer que suas contrações sejam coordenadas para produzir uma locomoção, ou mastigação ou eliminação. Células sensoriais são inúteis se sua informação não puder ser transmitida para o sistema muscular e assim por diante. Felizmente, as próprias células possuem as características básicas necessárias a criação de um elo de comunicação. A maioria das células mantém uma pequena diferença de tensão - uma polarização - entre as superfícies interna e externa da membrana que envolve as células. Um determinado distúrbio em qualquer ponto da membrana pode causar uma súbita despolarização naquele ponto e se propagar por certa distância na superfície da célula. Após a despolarização, a célula procura ativamente se repolarizar. Se as células possuírem uma forma alongada, com filamentos de um metro ou mais em casos extremos, teremos elementos perfeitos para um sistema de comunicação: células nervosas especializadas capazes de conduzir impulsos eletro-químicos por longas distâncias e a altas velocidades [47].

Especializações adicionais permitem que certas células se despolarizem devido a uma mudança da pressão, outras na presença de mudanças de temperatura, outras na mudança de intensidade luminosa e outras, ainda, ao receber impulsos de outras células, formando assim os órgãos sensoriais.

Com a articulação destas células, temos o início do sistema nervoso periférico e do sistema nervoso central. E estas células alongadas, capazes de transportar impulsos elétricos são chamadas de neurônios.

Segundo Churchland [47], os neurônios podem ser classificados em duas grandes classes: 1) células externas, que conectam diferentes regiões do cérebro entre si, conectam o cérebro aos efetadores (moto-neurônios), ou conectam os receptores sensoriais (sensório-neurônios) ao cérebro; e 2) interneurônios, que são confinados a região em que eles estão presentes.

Existem centenas de tipos de neurônios, cada um com suas funções características, forma e localização. Entretanto, a estrutura básica é sempre a mesma e consiste no corpo da célula, denominado soma, os dendritos e o axônio, como pode ser visto na Figura 6.1.

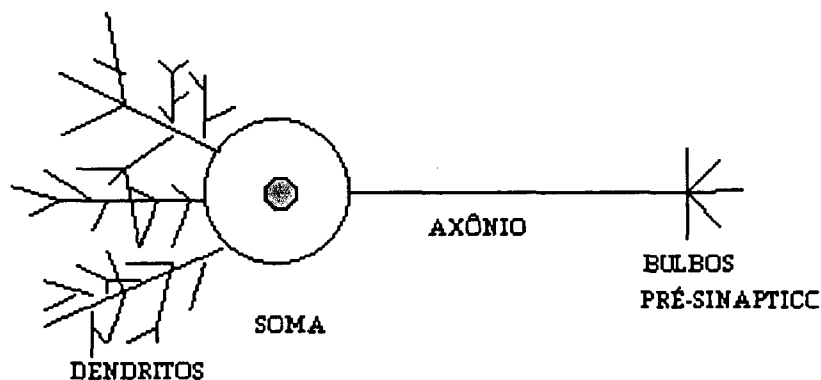


Figura 6.1: Representação simplificada de um Neurônio Biológico

O soma possui um diâmetro entre 5 e 100  $\mu\text{m}$  e contém o núcleo da célula. As funções necessárias à manutenção da vida dos neurônios, tais como síntese de enzimas e outras moléculas acontecem aqui.

Os dendritos atuam como os canais de entrada dos sinais externos recebidos pelos neurônios e o axônio atua como o canal de saída. Os axônios de vários outros neurônios fazem contato com os dendritos de um dado neurônio, ou mesmo com o corpo do neurônio. Estas conexões são chamadas de sinapses e elas permitem que uma célula influencie a atividade de outra da seguinte maneira:

Quando um pulso de despolarização - chamado potencial de ação ou “spike” - chega ao final de um axônio, ele provoca a liberação de uma substância química - chamada neuro-transmissor - contida em pequenos bulbos nas extremidades do axônio. Estes neuro-transmissores atravessam o pequeno espaço da sinapse e são recebidos pelos dendritos do neurônio seguinte. Dependendo da natureza dos neuro-transmissores liberados pelos bulbos e pela natureza dos receptores químicos que os recebem no outro lado da sinapse, a sinapse é chamada de inibitória ou excitatória.

Em uma sinapse inibitória, a transmissão sináptica provoca uma pequena hiperpolarização no potencial elétrico do neurônio afetado. Isto faz com que seja mais difícil para este neurônio se despolarizar e propagar seu próprio potencial de ação ao longo do seu axônio [47].

Ao contrário, em uma sinapse excitatória, a transmissão sináptica provoca uma pequena despolarização no neurônio afetado, rebaixando seu potencial elétrico em direção a um ponto mínimo crítico, onde ele repentinamente pode se despolarizar a ponto de iniciar a transmissão de seu próprio potencial de ação através do seu axônio.

Colocando os dois fatores juntos, cada neurônio é palco de uma competição (dispara-não-dispara) e um dos fatores que determina qual comportamento será o vencedor parece ser o número e a proximidade das sinapses excitatórias e inibitórias. Em um pequeno intervalo de tempo, estas conexões são uma característica relativamente estável para cada neurônio. No entanto, novas conexões podem crescer e outras deixarem de existir, algumas vezes em questões de minutos ou menos; deste modo, as propriedades funcionais de um neurônio são por si só plásticas, o que poderia explicar mudanças de comportamento em função do aprendizado [47].

### 6.3 Redes de Neurônios - O Cérebro

O sistema nervoso central, é uma rede que contém um grande número de neurônios interconectados. Como já foi visto, cada neurônio possui um comportamento bio-químico e bio-elétrico bastante complexo, no entanto, se acredita que o seu princípio computacional é bem simples: ele soma suas entradas e realiza uma operação de limiar e cada neurônio é capaz de ajustar sua saída como uma função relativamente simples das suas entradas. A pergunta que surge é: como pode um elemento tão simples produzir comportamentos tão complexos? A resposta parece estar na cooperação e na interação de uma grande quantidade destes elementos simples operando em paralelo.

O cérebro humano, por exemplo, possui em torno de  $10^{11}$  neurônios e cada

neurônio recebe sinais de  $10^4$  outros neurônios.

Excelentes descrições sobre a estrutura do cérebro, em particular do cérebro humano podem ser encontradas em [47] e [135].

Baseados nos fundamentos biológicos do funcionamento do cérebro e de seus neurônios, muitos pesquisadores têm tentado simular o seu funcionamento através de computadores digitais. No entanto, o grande número de neurônios e suas conexões torna praticamente impossível simular o funcionamento, por exemplo do cérebro humano, em todos os seus detalhes. Boers & Kuiper [30] calculam que a taxa de transmissão de informações no cérebro humano é da ordem de  $10^{13}$  bits por segundo e Schwartz [167] estima o número total de operações aritméticas necessárias para simular o funcionamento do cérebro humano em todos os seus detalhes em  $10^{18}$  por segundo, operando sobre uma memória de  $10^{16}$  bytes. Estes números são provavelmente um milhão de vezes maiores que o mais rápido e poderoso supercomputador a ser desenvolvido na próxima década [30].

Assim, a maioria dos pesquisadores procura desenvolver modelos que simulem apenas os aspectos considerados relevantes ou que sejam capazes de validar ou fornecer mais informações sobre teorias de funcionamento de algum aspecto do cérebro biológico. Nas seções seguintes procuramos mostrar como os mecanismos inspirados na neurobiologia foram utilizados para o desenvolvimento da maioria dos modelos de neurônios artificiais e redes de neurônios artificiais existentes hoje em dia.

## 6.4 A Abordagem Conexionista

### 6.4.1 Histórico

Apesar da abordagem algorítmica e dos CBI apresentarem, hoje em dia, um avanço tecnológico e uma cultura bem mais difundida, tanto a abordagem algorítmica quanto a conexonista apareceram praticamente na mesma época. Em 1943, Warren McCulloch e Walter Pitts estabeleceram as bases da neurocomputação, mostrando que qualquer função lógica poderia ser configurada através de um sistema de neurônios digitais interconectados (um dos primeiros computadores digitais - o ENIAC - surgiu em 1946), concebendo procedimentos matemáticos análogos ao funcionamento dos neurônios biológicos. Este desenvolvimento foi puramente conceitual, uma vez que estes autores não sugeriram aplicações práticas a partir do seu trabalho, mesmo porque os sistemas propostos por eles não tinham capacidade de aprendizado [76] [170].

Em 1949, Donald Hebb sugeriu no seu livro "Organization of Behavior", um modo de se proporcionar capacidade de aprendizado às redes neurais artificiais.



Através de experiências com animais, ele propôs que as mudanças nas forças das sinapses são proporcionais às ativações das mesmas. Este princípio, traduzido matematicamente, viabilizou o desenvolvimento de redes neurais artificiais eficazes. Esses princípios foram aplicados por Marvin Minsky na construção do Snark, o primeiro neurocomputador de que se tem notícia, em 1951. Tecnicamente ele foi um sucesso, uma vez que ajustava automaticamente os pesos entre as diversas sinapses, ou seja, demonstrava, ao menos teoricamente, que tinha capacidade de aprendizado. Contudo, este dispositivo nunca executou qualquer função útil no campo do processamento de informações, constituindo-se apenas em uma curiosidade acadêmica [76].

Em 1958, Frank Rosenblatt concebeu um dispositivo denominado de perceptron, que era uma rede neural com duas camadas de neurônios capaz de aprender de acordo com as regras propostas por Hebb. Tal aparelho tinha capacidade de treinamento supervisionado e foi utilizado com sucesso no reconhecimento de caracteres. Pela primeira vez, as redes neurais artificiais foram utilizadas com sucesso em uma aplicação prática. O êxito conseguido por esta abordagem fez com que muitos considerem Rosenblatt como o verdadeiro pai da neurocomputação.

Bernard Widrow, em 1960, desenvolveu um tipo diferente de processador para redes neurais, denominado ADALINE, o qual dispunha de uma poderosa estratégia de aprendizado. Ele ainda fundou a primeira empresa comercial para a produção de circuitos neurais digitais, a Memistor Corporation, que operou até meados da década de 1960.

Contudo, após esses espetaculares desenvolvimentos, a neurocomputação entrou em uma grande crise com a publicação do trabalho de Minsky & Papert [127] que demonstrava, de maneira formal, que o modelo de Rosenblatt era incapaz de aprender padrões não-linearmente separáveis (o famoso problema do OU-EXCLUSIVO). Minsky & Papert foram extremamente pessimistas em relação ao futuro das redes neurais artificiais.

Rosenblatt propôs como solução aumentar o número de camadas, mas, apesar de toda a sua visão e perspicácia neste campo, não logrou desenvolver um método de aprendizado eficaz para estas redes neurais mais avançadas. [76]

Apesar do descrédito gerado sobre a área da neurocomputação, entre 1967 e 1982 os estudos neste campo continuaram, ainda que englobadas em outras linhas de pesquisa, como processamento adaptativo de sinais, reconhecimento de padrões, modelamento biológico, etc. Este trabalho, ainda que silencioso, construiu as bases necessárias para que o desenvolvimento das redes neurais pudesse continuar de forma consistente.

Em 1974 aconteceu um fato que viria, mais tarde, a proporcionar o renascimento do interesse geral pelas potencialidades das redes neurais, foi quando Paul Werbos lançou as bases do algoritmo de retro-propagação ("backpropagation"), que permitia que redes neurais com múltiplas camadas apresentassem capacidade de aprendizado. Contudo, a potencialidade deste método tardou a ser reconhecida [76].

Em 1982 John Hopfield publicou um estudo que chamava a atenção para as propriedades associativas de uma classe de redes neurais que apresentava fluxo de dados multidirecional e comportamento dinâmico. Primeiramente ele mostrou que o sistema possuía estados estáveis e, posteriormente, que tais estados poderiam ser criados alterando-se os pesos das conexões entre os neurônios.

No entanto, os primeiros resultados que levaram a retomada do desenvolvimento sobre redes neurais foram publicados em 1986 e 1987, através dos trabalhos do grupo PDP (Parallel and Distributed Processing) [161] [121], onde ficou consagrada a técnica de treinamento por backpropagation. Nesta fase destacaram-se o surgimento do treinamento não-supervisionado, proposto por Teuvo Kohonen em 1984 e as novas topologias de redes neurais como a proposta por Bart Kosko, em 1987 (BAM - Bi-Directional Associative Memory)

Uma excelente fonte de referência histórica e biográfica dos principais pesquisadores envolvidos com a Abordagem Conexionista pode ser encontrada em [170], e também em [68].

#### 6.4.2 O Modelo Geral de Neurônio Artificial

A evolução das pesquisas no campo da abordagem conexionista levou ao desenvolvimento de uma infinidade de modelos de neurônios artificiais, de topologias de interconexão destes neurônios e algoritmos para aprendizado. Um trabalho que procurasse apresentar de maneira extensiva todos os modelos de neurônios, de topologias e de algoritmos de aprendizado, certamente ocuparia vários volumes e milhares de páginas. Excelentes referências para os principais modelos de RNAs, seus fundamentos, os algoritmos para sua implementação e suas aplicações características podem ser encontradas em [135], [68], [170], [161], [121].

*"Modern connectionism uses a variety of network architectures. All assume a collection of simple computational units with a set of weighted interconnections. As in perceptrons, the weights of the interconnections are modified through training to represent a concept. Modern training techniques for the weights on the interconnections take as many forms as there are architectures". [10]*

Como o objetivo principal deste trabalho é apresentar a utilização das redes neurais para a implementação de comportamentos em AAs, procuramos concentrar

nossa atenção nos modelos mais populares e apropriados para esta tarefa. Nesta seção apresenta-se um modelo formal de Neurônio Artificial e de Redes Neurais em grande parte extraído de de Azevedo [59] e Barreto [19] e baseado nos conceitos de sistema vistos anteriormente. A partir deste modelo formal genérico, pode-se derivar os diferentes modelos existentes de neurônios e redes neurais.

**Teorema 6.4.1** *O neurônio formal é um sistema dinâmico.*

**Prova:** Basta identificar as várias variáveis e funções com as da definição de sistema dinâmico. Assim, na definição de sistema dinâmico, (definição 2.3.9) se considera o seguinte objeto matemático:

$$S = \{T, U, \Omega, Y, \Gamma, X, \Phi, \eta\}$$

onde:

$T$  é o conjunto dos tempos;

$\Omega$  é o conjunto das funções de entrada  $\omega \in \Omega = \{\omega : T \rightarrow U\}$ ;

$U$  é o conjunto dos valores da entrada;

$Y$  é o conjunto dos valores da saída;

$\Gamma$  é o conjunto das funções de saída  $\gamma \in \Gamma = \{\gamma : T \rightarrow Y\}$ ;

$X$  é o conjunto dos estados;

$\Phi$  é a função de transição dos estados, neste caso também chamada de função de ativação:  $\Phi : T \times T \times X \times \Omega \rightarrow X$ ;

$\eta$  é a função de saída:  $\eta : T \times X \times U \rightarrow Y$ .

Identificando com as variáveis e funções do neurônio tem-se:

$T$  conjunto dos tempos é geralmente conjunto dos inteiros,

$\Omega$  o conjunto das funções de entrada ou provenientes dos órgãos sensores ou da saída de outros neurônios;

$\Gamma$  é o conjunto das funções de saída, cujos valores são representativos das frequências de descarga dos neurônios;

$U$  é o conjunto dos valores da entrada, valores representativos das saídas dos órgãos sensores ou dos valores de frequências de descarga de outros neurônios

$Y$  é o conjunto dos valores da saída, valor representativo da frequência de descarga do neurônio;

$X$  é o conjunto dos estados possíveis e representa a excitação do neurônio;

$\Phi$  é a função de ativação do neurônio:  $\Phi : T \times T \times X \times \Omega \rightarrow X$ , e mostra como a excitação do neurônio evolui;

$\eta$  é a função de saída:  $\eta : T \times X \times U \rightarrow Y$ , que freqüentemente é tomada como uma função cujo valor é sempre o valor de excitação do neurônio.

Tendo identificado todas as variáveis e funções envolvidas na definição de neurônio, pode-se dizer que o modelo de neurônio é um exemplo de sistema dinâmico.

A partir da definição formal para a representação de um neurônio artificial é possível descrever o funcionamento de diversos modelos de neurônios, bastando particularizar os parâmetros que definem o sistema. Particular atenção será dada à escolha da função de transição de estados e na função de saída e na maneira de combinar os valores de entrada dos neurônios.

### Combinação dos valores de entrada

De maneira genérica, os elementos básicos que compõem um NA podem ser vistos na Figura 6.2.

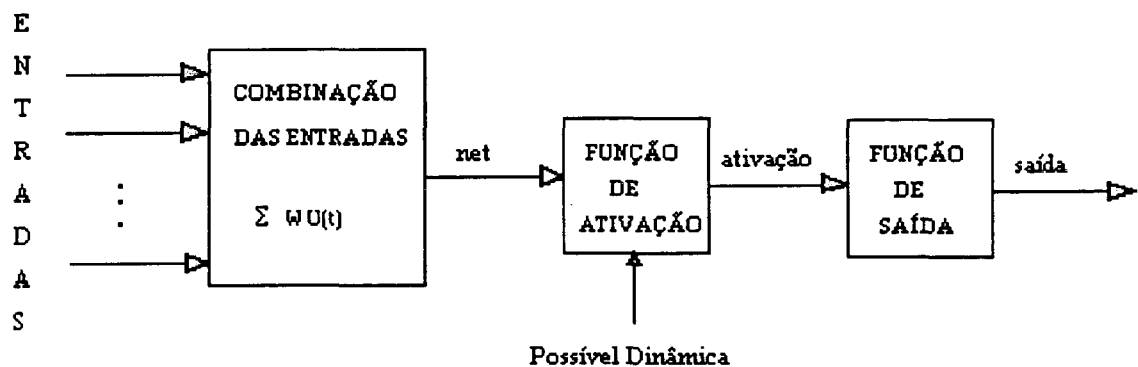


Figura 6.2: Elementos básicos que compõem um Neurônio Artificial

Como pode ser visto na figura, ao nível de rede, cada entrada do neurônio possui um conjunto de pesos associados, representado pelos rótulos do arco no modelo de Grafo que representa a RNA, como veremos a seguir. Este conjunto de pesos pode ser entendida como a “força” da conexão sináptica quando comparada com o neurônio biológico. As entradas de um neurônio podem ser as saídas de outros neurônios, entradas externas, um bias ou qualquer combinação destes elementos. O somatório de todas estas entradas dá origem ao chamado “net” de um neurônio.

É importante ressaltar que no formalismo apresentado, os pesos que representam a força de uma conexão sináptica entre dois neurônios, não fazem parte do neurônio

artificial formal, mas sim, são representados pelos rótulos do Grafo que forma a Rede de Neurônios. O que significa dizer que os pesos só possuem sentido prático quando um neurônio se conecta a outro, exercendo sobre este uma ação que é "modulada" pelo valor do peso.

Na notação usualmente adotada na literatura é comum encontrarmos todas as entradas de um neurônio incorporadas sob a representação  $u_k$  e os pesos por  $w_{ij}$  com  $j=1$  até  $n$ , A equação para o "net" fica então:

$$net_i(t) = \sum W_{ij}u_j(t) \quad (6.1)$$

■

onde  $w_{ij}$  é um número real que representa a conexão sináptica da entrada do  $i^{ésimo}$  neurônio com a saída do  $j^{ésimo}$  neurônio. A conexão sináptica é conhecida como excitatória se  $w_{ij} > 0$  ou inibitória caso  $w_{ij} < 0$ .

Após a determinação do  $net_i$ , o valor da ativação do neurônio é atualizado através da função de ativação e finalmente, o valor de saída do neurônio é produzido através da função de saída.

Analicamente, a ativação e a saída dos neurônios são dados pelas seguintes relações:

$$x(t+h) = \Phi(x(t), net(t), t, h) \quad (6.2)$$

■

$$y(t) = \eta(x(t), net(t), t) \quad (6.3)$$

■

A partir destas relações formais, podemos obter diferentes modelos a partir de modificações na função de ativação e na função de saída .

### A função de ativação $\Phi$

Examinando as equações anteriores podemos ver que os estados futuros de um neurônio são afetados pelo estado atual do neurônio e pelo valor do net de entrada. Este tipo de neurônio, que possui "memória" é conhecido como "neurônio dinâmico". Por outro lado, se considerarmos a função como constante, teremos neurônios que não possuem "memória", ou seja, o estado atual é igual aos estados anteriores e portanto o neurônio é conhecido como "neurônio estático".

### A função de saída $\eta$

Essencialmente, qualquer função contínua e monotonicamente crescente tal que  $x \in \mathfrak{R}$  e  $y(x) \in [-1, 1]$  pode ser utilizada como função de saída na modelagem neural. Existem, no entanto, uma série de funções mais comumente utilizadas como funções de saída em neurônios. Estas funções são:

A função linear

$$y(x) = ax \quad (6.4)$$

□

A função logística, que é a função unipolar mais popular

$$y(x) = \frac{1}{1 + e^{-kx}} \quad (6.5)$$

□

onde  $k$  é um fator de escala positivo.

A função tangente hiperbólica, que é a função bipolar mais popular

$$y(x) = \tanh(kx) = \frac{1 - e^{-kx}}{1 + e^{-kx}} \quad (6.6)$$

Obs.: É comum encontrarmos na literatura a descrição apenas de neurônios estáticos. Assim, em alguns casos podemos encontrar o termo função de ativação como sendo a função que converte o valor do “net” no valor de saída do neurônio. Ao longo deste texto, procuramos usar de maneira coerente o termo função de saída como sendo a função que converte o valor de ativação do neurônio no seu valor de saída. No entanto, o leitor deve ficar atento, pois nem sempre a literatura é consistente a este respeito.

### 6.4.3 Redes de Neurônios Artificiais

Para definir uma rede neural podemos utilizar a mesma definição que foi utilizada para um único neurônio. Neste caso, no entanto, nos defrontamos com uma questão: com nossas definições anteriores não é possível representar de maneira explícita os pesos das conexões entre os diferentes neurônios da rede. Para resolver este problema, podemos considerar uma rede neural como um Sistema Dinâmico Complexo

segundo a definição 2.3.16 apresentada na seção 2.3.1. De modo a facilitar a representação de redes neurais como um sistema dinâmico complexo, é útil introduzir alguns conceitos de Teoria dos Grafos.

**Definição 6.4.1** *Um Grafo  $G$  consiste de um conjunto finito não vazio de vértices  $V = v_i$  e de um conjunto não-ordenado de arcos  $A$  que conectam alguns pares de vértices. Cada par  $v_i, v_j$  de vértices em  $V$  é um arco de  $G$  que “liga”  $v_i$  e  $v_j$ .*

**Definição 6.4.2** *Um Grafo  $G$  é dito Rotulado quando os  $p$  vértices são distinguidos uns dos outros através de nomes ou rótulos.*

**Definição 6.4.3** *Um Grafo  $G$  é dito Arco-Rotulado quando também os arcos são distinguidos uns dos outros através de nomes ou rótulos.*

Uma vez estabelecida a definição de Grafo, é possível definir uma Rede Neural Artificial.

**Definição 6.4.4** *Uma Rede Neural Artificial (RNA) é um Sistema Dinâmico Complexo representado por um grafo arco-rotulado no qual cada vértice é um Sistema Dinâmico denominado Neurônio Artificial.*

Uma vez definido formalmente o conceito de Rede Neural, podemos definir vários tipos de redes neurais, seguindo as definições apresentadas na seção 2.3.1.

**Definição 6.4.5** *Uma Rede Neural Contínua no Tempo é uma rede neural definida em um subconjunto do tempo  $T = \mathbb{R}$ .*

**Definição 6.4.6** *Uma Rede Neural Discreta no Tempo é uma rede neural definida em um subconjunto do tempo  $T = \mathbb{Z}$ .*

**Definição 6.4.7** *Uma Rede Neural Invariante no Tempo, também chamada de Rede Neural Estacionária é uma rede neural em que a função de transição de estados  $\Phi$  depende de apenas um elemento de  $T$  e a função de saída  $\eta$  é independente de  $T$ .*

## 6.5 Topologias de RNAs

A maneira como os neurônios estão interconectados e as características dinâmicas dos neurônios formam as diferentes topologias de RNAs e estão intimamente relacionadas com o algoritmo de aprendizado utilizado para ensinar a rede e com a capacidade da rede para resolver certas classes de problemas. Nesta seção serão apresentadas as topologias mais comuns, visando sua utilização nas seções e capítulos seguintes, onde são mencionadas em teoremas e utilizadas em exemplos.

### 6.5.1 Redes Diretas

Redes Diretas (“Feedforward”) são aquelas cujo grafo não tem ciclos. Frequentemente é comum representar estas redes em camadas e neste caso são chamadas *redes em camadas*. Neurônios que recebem sinais de excitação são chamados da *camada de entrada*, ou primeira camada. Neurônios que têm sua saída como saída da rede pertencem a *camada de saída* ou última camada. Neurônios que não pertencem nem a camada de entrada nem a de saída são neurônios internos à rede e pertencem a uma ou mais camadas internas (“hidden layers”).

Estas redes são atualmente as mais populares, principalmente por existirem métodos de aprendizado fáceis de usar para estas redes. Um método bastante usado, mas muito ineficiente é a “backpropagation”. Por esta razão alguns autores chegam mesmo a chamar, impropriamente, este tipo de rede de rede “backpropagation”. Além disto, estas redes são capazes de aproximar, com maior ou menor precisão, dependendo do número de neurônios da rede, qualquer função não linear [51]. Entretanto, mesmo no caso de usarem neurônios dinâmicos (equação diferencial de primeira ordem ou a uma diferença finita), tem uma dinâmica muito limitada não podendo representar todos os sistemas dinâmicos [19].

#### Rede com uma única camada (“single-layer”)

Esta é uma das formas mais simples de redes em camadas. Aqui temos apenas uma camada de neurônios de entrada que se conectam com os neurônios da camada de saída. Esta rede é chamada de “single-layer” referindo-se apenas aos neurônios da camada de saída, pois como a função dos neurônios da camada de entrada é apenas de bufferização, eles não são considerados [79].<sup>1</sup>

A grande limitação deste tipo de topologia é a impossibilidade, demonstrada por Minsky & Papert, de se resolver problemas classificados como linearmente não-separáveis.

#### Redes multi-camadas (“multilayer”)

A segunda classe de redes neurais diretas se caracterizam pela presença de uma ou mais camada de neurônios internos. A função dos neurônios internos é extrair características dos neurônios da camada de entrada e fornecê-las como entrada para os neurônios da camada de saída. Deste modo, a rede é capaz de extrair estatísticas

---

<sup>1</sup>Pode-se encontrar alguns trabalhos que se referem a “camada” como sendo a camada de conexões. Neste trabalho nos referiremos sempre à camada como sendo camada de neurônios, e consideraremos os neurônios de entrada apenas como elementos de bufferização.



de mais alta ordem. Um exemplo de rede multi-camada pode ser visto na figura a seguir [79].

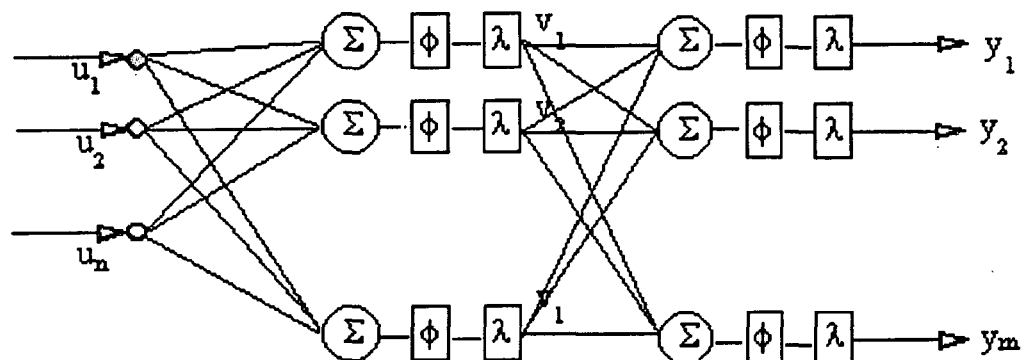


Figura 6.3: Rede Neural Artificial Direta Multi-Camada

Hoje em dia é fato bem conhecido que “redes diretas multi-camadas com funções de saída não lineares do tipo sigmoideal, termos de ‘bias’ e pesos de conexões ajustáveis são capazes de aproximar, com a precisão que se deseje, qualquer função mensurável” [52].

### 6.5.2 Redes Recorrentes

Redes com ciclos (ou com realimentação, ou com retroação, ou com “feedback”) são aquelas cujo grafo de conectividade contém ao menos um ciclo. Quando além disto envolvem neurônios dinâmicos, são chamadas *recorrentes*.

Uma rede recorrente pode ser composta por uma única camada de neurônios em que cada neurônio fornece o seu sinal de saída como entrada para cada um dos outros neurônios, como mostrado na Figura 6.4. A estrutura descrita nesta figura não possui camada escondida. Além disso os laços de realimentação envolvem a utilização de ramos particulares compostos por elementos de atraso-unitário (denotados por  $z^{-1}$ , o que resulta em um comportamento dinâmico não-linear em virtude da natureza não-linear dos próprios neurônios [79]. Esta característica de dinâmica não-linear da estrutura permite que este tipo de rede neural seja utilizado em problemas e aplicações que exijam representação de estados, tais como: processamento de voz, controle industrial, processamento adaptativo de sinais, predição de séries temporais, etc. [137]. Hoje em dia, as redes neurais recorrentes têm sido objeto de extensa pesquisa e num futuro próximo, acredita-se, suplantará em utilização as

redes neurais diretas, justamente pela sua capacidade de representação de sistemas dinâmicos, inexistente nas rede diretas com neurônios estáticos.

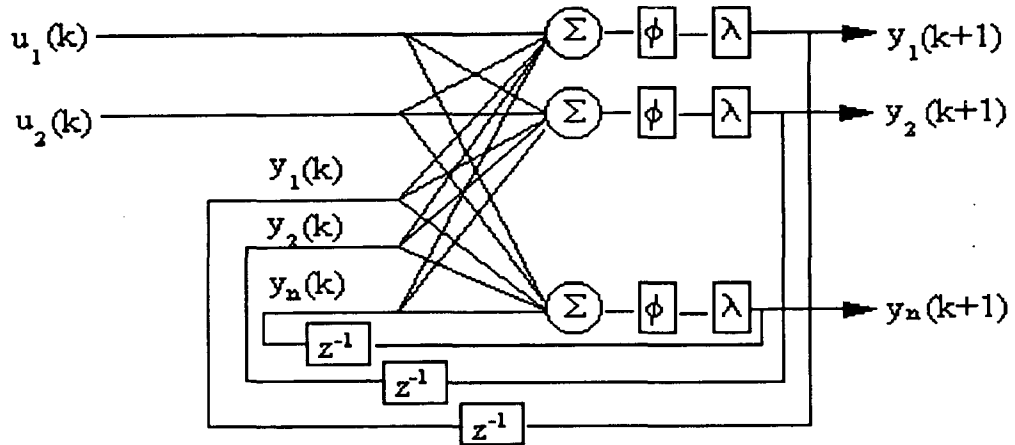


Figura 6.4: Rede Neural Artificial Recorrente com uma única camada de neurônios

Farta bibliografia já pode ser encontrada a respeito deste tipo de topologia, cada autor apresentando modelos e soluções diferentes. Uma referência inicial sugerida é o artigo de Giles, Kuhn & Williams [72] onde bons fundamentos teóricos podem ser encontrados. O leitor interessado em se aprofundar neste tema pode consultar [43], [46], [48], [71], [95] entre outros (existe uma Home Page na Internet mantida por Giles apenas com bibliografia na área de redes neurais recorrentes, no seguinte endereço:

<http://www.neci.nj.nec.com/homepages/giles/bibliographies/rnn.bib>).

### Redes recorrentes de tempo-real

Um dos modelos de redes neurais recorrentes mais interessantes e sobre o qual se desenvolverá grande parte do trabalho restante nesta proposta é um modelo proposto por Williams & Zipser [194] e que também é apresentado por Haykin [79] e por Saxén [163] chamado Rede Recorrente de Tempo Real (“Real-Time Recurrent Network”) e que será descrito a seguir.

Considere-se uma rede consistindo de um total de  $Z$  neurônios com  $M$  conexões de entradas externas. Seja  $x(k)$  um vetor de dimensões  $M$ -por-1 de sinais externos aplicado aos neurônios de entrada da rede no instante  $k$ , e seja  $y(k+1)$  um vetor

de dimensões  $N$ -por-1 correspondente à saída obtida nos neurônios de saída no instante  $k + 1$ . O vetor de entrada  $x(k)$  e o vetor de saída retardado por um passo são concatenados para formar o vetor  $u(k)$  de dimensão  $(M + N)$ -por-1, cujo  $i^{\text{ésimo}}$  elemento é representado por  $u_i(k)$ .

Podemos distinguir duas camadas distintas nesta rede, chamadas de *camada de entrada-saída concatenada* e *camada de processamento*, como pode ser visto na Figura 6.5. Neste caso, a rede possui  $M = 2$  e  $N = 4$ . A rede é completamente interconectada de modo que existem  $MN$  conexões diretas e  $N^2$  conexões realimentadas. Seja  $W$  uma matriz de dimensões  $N$ -por- $(M + N)$  chamada *matriz de pesos recorrentes da rede*.

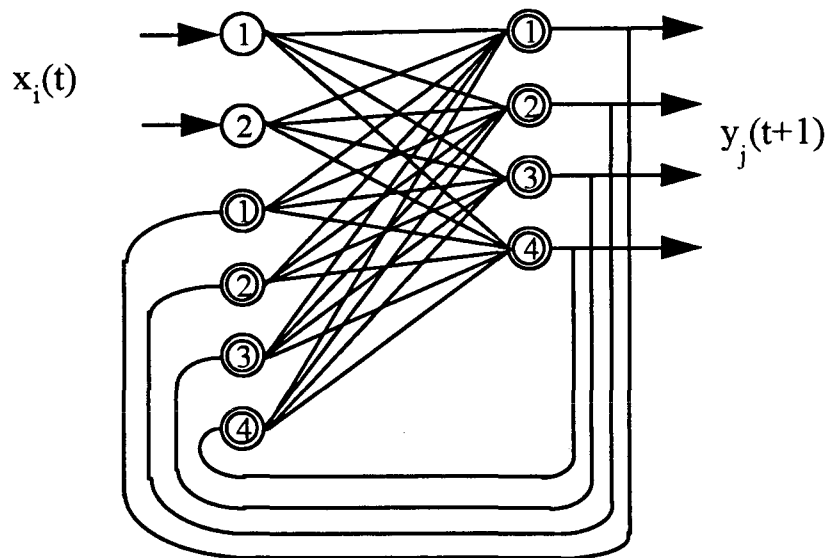


Figura 6.5: Um exemplo de rede recorrente de tempo-real

O valor do “net” para um determinado neurônio  $j$  no instante  $k$ , sendo que  $j$  corresponde a um neurônio da camada de processamento, é dado por:

$$net_j(k) = \sum w_{ji}(k)u_i(k) \quad (6.7)$$

O valor da saída do neurônio  $j$  no instante de tempo  $k + 1$  é obtido passando-se o “net” pela função de ativação (que introduz um retardo puro) e pela função de saída  $\eta$ , obtendo-se:

$$y_j(k + 1) = \eta(net_j(k)) \quad (6.8)$$

Saxém [163] implementou também uma ferramenta para modelagem e simulação utilizando este modelo de rede recorrente.

O capítulo seguinte aponta algumas limitações que podem advir deste modelo e propõe algumas extensões capazes de aprender autômatos finitos a fim de implementar comportamentos.

## 6.6 Alguns Teoremas

Com base no que foi visto até este momento, pode-se nesta seção apresentar uma série de teoremas propostos por Barreto [19] e que servirão como base para análise sobre as potencialidades computacionais das redes neurais e conseqüentemente dos tipos de comportamentos e os respectivos modelos de redes neurais capazes de implementá-los que será feita no próximo capítulo. Alguns dos teoremas aqui apresentados voltarão a ser examinados com maiores detalhes na seção 7.4. Aqui estes teoremas serão apenas apresentados, o leitor interessado em analisar as provas desenvolvidas para os teoremas é convidado a consultar [19].

**Teorema 6.6.1** *Toda RNA constituída apenas de neurônios estáticos, incluindo ciclos (ou seja, com retroação), é equivalente a uma outra rede estática sem ciclos.*

**Teorema 6.6.2** *Uma rede estática, direta, usada isoladamente, é incapaz de aproximar um sistema dinâmico.*

**Teorema 6.6.3** *Uma rede direta, dinâmica, (ou seja, com neurônios dinâmico) munida de uma retroação externa, é capaz de aproximar um sistema dinâmico.*

**Teorema 6.6.4** *Uma rede direta munida de um conjunto de retardos é capaz de aproximar um sistema dinâmico.*

## 6.7 Aprendizado de RNAs

Se nos CBIs, o problema a ser solucionado está em codificar de maneira adequada o algoritmo para a realização do processamento desejado (o que no caso da implementação de AAs parece ser tarefa impossível), o problema do aprendizado em redes neurais pode ser resumido como o sendo o problema de achar um conjunto de pesos que permitam à rede realizar o processamento desejado.

No caso das redes neurais, todos os métodos de aprendizado podem ser classificados em uma de duas categorias: aprendizado supervisionado e aprendizado

não-supervisionado, embora algumas arquiteturas possam incorporar ambas as categorias.

O aprendizado supervisionado é um processo que incorpora um professor externo e/ou uma informação global sobre a performance da rede. Técnicas de aprendizado supervisionado incluem decisões como: decidir quando terminar a fase de aprendizado, decidir por quanto tempo e com que frequência apresentar as associações a serem treinadas, e decidir como fornecer as informações sobre a performance ou erro cometido pela rede.

As redes que utilizam o aprendizado não-supervisionado são também conhecidas como redes auto-organizadas ("self-organizing"). Este processo de aprendizado não envolve um professor externo e se baseia apenas em informações locais e controle interno. O aprendizado não-supervisionado auto-organiza os pesos dos neurônios em função dos dados apresentados, tentando descobrir suas propriedades coletivas emergentes e separá-los em classes de padrões [59].

Como já foi mencionado anteriormente, em 1949 o psicólogo Donald O. Hebb apresentou uma proposta, hoje em dia conhecida como Regra de Hebb, que pode ser considerada a base para todas as regras de aprendizado existentes atualmente. Em seu livro intitulado *The Organization of Behavior* [80], ele postulou:

*"Quando um axônio de um neurônio A está próximo o suficiente para excitar um neurônio B, e repetidamente ou persistentemente toma parte do disparo de B, então, ocorre um certo processo de crescimento ou mudança metabólica em uma das duas células, de forma que a eficiência de A em contribuir para o disparo de B é aumentada."*

Em um livro mais recente, chamado 'Textbook of Psychology' (Psicologia na versão traduzida para o português [81]) Hebb trata também da questão da extinção ou esquecimento de respostas aprendidas, ou seja, se eventualmente o neurônio A passa a não tomar mais parte do disparo do neurônio B, então ocorre uma extinção da relação aprendida e que pode ser temporária ou duradoura [81].

*"A resposta extinguida pode ser uma bem firmada, já há muito consolidada, de modo que o processo não é uma inversão das mudanças sinápticas da aprendizagem. Parece, porém, que as tentativas de extinção devem acrescentar certa aprendizagem posterior que modifica ou inibe a aprendizagem inicial."*

A formulação matemática da regra de Hebb, contando inclusive com um termo que modela o esquecimento pode ser encontrada em [79], sendo bastante utilizada como algoritmo de aprendizado biologicamente inspirado para o treinamento de redes neurais que controlam robôs móveis autônomos, utilizando o princípio do condicionamento clássico. Trabalhos envolvendo implementações deste algoritmo podem

ser vistos em [53] e [63].

### 6.7.1 Aprendizado em Redes Neurais Diretas - o algoritmo 'Backpropagation'

O algoritmo Backpropagation foi desenvolvido de maneira independente por vários pesquisadores trabalhando em diferentes áreas aproximadamente na mesma época. Em 1974, Werbos [191] descobriu o algoritmo enquanto desenvolvia sua tese de doutorado em estatística e o chamou de "Algoritmo de Realimentação Dinâmica". Parker [138] em 1982 redescobriu o algoritmo e chamou-o de "Algoritmo de Aprendizado Lógico". Porém, foi com o trabalho de Rumelhart, Hinton & Williams do grupo PDP do MIT [161], que em 1986 divulgou e popularizou o uso do Backpropagation para o aprendizado em redes neurais. O algoritmo Backpropagation é hoje em dia a técnica de aprendizado supervisionado mais utilizada para redes neurais multi-camada direta.

Uma descrição sumária da operação da rede é apropriada neste instante para ilustrar como o Backpropagation é utilizado para o aprendizado de problemas de mapeamento complexo. Basicamente, a rede aprende um conjunto pré-definido de pares de exemplos de entrada/saída em ciclos de propagação/adaptação. Depois que um padrão de entrada foi aplicado como um estímulo aos elementos da primeira camada da rede, ele é propagado por cada uma das outras camadas até que a saída seja gerada. Este padrão de saída é então comparado com a saída desejada e um sinal de erro é calculado para cada elemento de saída.

O sinal de erro é então retro-propagado da camada de saída para cada elemento da camada intermediária anterior que contribui diretamente para a formação da saída. Entretanto, cada elemento da camada intermediária recebe apenas uma porção do sinal de erro total, proporcional apenas à contribuição relativa de cada elemento na formação da saída original. Este processo se repete, camada por camada, até que cada elemento da rede receba um sinal de erro que descreva sua contribuição relativa para o erro total. Baseado no sinal de erro recebido, os pesos das conexões são, então, atualizados para cada elemento de modo a fazer a rede convergir para um estado que permita a codificação de todos os padrões do conjunto de treinamento.

A formalização matemática do algoritmo Backpropagation pode ser encontrado em várias bibliografias, recomendamos ao leitor a descrição do algoritmo original descrito por Rumelhart, Hinton e Williams em [161].

### 6.7.2 Aprendizado em Redes Neurais Recorrentes

Da mesma maneira que para as redes neurais diretas com neurônios estáticos, nas redes neurais recorrentes o objetivo do aprendizado é determinar um algoritmo adaptativo ou regra que ajuste os pesos da rede baseado na relação entre os pesos de entrada/saída apresentados durante o treinamento.

O aprendizado em redes neurais recorrentes é tarefa bem mais complexa que o aprendizado em redes diretas e uma série de propostas têm sido apresentadas como métodos para o treinamento deste tipo de rede [183], [42], [64], desde o aprendizado aleatório, passando pelo backpropagation através do tempo (“backpropagation through time”) mencionado por Rumelhart e McClelland [161] e aperfeiçoado por outros pesquisadores [192], o método de Levenberg-Marquadt utilizado por Saxén [163], o backpropagation dinâmico (“dynamic backpropagation”) proposto por Narendra [134] ou ainda o algoritmo de aprendizado supervisionado de tempo-real temporal (“real-time temporal supervised learning algorithm”) desenvolvido por Williams & Zipser [194], sendo ainda um campo aberto para pesquisas.

No próximo capítulo propõe-se um algoritmo de aprendizado baseado no algoritmo backpropagation para aprendizado de autômatos finitos em redes recorrentes, bem como um algoritmo de aprendizado biologicamente inspirado, baseado no princípio de tentativa-e-erro, que pode ser utilizado tanto em redes diretas como em redes recorrentes.

## Capítulo 7

# Agentes Autônomos e Paradigmas Biológicos

### 7.1 Introdução

Nos capítulos anteriores procurou-se reunir subsídios sobre diversos paradigmas biológicos considerados importantes e inspiradores para se atingir um dos objetivos primordiais deste trabalho, a busca de ferramentas inspiradas na Natureza que possam lançar novas idéias e facilitar o desenvolvimento de AAs mais úteis, eficientes e robustos. Ao mesmo tempo em que estes paradigmas naturais eram estudados, procurou-se, também, analisar que paradigmas computacionais, inspirados nestes paradigmas naturais, poderiam ser utilizados ou desenvolvidos bem como, de que modo estes paradigmas computacionais deveriam ser empregados.

Assim, ao longo deste estudo, diversas idéias e reflexões foram surgindo e amadurecendo, resultando no desenvolvimento de uma série de sistemas e ferramentas computacionais baseadas nos paradigmas biológicos e orientadas para auxiliar na implementação de AAs. Além disso, diversas experiências foram feitas visando testar, ilustrar e aperfeiçoar estas idéias.

O objetivo deste capítulo é apresentar e descrever estas idéias, implementações e experiências, procurando mostrar como elas podem ser utilizadas na tarefa de desenvolvimento de AAs.

### 7.2 Idéias Gerais

Como já foi visto nos capítulos anteriores, o objeto de descrição de um AA é o seu *comportamento*, ou seja, como ele transforma as informações recebidas pelos seus sensores e que refletem um estado observável do ambiente, em comandos para os



seus atuadores, que eventualmente farão com que este agente se movimente e aja no ambiente, modificando-o a fim de cumprir uma determinada tarefa para a qual foi projetado.

Viu-se também que existem *classes de comportamentos* que estão relacionadas basicamente com o tempo e “grau” de importância de um estímulo externo percebido pelos sensores, em provocar uma reação dos atuadores. Esta relação pode variar desde a extremamente direta e imediata (o comportamento reflexivo) até a extremamente indireta e retardada (o comportamento racional) passando por vários graus intermediários em que eventos anteriores e fatores hormonais internos estão presentes. A entidade responsável pela geração do comportamento é o sistema nervoso central e o tipo de comportamento que pode ser gerado está diretamente relacionado com a forma como se estruturam as conexões entre os neurônios que compõem o sistema nervoso. Assim, nas formas mais primitivas de comportamento, o conjunto de neurônios que os implementam não são capazes de continuar sua atividade após cessada a estimulação externa. A medida que os neurônios se interligam através de conexões reentrantes formando arcos fechados, o sistema é capaz de continuar sua atividade mesmo após cessada a estimulação, gerando comportamentos mais complexos. Além disso, um sistema de neurônios pode ser excitado por outro sistema de neurônios sem a presença de um fato sensorial externo, residindo aí a base do processo de pensamento e raciocínio. “Não é necessário, por exemplo ver o alimento dentro de uma panela para pensar no alimento [81]”.

A Figura 7.1, inspirada em [81], mostra um diagrama das relações entre neurônios efetivamente observados em um sistema nervoso biológico. O axônio entrante excita os dendritos de quatro neurônios, A, B, C e D. Dentre estes, B e C enviam impulsos para fora do sistema para excitar outros sistemas neuronais ou musculares, mas a partir de A e D são enviados impulsos apenas dentro do próprio sistema. A-B, B-E e B-E-E’ formam circuitos fechados que podem reter excitações e fazer com que B continue enviando impulsos para fora do sistema [81]. Estes circuitos fechados podem representar, então, estados cognitivos, imaginações e pensamentos que fazem com que o comportamento se torne bastante complexo e dependa apenas de forma indireta de um estímulo externo.

Observou-se ainda, que os comportamentos podem ser descritos através de *Autômatos de Estados Finitos* (AEFs). Assim, ao implementar um AA, uma das maneiras de descrever o funcionamento desejado para este agente é através da utilização de AEFs com vários níveis de detalhamento.

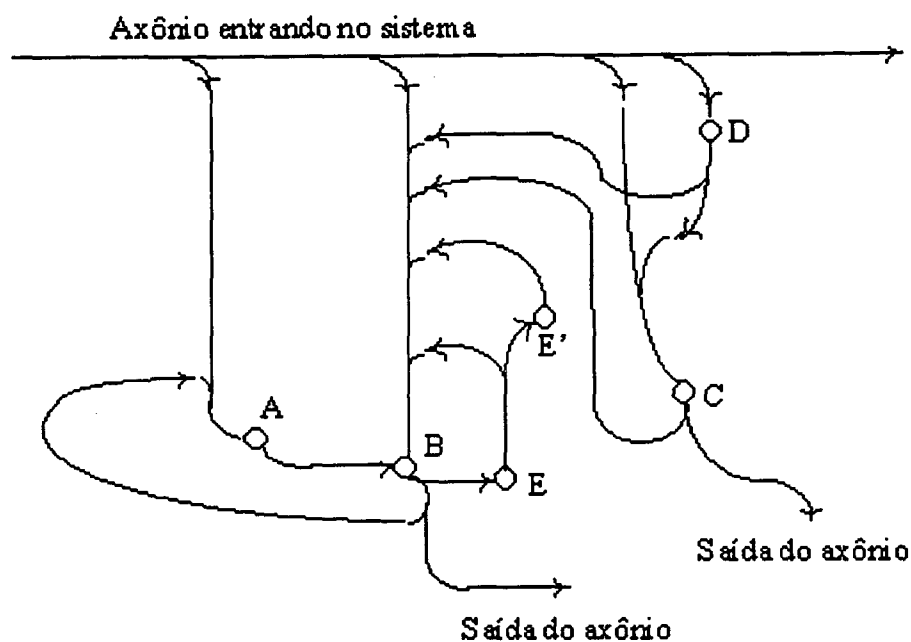


Figura 7.1: Diagrama das relações entre neurônios efetivamente observados em um sistema nervoso biológico

Se nos organismos naturais a entidade responsável por implementar os comportamentos é o sistema nervoso central, parece natural pensar-se na utilização da *abordagem conexionista*, através de *Redes Neurais Artificiais* (RNAs) para a implementação de comportamentos em sistemas artificiais. Deve-se procurar, então, uma maneira de fazer com que estas RNAs implementem os AEFs que descrevem os comportamentos necessários. Neste aspecto, então, se concentra grande parte das idéias e experiências deste capítulo.

Outra maneira de se implementar comportamentos em AAs e que se encontra de forma abundante na Natureza é através da descrição pura e simples do Comportamento Observável que se deseja do agente. Ou seja, para o agente apresenta-se apenas um sistema descrito ao nível comportamental contendo os pares de entrada-saída desejados e permite-se que o sistema induza um sistema dinâmico compatível para a implementação deste sistema comportamental.

Por fim, demonstrou-se também que a complexidade do comportamento apresentado varia com o grau de complexidade do sistema nervoso central, ou seja, comportamentos simples podem ser implementados com redes neurais simples (poucos neurônios e apenas conexões diretas), enquanto comportamentos complexos necessitam redes neurais mais complexas (muitos neurônios e grande número de conexões formando ciclos).

A complexidade dos sistemas nervosos nos seres vivos foi determinada através de processos evolucionários que se estenderam por milhões de anos; quanto mais complexo o ambiente se tornava, maior o repertório e a complexidade dos comportamentos necessários para sobreviver naquele ambiente. Como já sugerido por Vaario [184], o processo evolucionário pode ser encarado como uma ferramenta para ir aumentando gradativamente a complexidade do sistema, ou seja, seu repertório de comportamentos. A medida que a complexidade do sistema, e em especial de seus comportamentos for aumentando, aumenta também a possibilidade de sobrevivência do sistema.

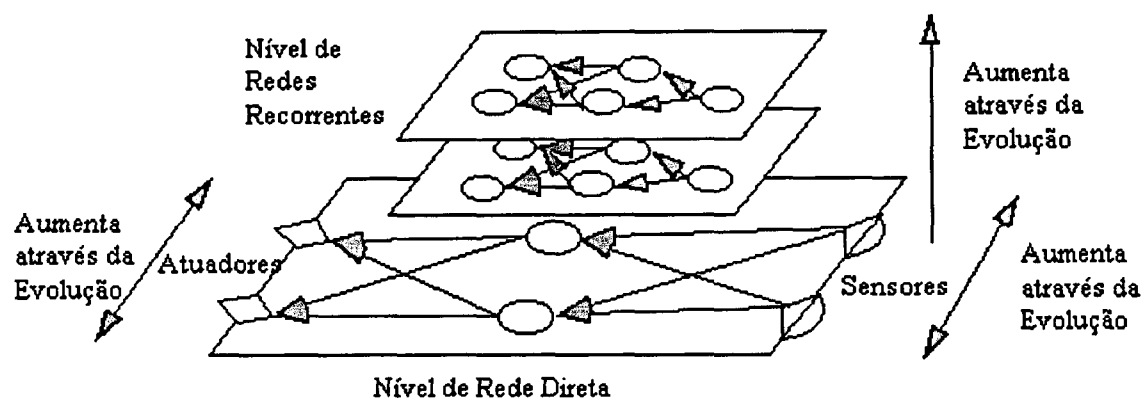


Figura 7.2: Aumento da complexidade do sistema nervoso através de processos evolucionários

Assim, dado um ambiente onde um AA deva operar, pode-se através dos paradigmas da *computação evolucionária* selecionar a quantidade ótima de recursos (em termos de topologia e quantidade de neurônios) da RNA capaz de implementar o repertório de comportamentos necessários para a “sobrevivência” e operação do AA no ambiente determinado.

### 7.3 Capacidade de Generalização de RNAs

Ao se estudar as características de comportamentos e aprendizado dos animais visando sua aplicação no desenvolvimento de AAs, um fato que primeiro chama a atenção é a questão do dilema entre a especialização e a generalização por eles apresentada e como este dilema se manifesta ao se implementar os correspondentes paradigmas artificiais, seja através de RNAs ou outro método computacional.

A característica de especialização se refere à capacidade de um sistema de fornecer respostas precisas e acuradas para as questões apresentadas. Especificamente no caso deste trabalho, se refere à capacidade de um animal (natural ou artificial) de fornecer as saídas mais precisas possíveis para os atuadores em resposta a um dado estímulo percebido pelos sensores, na implementação de um dado comportamento. Já a generalização é a capacidade de um sistema de fornecer respostas corretas (não necessariamente precisas) para as questões dadas, sendo que estas questões apresentadas não foram vistas anteriormente durante o aprendizado [151] [14] [6] [131].

A generalização pode, por sua vez, ser subdividida nas capacidades de interpolação e de extrapolação. Por interpolação entende-se a capacidade de fornecer respostas corretas a questões que se situam dentro do intervalo de domínio das questões apresentadas durante o aprendizado. A extrapolação é a capacidade de responder corretamente a questões que fazem parte do contexto mas que estão fora do intervalo de domínio das questões vistas durante o aprendizado [154].

Os conceitos de especialização e generalização estão muito presentes quando examina-se as várias classes de comportamento exibidas pelos animais. Alguns tipos de comportamentos estereotipados são extremamente especializados, por exemplo o comportamento alimentar do Urso Panda que se alimenta apenas de folhas de uma variedade de bambus, ou rituais de acasalamento apresentados por várias espécies de insetos. Se por um lado esta especialização auxilia na sobrevivência do indivíduo em condições ambientais ideais, por outro lado parece dificultar a sua sobrevivência quando o ambiente sofre alterações. Parece que quanto mais especializado for o comportamento, menos plástico ele é, ou seja, mais difícil se torna qualquer modificação no comportamento por um processo de aprendizado qualquer.

Já animais cujo repertório de comportamentos é menos estereotipado, possuem uma capacidade de generalização maior. Esta capacidade de generalização permite uma maior plasticidade dos comportamentos apresentados que podem ser modificados através de algum processo de aprendizado. Esta capacidade de generalização traz uma aptidão maior para a sobrevivência da espécie em função de alterações ambientais [154].

O ser humano é o ser vivo mais generalista que se tem conhecimento, senão vejamos, ele não é especializado para realizar nenhuma tarefa específica que eventualmente pudesse facilitar a sua sobrevivência em ambientes hostis, tais como correr, saltar, subir em árvores, lutar, farejar, não possui alta acuidade visual nem auditiva. No entanto, a sua capacidade de generalização o levou a apresentar uma enorme capacidade de aprendizado e conseqüentemente de sobrevivência.

Podemos, na área computacional observar este mesmo tipo de dilema. Paradig-

mas tradicionais de computação, tais como os CBIs e os programas que eles executam, apresentam alto grau de especialização, apresentando uma ótima performance na solução de problemas algorítmicos. No entanto, seu desempenho é extremamente sofrível quando utilizados para resolver problemas imprecisos, mal-definidos ou nebulosos. Por outro lado, novos paradigmas, tais como a abordagem conexionista, parecem possuir uma boa capacidade para resolver esta classe de problemas imprecisos, apesar de eventualmente se sacrificar a precisão da resposta obtida.

A fim de verificar a capacidade de generalização das RNAs, primeiramente estabeleceu-se um determinado problema sobre o qual realizou-se uma série de experimentos.

### 7.3.1 O Problema do Aprendizado de Trajetórias

A capacidade de interceptação de objetos submetidos a movimentos balísticos está presente em grande número de animais. Considere por exemplo o que ocorre quando se joga um pedaço de madeira na direção de um cachorro, ou quando jogamos basquete, volei, ou mesmo futebol. Nestas situações a atividade de interceptação de objetos submetidos a trajetórias balísticas está sempre presente.

Uma característica da trajetória balística é que, no caso do plano, ela depende inteiramente das condições iniciais de lançamento, isto é, a velocidade e o ângulo iniciais, como pode-se ver pelas equações a seguir:

$$x = v_0 \cos(\Theta)t \quad (7.1)$$

□

e

$$y = v_0 \sin(\Theta)t - \frac{t^2}{2} \quad (7.2)$$

□

A combinação destas duas variáveis é capaz de gerar trajetórias que cobrem todo o espaço de estados  $\mathcal{R}^2$ .

Dentro de certos limites, o homem ou mesmo um cão é capaz de prever a trajetória balística de um objeto, mesmo que este tenha sido lançado com uma velocidade e ângulo inicial nunca vistos anteriormente. No caso de se utilizar uma RNA para executar esta tarefa, é impossível treinar a rede neural com todos os conjuntos possíveis de trajetórias. Esta rede deve apresentar uma capacidade de

generalização tal que, após aprender um determinado conjunto de trajetórias, ela possa prever todas as demais.

### 7.3.2 Implementação dos Experimentos

A rede neural utilizada para implementar o aprendizado da trajetória balística possui uma topologia direta com neurônios estáticos (feedforward). O algoritmo de aprendizado utilizado é o backpropagation. A rede possui três neurônios na camada de entrada, uma camada escondida e dois neurônios na camada de saída, como pode ser visto na Figura 7.3. Os neurônios de entrada correspondem ao ângulo inicial, à velocidade inicial e ao instante de tempo considerado. Na camada de saída obtém-se a posição  $(x, y)$  do objeto correspondente “as condições iniciais e ao instante considerado” [151].

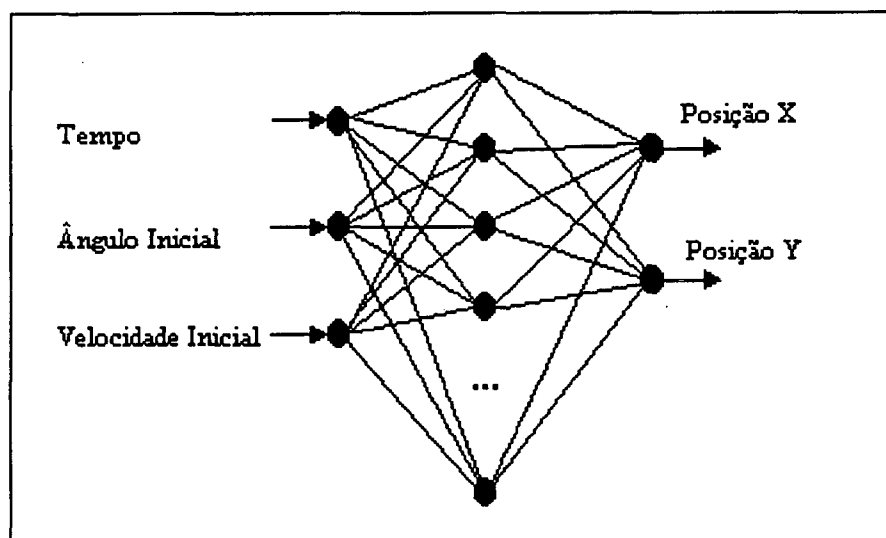


Figura 7.3: Rede Neural Direta para geração de trajetória balística

#### Experimento 1

Para que pudesse observar a capacidade de generalização desta rede neural, em um primeiro experimento ela foi treinada para aprender a trajetória balística de um objeto lançado com uma velocidade inicial ( $10m/s$ ) e com dois ângulos iniciais de lançamento ( $30^\circ$  e  $60^\circ$ ). Alguns pontos das coordenadas  $(x, y)$  da trajetória seguida por um objeto lançado sob estas duas condições iniciais, durante 2 segundos, foram

apresentados para a rede durante a fase de treinamento. Após o treinamento, a rede foi solicitada a fornecer os mesmos 2 segundos de trajetória para um objeto lançado com a mesma velocidade inicial, porém com ângulos de lançamento que variavam entre  $0^\circ$  e  $90^\circ$ . As trajetórias, real e calculada pela rede, para ângulos de lançamento de  $30^\circ$ ,  $45^\circ$  e  $60^\circ$  podem ser vistos na Figura 7.4. Também os valores máximos de erro entre a trajetória real e a trajetória calculada pela rede foram plotados e podem ser vistos na Figura 7.5.

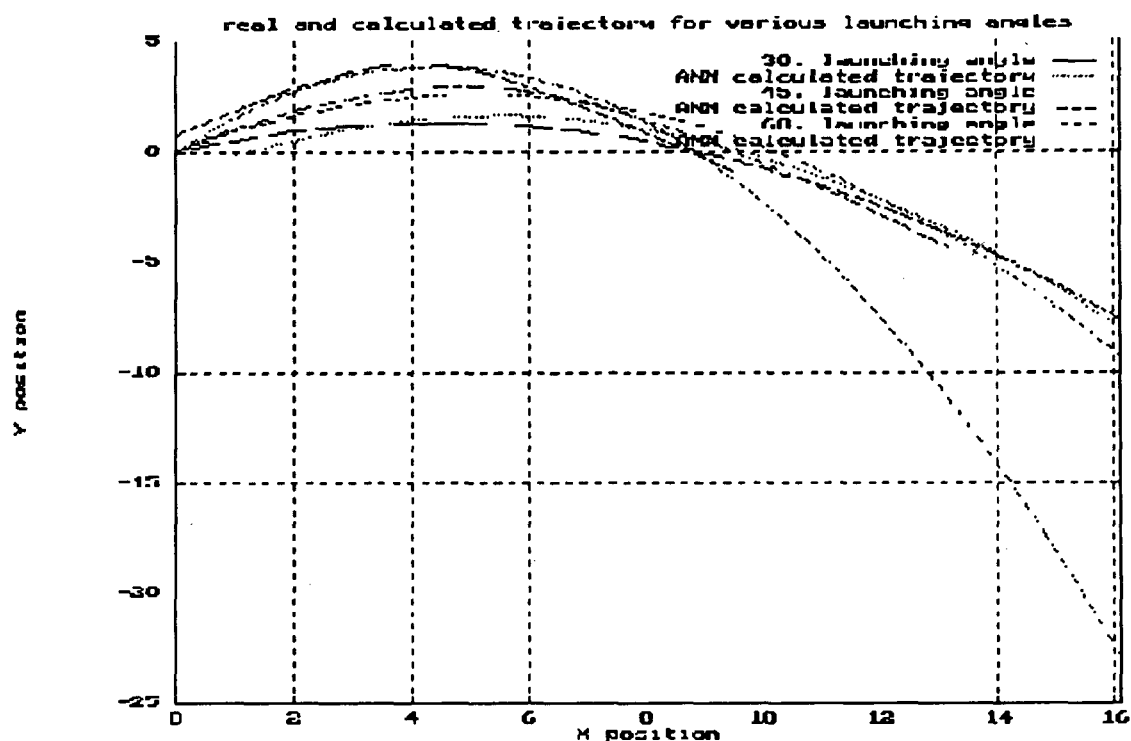


Figura 7.4: Trajetórias, real e calculada pela rede para ângulos de lançamento de  $30^\circ$ ,  $45^\circ$  e  $60^\circ$

Pode-se observar que para ângulos de lançamento dentro do intervalo composto pelos ângulos utilizados para treinar a rede, a interpolação feita pela rede foi bastante boa. No entanto, a sua capacidade de extrapolação para ângulos de lançamento fora do intervalo de ângulos utilizados durante a fase de treinamento é bastante limitada.

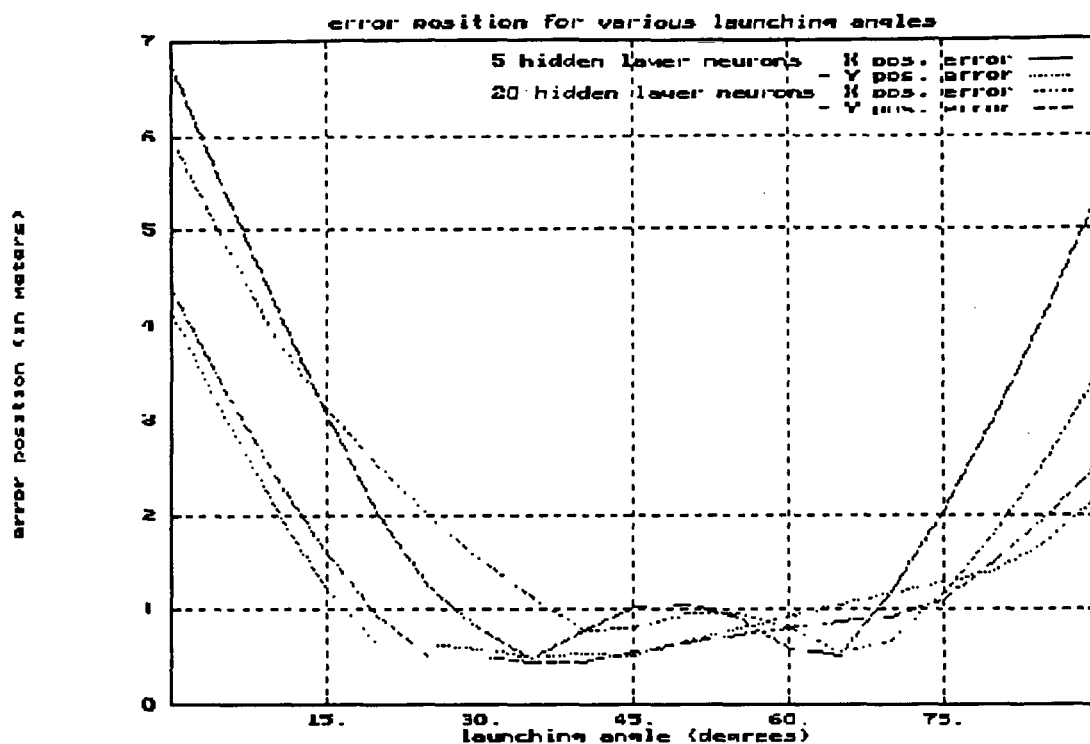


Figura 7.5: Erro máximo entre as trajetórias reais e calculadas pela rede para ângulos de lançamento entre  $0^\circ$  e  $90^\circ$ , com rede treinada para velocidade inicial de  $10\text{m/s}$  e ângulos de lançamento de  $30^\circ$  e  $60^\circ$

## Experimento 2

Neste segundo experimento desejou-se verificar a capacidade de interpolação da rede neural quando os conjuntos apresentados durante a fase de treinamento representavam trajetórias muito distantes uma das outras. Agora a rede foi treinada para trajetórias que correspondiam a velocidade inicial de  $10\text{m/s}$  e ângulos de lançamento de  $15^\circ$  e  $75^\circ$ . Como pode ser visto na Figura 7.6, a capacidade de generalização da rede não foi suficiente para interpolar corretamente trajetórias entre os dois ângulos de lançamento apresentados. Isto reforça a idéia de que se uma rede neural direta com neurônios estáticos não possui capacidade de extrapolação, sua capacidade de interpolação é limitada pelo grau de diversidade dos exemplos utilizados durante o treinamento.



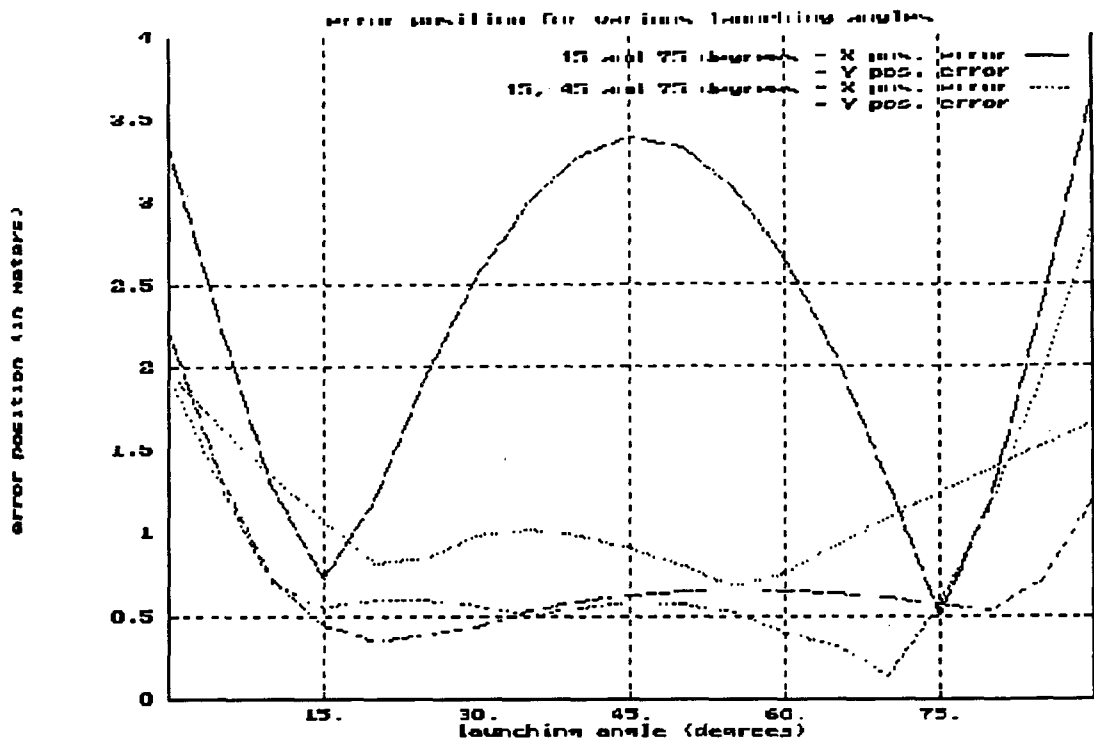


Figura 7.6: Erro máximo entre as trajetórias reais e calculadas pela rede para ângulos de lançamento entre  $0^\circ$  e  $90^\circ$ , com rede treinada para velocidade inicial de  $10m/s$  e ângulos de lançamento de  $15^\circ$  e  $75^\circ$

### Algumas conclusões

Os resultados obtidos dos experimentos permitem concluir que, em geral, redes neurais artificiais possuem uma boa capacidade de generalização, no sentido de que elas são capazes de produzir respostas corretas (apesar de não exatas) para questões não apresentadas durante a fase de treinamento. No entanto, para o caso da topologia e do exemplo apresentado, esta generalização ocorre apenas no sentido de interpolação entre as questões apresentadas na fase de treinamento, o que mostra a importância da escolha correta do conjunto de treinamento ao se utilizar redes neurais artificiais do tipo feedforward com neurônios estáticos e algoritmo de aprendizado do tipo backpropagation [151] [154]. Estudo detalhado sobre a importância da escolha do conjunto de treinamento para o algoritmo de aprendizado backpropagation foi feito por de Azevedo e pode ser encontrado em [59].

A capacidade de generalização, no sentido da extrapolação, pode ser uma característica extremamente desejável de determinado sistema e será examinada na próxima seção.

### 7.3.3 RNAs com Capacidade de Extrapolação

Com o que foi visto até aqui, pode-se supor que a extrapolação é uma capacidade bem mais complexa e difícil que a interpolação. Parece que um requisito para a extrapolação é de certa forma a aquisição de conhecimentos sobre os “princípios de funcionamento” do tema que envolve a questão que se quer resolver. De certo modo, redes neurais feedforward com neurônios estáticos não são capazes de representar estes “princípios”, principalmente quando estes “princípios” envolvem o funcionamento de sistemas dinâmicos.

Tome-se o seguinte exemplo: É bem sabido que redes neurais feedforward podem facilmente resolver o problema da paridade para um vetor de entrada com um certo número de bits [68] [161]. Digamos, por exemplo, um vetor de entrada de 8 bits, uma rede direta com 8 neurônios na camada de entrada, 8 neurônios na camada escondida, 1 neurônio na camada de saída e uma amostra representativa dos 256 possíveis vetores de entrada como conjunto de treinamento, é suficiente para resolver o problema [68]. Esta rede deveria essencialmente aprender a contar o número de bits em 1 no conjunto de treinamento.

Uma vez treinada, esta rede é capaz de interpolar entre os exemplos apresentados, chegando a acertar o valor da paridade para todos os 256 valores possíveis de serem questionados. O problema está justamente na limitação da rede, não pode-se dizer que ela aprendeu o “princípio de funcionamento” da questão da paridade, pois ela não é capaz de responder qual a paridade para um vetor de entrada de 9, 10 ou mais bits.

Vê-se que a própria topologia e modo de funcionamento da rede limita a sua capacidade de extrapolação. Uma maneira de fazer com que uma rede neural fosse capaz de entender o “princípio de funcionamento” da questão da paridade seria ensinar para ela, por exemplo, o próprio autômato que determina a paridade. Este autômato é mostrado na Figura 7.7. Somente é possível implementar este autômato em uma rede neural artificial se esta rede neural possuir uma topologia do tipo recorrente, como visto no capítulo anterior. Esta rede possui um único neurônio na camada de entrada, um valor diferente de zero de neurônios na camada escondida e um neurônio na camada de saída que realimenta suas saídas para os neurônios da camada escondida, como pode ser visto na Figura 7.8 [155] [158].

Uma vez que a rede tenha aprendido o autômato, o próprio conceito de paridade foi aprendido e, portanto, a rede será capaz de calcular a paridade de um

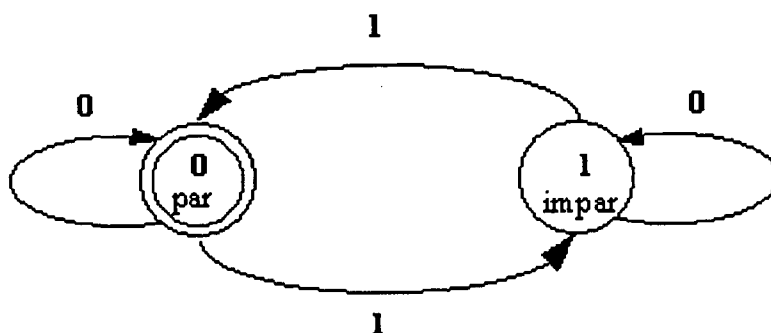


Figura 7.7: Autômato finito que implementa o cálculo da paridade de um “string”

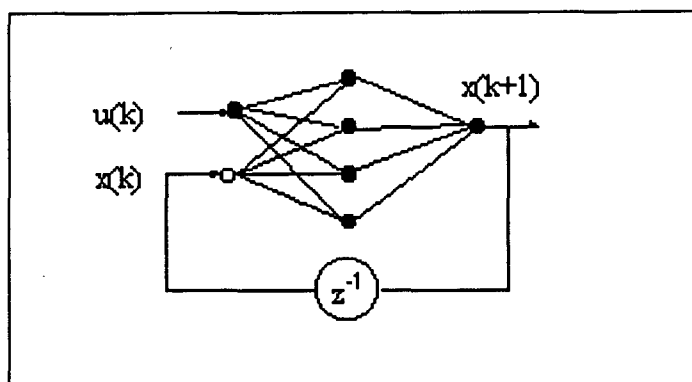


Figura 7.8: Rede Neural Artificial Recorrente que modela o autômato finito que implementa o cálculo da paridade de um “string”

vetor com um número qualquer de bits, bastando que os bits sejam apresentados seqüencialmente ao neurônio de entrada da rede.

Outra maneira de treinar a rede é apresentar para ela exemplos de vetores com paridade par e vetores com paridade ímpar. Ao ser apresentada a estes exemplos, seria interessante que a rede fosse capaz de induzir <sup>1</sup> um autômato capaz de implementar um reconhecedor para os exemplos apresentados. Uma vez induzido este autômato, então a rede teria adquirido o “princípio de funcionamento” da questão e como anteriormente poderia extrapolar para vetores de entrada com qualquer número de bits.

Este mesmo princípio se estende para sistemas dinâmicos contínuos no tempo, sendo de grande interesse na utilização de redes neurais em sistemas de controle de processos dinâmicos.

<sup>1</sup>indução s.f. - operação de estabelecer uma proposição geral com base no conhecimento de certo número de dados singulares [60].

## 7.4 Topologias de RNAs e suas Potencialidades Computacionais

Ao longo deste trabalho, várias vezes já foi mencionado o fato de um AA se caracterizar por ser capaz de operar determinada tarefa de maneira autônoma em um ambiente dinâmico. Também já foi visto que a tarefa de projetar, de maneira explícita, um agente para operar neste tipo de ambiente é tarefa praticamente impossível, pois é extremamente difícil antever na fase de projeto todas as possíveis situações com as quais o agente irá se defrontar no decorrer de sua operação. Na Natureza, os mecanismos evolucionários foram os responsáveis por prover novas características e capacidades aos organismos sempre que o repertório de capacidades já existentes não se mostrava suficiente para uma operação eficiente em um ambiente cada vez mais complexo. Geralmente, estas novas características eram geradas sobre as características já existentes, levando a um aumento crescente também da complexidade dos organismos com conseqüente aumento da inteligência e, portanto, do aumento da potencialidade do organismo em realizar comportamentos e tarefas também mais complexas.

Assim como o aumento de tamanho e complexidade do sistema nervoso levou este sistema a ser capaz de executar tarefas mais complexas, podemos observar que um aumento de tamanho e complexidade de estruturas de redes neurais artificiais também permite um aumento das potencialidades computacionais destas estruturas.

O que se deseja demonstrar nesta seção é que, dependendo da estrutura e do número de camadas de neurônios de redes neurais artificiais, existem limitações quanto às potencialidades computacionais destas redes. Queremos mostrar que, assim como redes neurais diretas com neurônios estáticos que não possuam camada de neurônios intermediários não são capazes de aprender padrões não-linearmente separáveis e que qualquer rede direta com neurônios estáticos é incapaz de modelar sistemas dinâmicos, também redes neurais recorrentes sem camada intermediária de neurônio são incapazes de aprender certa classe de autômatos finitos [158].

Barreto [19] propôs uma classificação dos problemas que podem ser tratados pelos neurocomputadores. A classificação proposta por Barreto é a seguinte:

1. Problemas estáticos linearmente separáveis.

Trata-se de problemas envolvendo a implementação de uma função (por ser um problema estático) e que podem ser resolvidos por um perceptron de uma camada de conexões.

2. Problemas estáticos linearmente não-separáveis.

Trata-se de problemas envolvendo a implementação de uma função (por ser um problema estático) e que podem ser resolvidos por uma rede direta, com neurônios estáticos, exigindo ao menos uma camada de neurônios internos.

### 3. Problemas dinâmicos com dinâmica finita.

Os problemas com dinâmica finita são aqueles que a duração da resposta do sistema após uma entrada dura um tempo finito. Um exemplo são os filtros FIR (“Finite Impulse Response”). Estes problemas podem ser resolvidos por uma rede direta com neurônios dinâmicos.

### 4. Problemas dinâmicos com dinâmica infinita.

Os problemas com dinâmica infinita são aqueles que a duração da resposta do sistema após uma entrada pode durar um tempo infinito. Um exemplo são os filtros IIR (“Impulse Infinite Response”). Estes problemas devem ser abordados por rede com retroação e com neurônios dinâmicos ou rede estática e conjunto de retardos. Neste caso o problema da estabilidade da rede, ou seja se a rede encontrará ou não uma solução e quanto tempo será necessário é problema em aberto.

## 7.4.1 Complexidade de Redes Neurais

A questão da complexidade se refere ao estudo da relação entre a quantidade de recursos computacionais e o tamanho de problema a ser solucionado. No caso das redes neurais esta questão pode ser dividida em dois pontos:

- Para um dado problema:

1. como definir a estrutura da rede neural capaz de solucioná-lo?
2. se o problema puder ser resolvido por uma rede neural, qual o menor tamanho da rede capaz de resolvê-lo?

Convém ressaltar, ainda, que existem pesquisadores que relacionam a questão da complexidade com o tempo necessário para treinar a rede [199] [111], o que não é o caso no presente trabalho.

Os primeiros modelos de redes neurais artificiais da década de 60 eram baseados no modelo de perceptron proposto por Rosenblatt e possuíam apenas duas camadas de neurônios. O que se notava, na época, é que, para alguns tipos de problemas de reconhecimento de padrões, estas redes não conseguiam convergir para valores de erro aceitáveis, sem que se soubesse exatamente o porquê. Até que em 1969

Minsky e Papert escreveram o famoso livro chamado *Perceptrons* [127]. O livro utiliza conceitos modernos de matemática, tais como topologia e teoria de grupo, para explicar as limitações e capacidades adaptativas e computacionais de redes simples de perceptrons.

A fim de fornecer subsídios para a elaboração de uma Teoria Geral de Complexidade de Redes Neurais, uma boa abordagem parece ser inicialmente continuar o trabalho de Minsky e Papert, procurando por novas classes de problemas e as respectivas estruturas de redes neurais capazes de resolvê-los.

Após o trabalho pioneiro de Minsky e Papert, diversos trabalhos na área da teoria da complexidade para redes neurais apareceram [2] [1] [26] [40] e [62].

Na sua visão de 1988 sobre a área, Minsky e Papert disseram que pouco progresso havia sido feito para formalizar as bases conceituais sobre RNAs. “...*there has been little clear-cut change in the conceptual basis of the field (connectionism)*”. “...*The issues that were then obscure (1969) remain obscure today because no one yet knows how to tell which of the present discoveries are fundamental and which are superficial.*” [128].

Nesta seção, nos restringimos principalmente à primeira questão, qual seja a de dada uma determinada classe de problemas, determinar qual a estrutura de rede neural capaz de resolvê-la.

#### 7.4.2 Teoremas de Complexidade de Redes Neurais

Uma vez estabelecidos os conceitos gerais sobre complexidade de redes neurais, voltamos o foco da discussão sobre que classes de problemas são melhor tratados por uma ou outra estrutura de rede neural.

Minsky e Papert tentaram abordar esta questão na edição de 1988 do livro *Perceptron's* [128] e de certa forma eles vislucbraram o caminho a ser seguido, enfatizando o grande poder representacional das redes neurais com ciclos, porém não foram adiante na questão.

“...*multilayer networks will be no more able to recognize connectedness than are perceptrons. (this is not to say that multilayer networks do not have advantages. For example, the product rule can yield logarithmic reductions in the orders and numbers of units required to compute certain high-order predicates. Furthermore, units that are arranged in loops can be of effectively unbounded order/ hence some such networks will be able to recognize connectedness by using internal serial processing.*” [128](página 252).

Voltamos agora a rever alguns teoremas já apresentados na seção 6.6 relacionando a estrutura da RNA com os tipos de sistemas que ela é capaz de modelar [19] [158].

**Teorema 7.4.1** *Uma Rede Neural Direta (feedforward), com neurônios estáticos é incapaz de modelar um sistema dinâmico.*

**Prova:** Uma Rede Neural Direta com neurônios estáticos é um Sistema Estático. A prova é consequência de que um sistema estático não pode representar um sistema dinâmico, pois para cada valor de estado temos um sistema estático diferente.

**Teorema 7.4.2** *Todo autômato finito pode ser representado por uma Rede Neural Recorrente.*

**Prova:** A prova será feita de modo construtivo utilizando neurônios binários, ou seja, com valores de saída binários. Um autômato finito pode ser descrito através do seguinte conjunto de equações:

$$x(t+1) = \Phi(x(t), u(t)) \quad (7.3)$$

$$y(t) = \eta(x(t), u(t)) \quad (7.4)$$

A função  $\eta$  pode ser implementada levando em consideração todas as saídas possíveis e utilizando tantos neurônio binários quantos forem necessários para codificar estas saídas. Devemos utilizar também um conjunto de neurônios cujas saídas codifiquem os estados do autômato. Os neurônios de entrada recebem o valor  $u(t)$  no instante considerado e o estado do autômato  $x(t)$  atrasados por um conjunto de elementos de retardo  $z^{-1}$  como pode ser visto na Figura 7.9.

**Teorema 7.4.3** *Uma Rede Neural Recorrente pode modelar um sistema dinâmico.*

**Prova:** Se considerarmos que todo Sistema Dinâmico pode ser representado com a precisão desejada por um autômato, e utilizando o teorema anterior que mostra que todo autômato finito pode ser representado por uma rede neural recorrente, então todo sistema dinâmico pode ser modelado através de uma Rede Neural Recorrente.

As características de dinâmica não-linear das Redes Neurais Recorrentes permitem que este tipo de rede neural seja utilizado em problemas e aplicações que necessitam de representação de estados, tais como: processamento de voz, controle industrial, processamento de sinais adaptativos, previsão de séries temporais e implementação de comportamentos reativos em AAs entre outras.

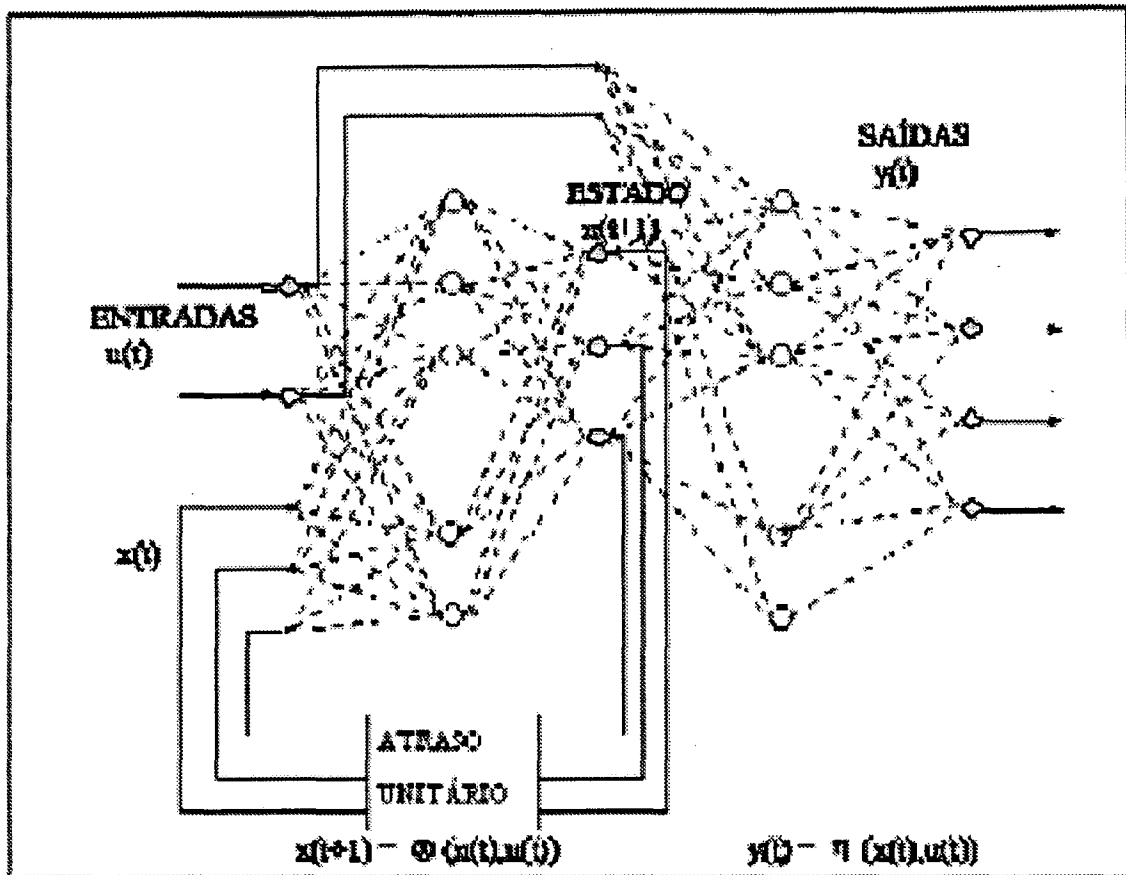


Figura 7.9: Modelo de Rede Neural Recorrente capaz de implementar um autômato finito

### 7.4.3 Classes de Problemas

Primeiramente Minsky e Papert definiram a ordem de um predicado como sendo o tamanho da maior conjunção na forma lógica da mínima soma-de-produtos para aquele predicado, ou seja, quantas linhas retas são necessárias para separar os pontos que compõem as categorias do predicado. Assim, enquanto o "OU" e o "E" são predicados de primeira ordem, o 'OU-EXCLUSIVO' é um predicado de ordem 2. Já a PARIDADE é um predicado de ordem infinita ou de ordem  $R$  onde  $R$  é o número de entradas [142].

Após catalogar a ordem de várias funções geométricas, Minsky e Papert voltaram sua atenção para o problema do aprendizado. Eles mostraram que a medida que a ordem dos predicados aumenta, o número de coeficientes pode crescer exponencialmente levando um sistema a necessitar quantidades enormes de memória e



estendendo o processo de convergência a um número infinito de iterações [142].

O leitor que desejar conhecer em detalhes o formalismo matemático da demonstração de Minsky e Papert é convidado a se referenciar diretamente à referência original [127] ou ainda, a um artigo de Pollack [142] onde ele faz uma análise específica sobre a edição expandida de 1988 do livro de Minsky e Papert.

Pode-se resumir a idéia de Minsky e Papert dizendo que redes neurais diretas com neurônios estáticos são capazes de aprender apenas funções lógicas de primeira ordem, ou seja, distinguir apenas padrões linearmente separáveis ou predicados de ordem 1 [68].

Já redes neurais diretas com neurônios estáticos com uma camada intermediária de neurônios e função de saída não-linear são capazes de modelar qualquer função estática no grau de precisão que se desejar, ou seja, estas redes podem resolver predicados de ordem 2 ou superior, no entanto, o número de coeficientes necessários para resolver predicados de ordem muito alta pode tornar inviável a utilização deste tipo de rede [158] [82] [87].

Ao chegar-se nos problemas dinâmicos, vê-se que apenas redes neurais recorrentes são capazes de resolver este tipo de problema. A questão, agora, diz respeito à topologia da rede recorrente e que tipo de problemas dinâmicos ela é capaz de resolver. Este trabalho irá se ater apenas ao caso da implementação de sistemas dinâmicos discretos, especificamente o aprendizado de autômatos finitos por redes neurais recorrentes.

#### 7.4.4 A Paridade como um problema de ordem infinita

Tome-se inicialmente uma rede recorrente com apenas uma camada de neurônios de entrada e uma camada de neurônios de saída. A saída de cada neurônio é ligada através de um conjunto de linhas de retardo à entrada de cada um dos outros neurônios da camada de saída, inclusive para si próprio, como pode ser visto na Figura 7.10. O problema proposto para esta rede é a implementação de um autômato finito. Tomemos, por exemplo, o autômato finito que implementa a paridade como pode ser visto na Figura 7.7. Neste caso, a paridade é a extensão do predicado “OU-EXCLUSIVO” com ordem infinita.

O que se deseja é fazer neste caso é, dado um “string” de bits apresentado à entrada da rede, obter na sua saída o valor da paridade do “string” apresentado.

A rede recorrente contém apenas um neurônio de entrada e um neurônio de saída e pode ser vista na Figura 7.10. Uma maneira de treinar esta rede é utilizar o

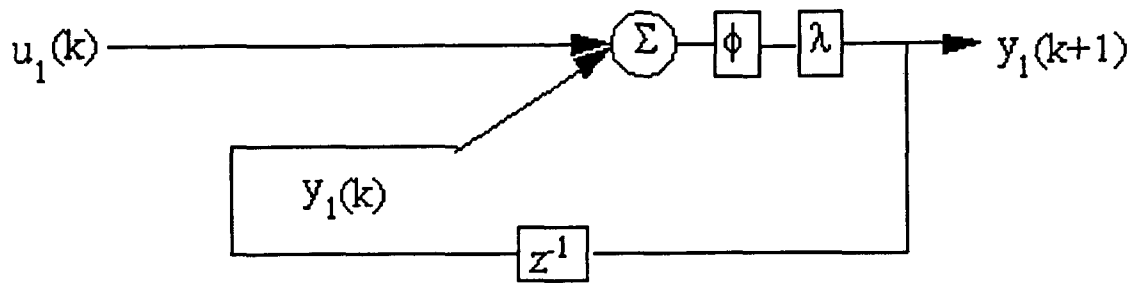


Figura 7.10: Rede Neural Artificial Recorrente sem camada intermediária para tentar implementar o autômato finito que implementa o cálculo da paridade de um “string”

algoritmo do tipo backpropagation já apresentado anteriormente. Para isto, durante a fase de treinamento a rede é transformada em uma rede direta equivalente e o conjunto de vetores de entrada para o treinamento inclui o valor do estado anterior do autômato finito. A rede direta equivalente pode ser vista na Figura 7.11 e o conjunto de treinamento na Tabela 7.1.

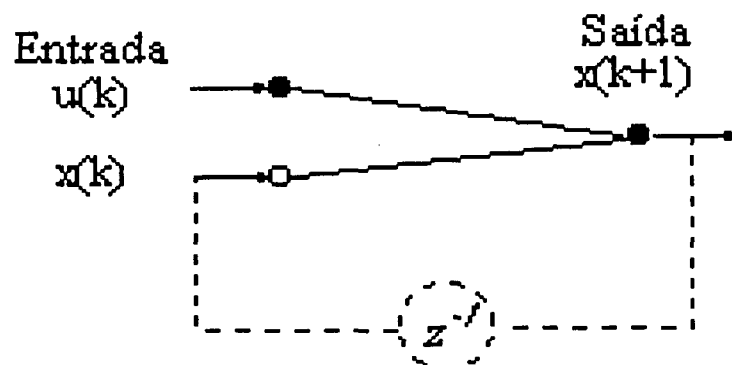


Figura 7.11: Rede Neural Direta equivalente à rede recorrente sem camada intermediária incapaz de aprender o autômato finito da paridade

O estado 0 significa paridade par e o estado 1 corresponde à paridade ímpar. Assim, se o autômato estava no estado 0 e o próximo bit do “string” for 1, a paridade passa a ser ímpar (estado 1), caso o próximo bit fosse 0, a paridade permaneceria sendo par (estado 0). Se o autômato estivesse no estado 1 e o próximo bit do “string” for 1, a paridade passa a ser par (estado 0), caso o próximo bit fosse 0, a paridade permaneceria sendo ímpar (estado 1).

Entrada $u(k)$	Estado $x(k)$	Estado $x(k+1)$
0	0	0
1	0	1
0	1	1
1	1	0

Tabela 7.1: Conjunto de treinamento para o autômato que implementa o cálculo da paridade

Ao analisar-se a Tabela 7.1, contendo o conjunto de treinamento, percebe-se claramente que o conjunto de treinamento representa a função lógica “OU-EXCLUSIVO”. Como a rede neural direta equivalente a rede recorrente para o treinamento não possui camada intermediária de neurônios, é impossível implementar esta função com esta topologia de rede neural. Logo, autômatos que implementam extensões de predicados de ordem maior que 1, não podem ser implementados utilizando redes neurais recorrentes sem camada intermediária de neurônios.

Por extensão, podemos perceber que redes neurais recorrentes com uma camada intermediária de neurônios são capazes de implementar qualquer autômato finito.

## 7.5 Implementando Comportamentos com RNAs

### 7.5.1 Taxias e Reflexos

Conforme já foi visto anteriormente, formas simples de comportamentos estereotipados, tais como as taxias e os reflexos, são respostas instantâneas e bem definidas a estímulos específicos recebidos pelos órgãos sensoriais. Uma vez terminado o estímulo, cessa também a resposta a ele. Tendo em mente uma perspectiva evolucionária, pode-se imaginar uma forma de vida extremamente primitiva, com um sistema nervoso sensório-motor extremamente rudimentar, com apenas uns poucos neurônios. O único requisito para a sobrevivência deste tipo de organismo seria a habilidade de sentir (provavelmente através de sensores químicos) a presença de alimento e de se mover na sua direção em um ambiente livre de predadores e de obstáculos. É fácil verificar que esta forma de comportamento simples, puramente reflexivo, pode ser descrita através de uma função combinacional ou contínua apenas dos valores presentes nos sensores, e que, portanto, pode ser implementada utilizando redes neurais diretas [153].

Outras evidências de que comportamentos reflexivos e taxias são implementados

através das conexões diretas entre os neurônios podem ser fartamente encontradas na Natureza. No próprio ser humano, sabe-se que os circuitos responsáveis pelos reflexos incondicionados se localizam ao nível da medula espinhal [81].

É importante ressaltar também que na Natureza estes tipos de comportamentos não foram definidos ou aprendidos explicitamente, mas emergiram como resultado do processo evolucionário. Também é interessante notar que alguns destes comportamentos podem ser implementados conectando-se diretamente os neurônios sensores aos neurônios atuadores, o que corresponde a redes neurais diretas sem camada escondida. Outros comportamentos, no entanto, requerem a existência de interneurônios, ou seja, redes neurais com uma camada intermediária.

É fácil demonstrar que um comportamento de taxia do tipo movimento em direção a um estímulo, como um alimento, já descrito no exemplo 3.4.1, pode ser implementada por uma rede com apenas dois sensores e dois atuadores conectados de forma cruzada, como mostrado na Figura 7.12. A função combinacional que descreve o movimento é dada pela Tabela 7.2.

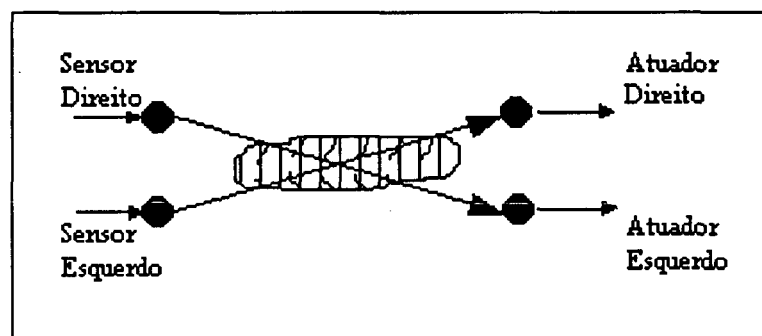


Figura 7.12: Rede Neural capaz de implementar o comportamento de “movimento em direção ao alimento”

Note-se que se a criatura não for capaz de detectar a presença do alimento o seu comportamento é o de permanecer imóvel até que alguma forma de alimento ingresse no seu campo de percepção. Uma criatura que caso não detectasse presença de alimento, pudesse se deslocar em círculos, teria maiores chances de encontrar o alimento e, portanto, aumentaria suas chances de sobrevivência em um ambiente em que o alimento se tornasse escasso. Este comportamento, bastante parecido com o comportamento anterior, pode ser descrito pela Tabela 7.3.

Repare agora que para implementar este novo comportamento, não bastam as conexões cruzadas entre os neurônios. É necessário, agora, a presença de um neurônio

SENSOR ESQUERDO	SENSOR DIREITO	ATUADOR ESQUERDO	ATUADOR DIREITO
-1 - não detecta alimento		-1 - atuador deslig.	
+1 - detecta alimento		+1 - atuador ligado	
-1	-1	-1	-1
-1	+1	+1	-1
+1	-1	-1	+1
+1	+1	+1	+1

Tabela 7.2: Função combinacional que descreve o comportamento de taxia

SENSOR ESQUERDO	SENSOR DIREITO	ATUADOR ESQUERDO	ATUADOR DIREITO
-1 - não detecta alimento		-1 - atuador deslig.	
+1 - detecta alimento		+1 - atuador ligado	
-1	-1	-1	+1
-1	+1	+1	-1
+1	-1	-1	+1
+1	+1	+1	+1

Tabela 7.3: Função combinacional que descreve o novo comportamento

de “bias” e uma camada intermediária de neurônios de modo a manter um dos atuadores ligado, mesmo quando a criatura não for capaz de detectar a presença de alimento, como pode ser visto na Figura 7.13.

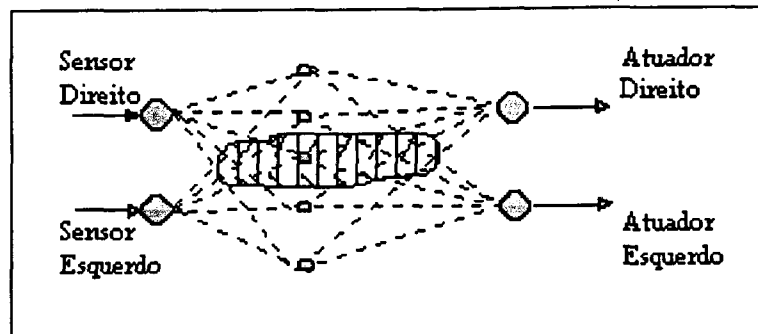


Figura 7.13: Criatura primitiva com sistema nervoso composto apenas de conexões diretas, porém com uma camada de neurônios intermediários

## 7.5.2 Comportamentos Reativos e Instintivos

A medida em que a população das criaturas primitivas descritas na seção anterior começa a ficar maior, a disponibilidade de comida se torna menor e o ambiente se torna mais complexo e hostil, e apenas aquelas criaturas com um grau de aptidão maior terão maiores chances de sobrevivência. Em um certo instante, a capacidade de sobrevivência não pode mais ser alcançada apenas com um repertório limitado de comportamentos puramente reflexivos. É necessário, então, um sistema nervoso mais complexo, com uma nova topologia que inclua neurônios conectados em ciclo entre si. Esta nova arquitetura de sistema nervoso permite, assim, a emergência de novos tipos de comportamentos, mais complexos e versáteis, aumentando as chances de sobrevivência destas novas criaturas.

O próximo passo deste trabalho envolve a implementação deste modelo de sistema nervoso em uma arquitetura de RNA que apresente as mesmas características.

Como já foi visto anteriormente, é possível descrever uma série de classes de comportamento estereotipado através de Autômatos de Estados Finitos, entre elas os comportamentos reativo e instintivo. Assim, propõe-se aqui uma arquitetura de rede neural capaz de implementar estes autômatos tanto a nível dos seus estados como a nível das saídas geradas [156].

Esta arquitetura de rede neural proposta é na verdade composta por duas redes interconectadas e cujas topologias já foram descritas anteriormente.

A primeira rede é utilizada para mapear os estados do AEF que se deseja implementar. Cada estado corresponde a um dado comportamento da seqüência de comportamentos que compõem, por exemplo, o comportamento reativo. Nesta rede, os neurônios de entrada atuam como órgãos sensoriais, recebendo os sinais externos provenientes do ambiente. Os neurônios da camada de saída são neurônios dinâmicos com função de saída linear e função de ativação do tipo atraso unitário. A saída destes neurônios está conectada a todos os neurônios da camada escondida. Este por sua vez são neurônios estáticos com função de saída do tipo tangente hiperbólica. Um diagrama simplificado desta primeira rede pode ser visto na Figura 7.14.

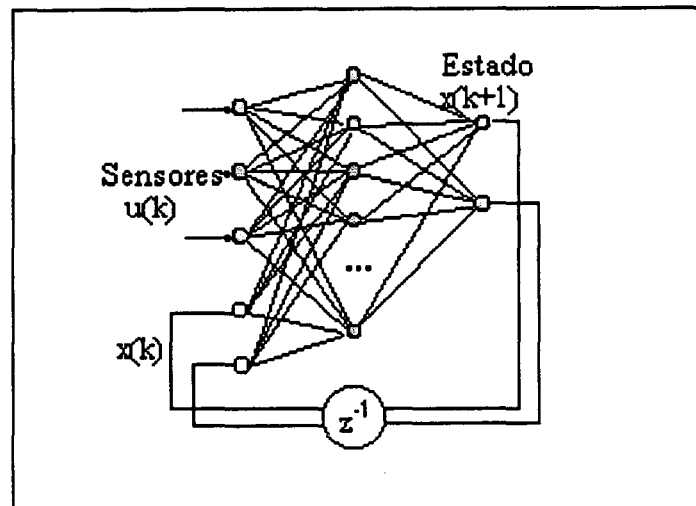


Figura 7.14: Rede Neural Recorrente com camada escondida capaz de aprender os estados de um Autômato de Estados Finitos

É necessário, ainda, mapear cada estado nos respectivos valores para os atuadores, que vão efetivamente caracterizar o comportamento da criatura. Para isto, uma segunda rede é utilizada. Agora, basta uma rede neural direta com neurônios estáticos, cujos neurônios de entrada são compostos pelos neurônios de entrada e de saída da primeira rede, como pode ser visto na Figura 7.15.

Um experimento foi proposto objetivando implementar um comportamento reativo bastante simples em um AA, de modo a testar as idéias apresentadas. Este comportamento reativo foi descrito utilizando um autômato de estados finitos e este

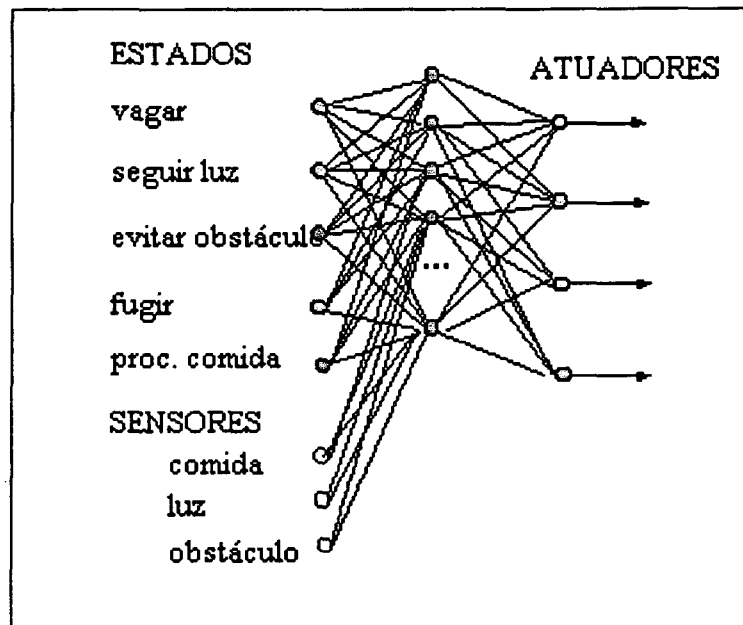


Figura 7.15: Rede Neural Direta capaz de aprender as saídas de um Autômatos de Estados Finitos

autômato foi implementado no modelo de rede neural descrito anteriormente. O comportamento implementado que permitiria a um AA percorrer um dado ambiente em busca de “tokens” ou sinais que comporiam um trilha é chamado de “tracker task” e foi descrito por Jefferson et al. [88] e extraído de Angeline [10].

*“Neste problema, uma formiga simulada é colocada em uma grade toroidal bidimensional que contém uma trilha de alimento, como pode ser visto na Figura 7.17. A formiga deve percorrer a grade, coletando ao longo do caminho todo alimento que for encontrado. O objetivo da tarefa é descobrir um controlador para a formiga que seja capaz de coletar a maior quantidade de alimento em um dado intervalo de tempo.*

*De acordo com Jefferson et al., cada formiga é controlada por uma rede neural com dois neurônios de entrada e quatro neurônios de saída, como pode ser visto na Figura 7.16. O primeiro neurônio de entrada,*



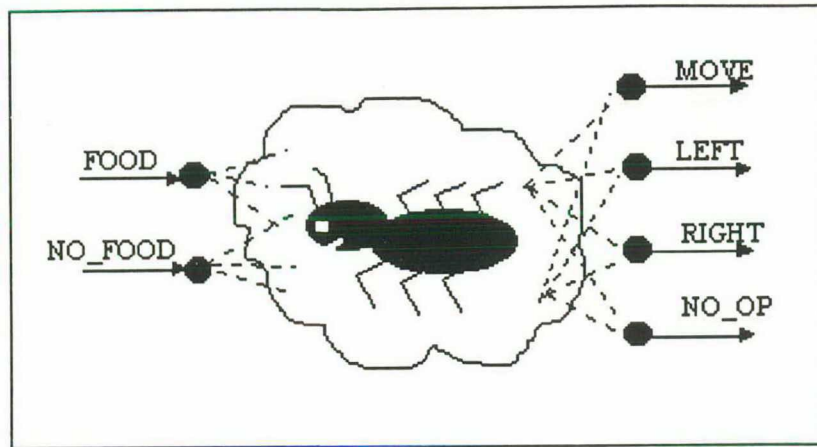


Figura 7.16: Representação simplificada da formiga simulada com seus sensores e atuadores

*chamado de FOOD na figura é acionado caso exista alimento no quadrado imediatamente a frente da formiga. O segundo neurônio de entrada, chamado de NO FOOD é o complemento do primeiro neurônio e é acionado caso não exista alimento no quadrado imediatamente a frente da formiga. Isto restringe o conjunto de valores de entrada válidos apenas a  $(1,-1)$  e  $(-1,1)$ .*

*Cada um dos quatro neurônios de saída, chamados de MOVE, LEFT, RIGHT e NO-OP na figura, corresponde a uma das seguintes ações possíveis para a formiga - andar um quadro para a frente, virar à esquerda  $90^\circ$ , virar à direita  $90^\circ$ , e não fazer nada respectivamente. A cada passo, a ação correspondente ao neurônio que possuir o maior valor de saída para aquele passo é executada.*

*Uma possível solução para este problema é o AEF sugerido explicitamente por Jefferson et al. e mostrado na Figura 7.18. Esta máquina simples descreve o seguinte comportamento: vá para o quadrado da frente se houver comida naquele quadrado, caso contrário, vire para a direita até quatro vezes procurando por comida. Se neste processo alguma comida for encontrada, vá para o quadrado da frente, caso contrário, vá para o quadrado da frente e repita todo o procedimento.”*

Este tipo de comportamento só pode ser implementado através de uma rede neural do tipo recorrente, pois cada movimento depende não apenas da existência de comida no quadrado em frente, mas também é necessário “memorizar” o número de vezes que a saída “virar a direita  $90^\circ$ ” foi produzida anteriormente.

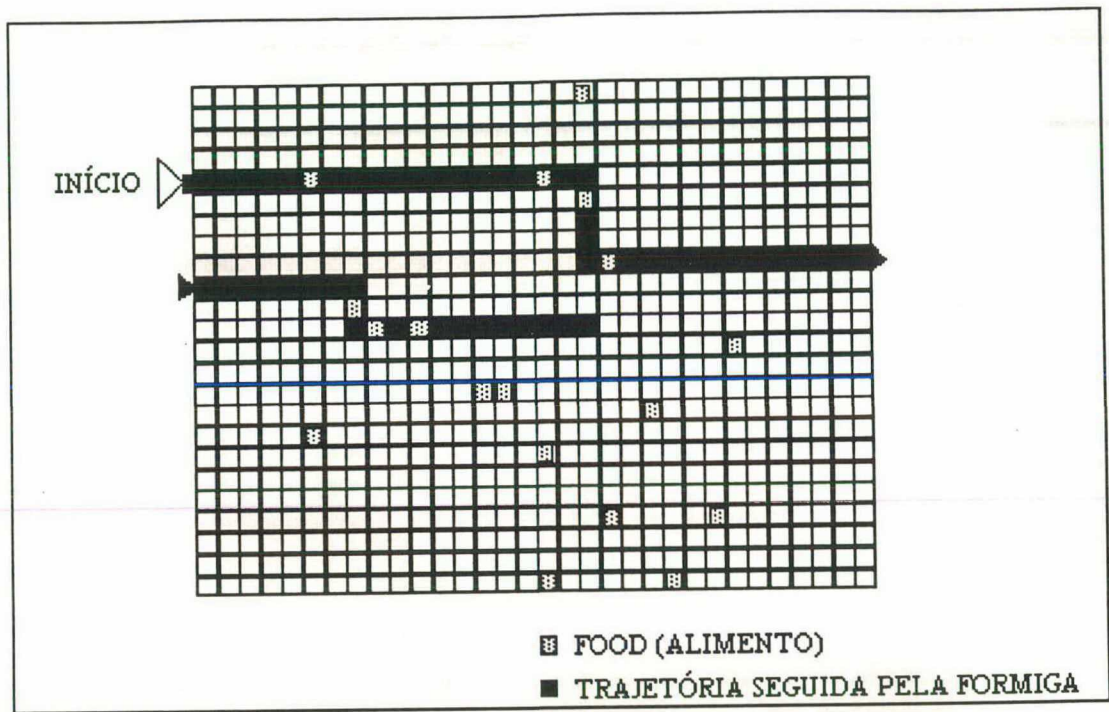


Figura 7.17: Plano toroidal onde a formiga é posta para operar e trajetória seguida pela formiga

O diagrama esquemático da rede neural composta capaz de implementar o comportamento descrito na formiga simulada é mostrado na Figura 7.19. Neste caso, a parte recorrente da rede foi treinada para reconhecer os estados do AEF que descreve o comportamento, enquanto que a parte direta da rede gera a saída para os atuadores correspondente às saídas do AEF, baseada no estado atual do AEF e do valor dos sensores presente nos neurônios de entrada.

O problema a ser enfrentado agora diz respeito a como ensinar os estados do AEF à parte recorrente da rede neural, uma vez que, como já foi visto no capítulo anterior, o treinamento de redes neurais recorrentes é tarefa bem mais complexa quando comparado com o treinamento de redes neurais diretas.

Na Natureza grande parte dos comportamentos estereotipados apresentados pelos animais parecem ser determinados geneticamente através de mecanismos evolucionários, ou seja, os sistemas nervosos parecem já vir de certa modo “pré-conectados” de modo a executarem os comportamentos determinados. Este é um mecanismo biológico importante e sua implementação será examinada na seção 7.7.

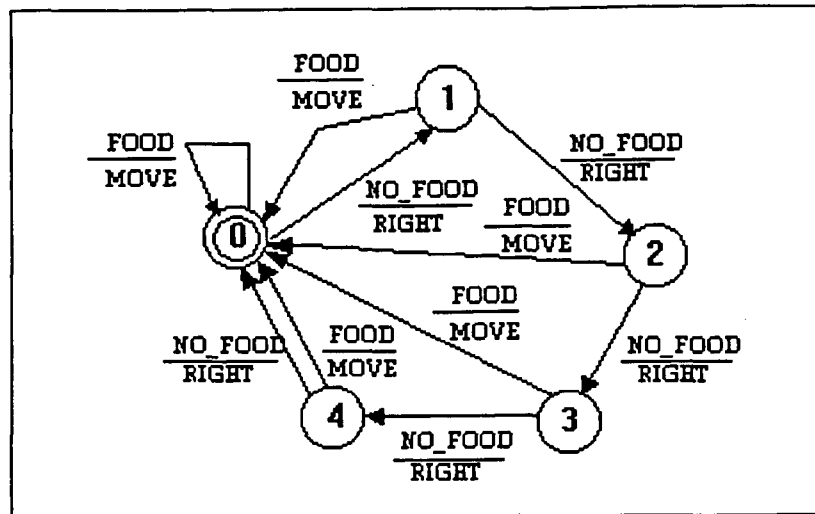


Figura 7.18: Autômato de Estados Finitos que descreve um possível comportamento para a formiga

Sem esquecer este fato, é interessante neste trabalho, examinar-se também mecanismos não evolucionários para o aprendizado de redes neurais, em função do tempo necessário para se desenvolver este aprendizado. Se o aprendizado evolucionário se desenrola ao longo de várias gerações, o aprendizado não evolucionário, se desenrola durante o período de existência do organismo, o que representa uma escala de tempo muito menor.

Na próxima seção, então, são apresentadas algumas idéias de como fazer este treinamento no caso do aprendizado dos estados de autômatos finitos.

## 7.6 Aprendizado em Rede Neurais Recorrentes

Em muitos trabalhos envolvendo a utilização de redes neurais recorrentes para a modelagem de sistemas dinâmicos, os modelos de redes recorrentes utilizados e a própria característica do problema que se está tratando, tornam o treinamento deste tipo de rede neural tarefa bastante complexa. No caso específico de sistemas dinâmicos contínuos, os valores que se conhecem são os conjuntos dos valores de entrada e a resposta do sistema constitui o conjunto dos valores de saída. Normalmente nestes sistemas não estão disponíveis o conjunto de estados, tão pouco a função de transição de estados. Isto faz com que, nestes casos, a própria rede recorrente tente induzir a função de transição de estados baseada nos conjuntos de valores de entrada e saída do sistema apresentados durante a fase de treinamento.

O fato do modelo de rede neural recorrente utilizado neste trabalho para a im-

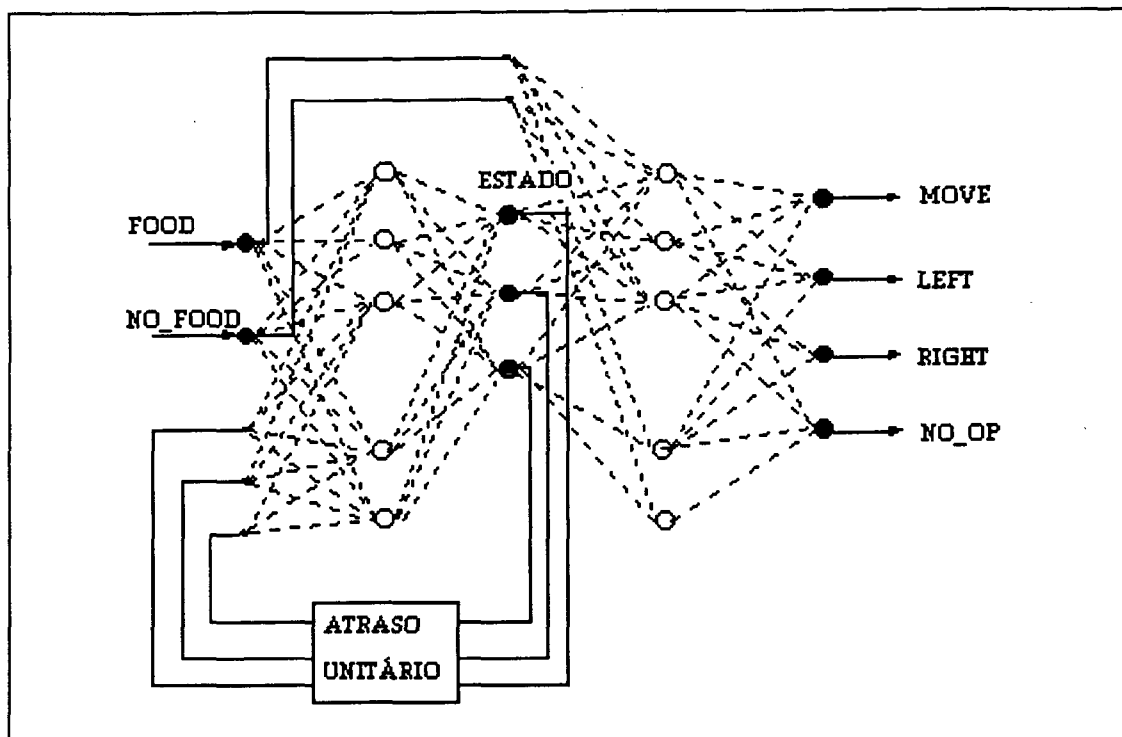


Figura 7.19: Rede Neural Artificial Composta que implementa o comportamento descrito para a formiga

plementação de autômatos finitos possuir, na sua camada de saída, neurônios que codificam de maneira explícita os estados que compõem o autômato facilita bastante a tarefa de aprendizado. Apesar disto, mais adiante neste capítulo, será apresentado um modelo de rede recorrente capaz de induzir autômatos finitos baseado em exemplos de entrada/saída desejados ou ainda, exemplos válidos e não-válidos de uma gramática.

### 7.6.1 Um Algoritmo do Tipo Backpropagation

Como neste trabalho os comportamentos a serem implementados são descritos através de AEFs, a codificação dos estados que compõem o autômato de maneira explícita na parte recorrente da rede neural composta utilizada para implementar o comportamento facilita o desenvolvimento de algoritmos de aprendizado.

Em um primeiro momento, desenvolveu-se um algoritmo do tipo backpropagation para treinar a parte recorrente da rede neural composta.

Isto é feito basicamente transformando a rede neural recorrente em uma rede neural direta durante a fase de treinamento. Durante esta fase as linhas de atraso que conectam a saída dos neurônios de saída à entrada dos neurônios da camada inter-

mediária são retiradas e neurônios correspondentes ao valor de saída dos neurônios da camada de saída no instante de tempo  $k$  são inseridos como neurônios de entrada da rede, transformando a rede recorrente em uma rede direta equivalente, cujos neurônios de entrada agora são compostos pelos neurônios correspondentes aos valores dos sensores no instante  $k$  e pelos neurônios que representam o estado do AEF também no instante  $k$ .

O conjunto de treinamento a ser apresentado para a rede é composto agora pela seqüência temporal de todos os possíveis eventos que acionam o AEF, de tal modo que se percorra todos os estados possíveis do AEF. Deste modo, apresenta-se na entrada da rede um valor de entrada  $u(k)$  e o respectivo valor do estado do AEF  $x(k)$  no mesmo instante  $k$ . Ao deixar-se a rede propagar para a frente estes sinais de entrada, obtém-se na saída o próximo estado calculado pela rede  $x_c(k+1)$ . Como o próximo estado desejado  $x_d(k+1)$  é conhecido (é o estado para o qual o AEF evolui quando ocorre o evento  $u(k)$ ), basta subtrair  $x_d(k+1)$  por  $x_c(k+1)$  obtendo um valor de erro e aplicando o algoritmo tradicional de backpropagation.

Um exemplo de comportamento descrito através de um AEF e que foi implementado em uma rede recorrente utilizando o algoritmo especificado acima foi extraído de [90] e é apresentado a seguir:

### Lâmpadas e Botões

Suponha-se um agente bastante simples cuja entrada sensorial é capaz de perceber a condição de duas lâmpadas. Se a primeira lâmpada estiver acesa, o primeiro sensor é acionado, caso a segunda lâmpada esteja acesa, o segundo sensor é acionado. O agente possui também um atuador capaz de agir sobre dois botões. Se o valor do atuador for -1, o primeiro botão é acionado, se o valor do atuador for 1, o segundo botão é acionado. Uma ou nenhuma das lâmpadas podem estar acesas em um determinado instante, mas não as duas simultaneamente. O comportamento desejado é que o agente pressione o botão correspondente a lâmpada que estiver ligada, se eventualmente uma delas estiver ligada. Caso nenhuma das lâmpadas esteja acesa, o agente deve pressionar o botão correspondente à lâmpada que estava acesa mais recentemente.

O AEF mostrado na Figura 7.20 descreve formalmente este comportamento. Note-se que neste caso, o próprio valor do estado corresponde de maneira unívoca aos valores de saída do AEF.

A rede neural direta equivalente à rede recorrente utilizada para o aprendizado

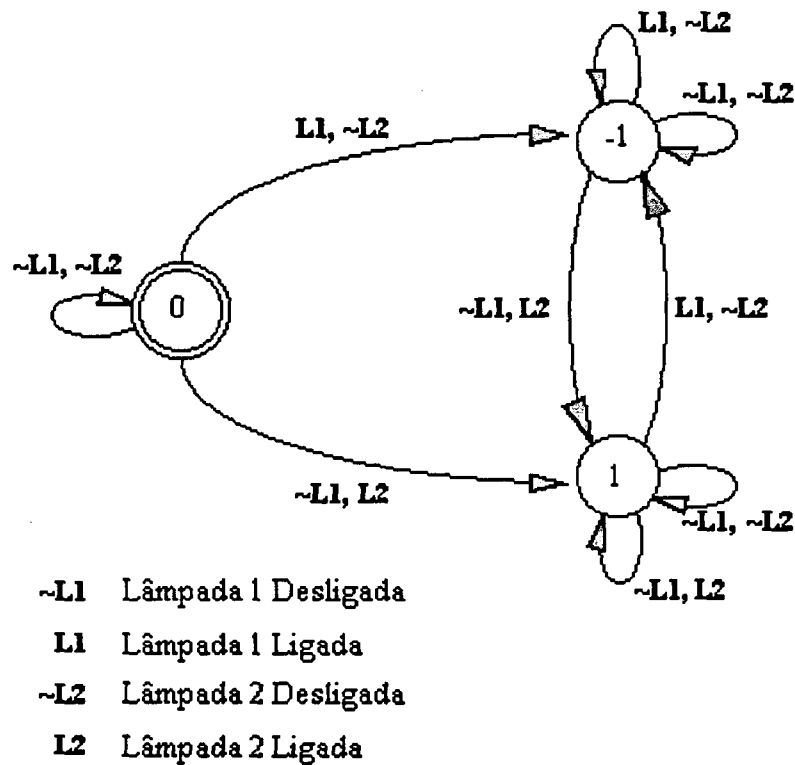


Figura 7.20: Autômato finito que descreve o comportamento do agente que resolve o problema das lâmpadas e dos botões

dos estados do AEF pode ser vista na Figura 7.21 e o conjunto de treinamento utilizado para treinar a rede é apresentado na Tabela 7.4. Aqui o valor -1 corresponde a lâmpada desligada e o valor 1 corresponde a lâmpada ligada. O estado inicial é indeterminado e corresponde ao valor intermediário 0.

Após o aprendizado, uma vez estabelecidos todos os pesos das conexões, basta retornar à configuração original da rede, sendo que os pesos calculados para as conexões entre os neurônios de entrada correspondentes ao estado no instante  $k$  ( $x(k)$ ), passam a ser o valor dos pesos das conexões dos ciclos entre os neurônios de saída da rede e os neurônios da camada intermediária da rede recorrente.

## 7.6.2 Um Algoritmo Biologicamente Inspirado

Apesar do algoritmo backpropagation de aprendizado ser extremamente eficiente em termos de velocidade de convergência e precisão obtida, ele apresenta algumas limitações. Em primeiro lugar, deve-se ter em mente que o algoritmo backpropagation

Entradas $u(k)$		Estado $x(k)$	Estado $x(k+1)$
lâmpada 1	lâmpada 2		
-1	-1	0	0
+1	-1	0	-1
+1	-1	-1	-1
-1	-1	-1	-1
-1	+1	-1	+1
-1	+1	+1	+1
-1	-1	+1	+1
+1	-1	+1	-1
+1	-1	-1	-1
-1	-1	0	0
-1	+1	0	+1
-1	+1	+1	+1
-1	-1	+1	+1
+1	-1	+1	-1
+1	-1	-1	-1
-1	-1	-1	-1
-1	+1	-1	+1
-1 - lâmpada desligada		-1 - botão 1 acionado	
+1 - lâmpada ligada		+1 - botão 2 acionado	
		0 - nenhum botão acionado	

Tabela 7.4: Conjunto de treinamento para o AEF que implementa o comportamento das lâmpadas e dos botões

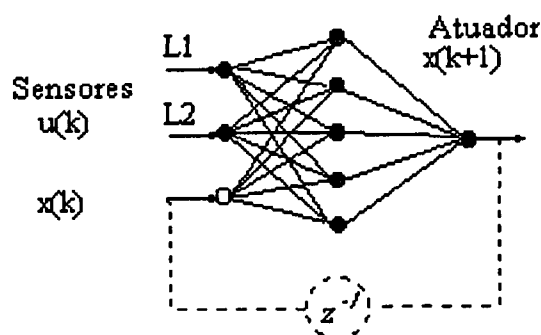


Figura 7.21: Rede Neural Direta equivalente à rede recorrente capaz de aprender o autômato finito da lâmpadas e botões através do algoritmo de backpropagation

não parece ser biologicamente plausível, uma vez que existem evidências científicas da impossibilidade de existir um circuito de retropropagação de erros que não são locais à sinapse entre os neurônios do sistema nervoso natural. Além disso, para poder utilizar o algoritmo backpropagation, deve-se conhecer de antemão os estados que compõem o autômato, o que eventualmente pode não acontecer se desejarmos que a própria rede induza os estados do autômato. Assim, nesta seção apresenta-se um algoritmo de aprendizado que é biologicamente inspirado, baseado nos paradigmas de tentativa-e-erro e aprendizado por reforço para determinação dos pesos de conexões e que pode ser utilizado tanto em redes neurais diretas, como recorrentes.

A idéia geral deste algoritmo se baseia no princípio de que os neurônios poderiam alterar o valor de suas conexões sinápticas por um mecanismo de punição e recompensa ou grau de satisfação das respostas dadas pela rede. Se as respostas se aproximam do valor desejado as conexões são “reforçadas”, enquanto que se as respostas se afastam dos valores desejados as conexões são “enfraquecidas”.

Num primeiro momento os pesos iniciais das conexões possuem valores aleatórios. Durante o processo de aprendizado, ao alterar-se o valor de peso de uma conexão qualquer observa-se se a rede convergiu no sentido da minimização de um valor de erro global. Em caso afirmativo, novas alterações que eventualmente ocorram no peso daquela conexão seguirão a mesma direção anterior, ou seja se um aumento do valor do peso de determinada conexão levou a uma diminuição do erro global, então existe uma grande probabilidade de que um novo aumento do peso daquela conexão continue a diminuir o erro. Caso contrário, se um aumento do valor do peso de uma conexão levou a um aumento do valor do erro global da rede, então, se invertermos a direção, isto é, diminuirmos o valor daquele peso, existe uma grande probabilidade de que o erro venha a diminuir.

O algoritmo é descrito de modo mais detalhado a seguir.



1. Inicialize randomicamente, com valores pequenos, o valor dos pesos das conexões da rede neural e cada conexão com uma “direção de perturbação” com valor +1 ou -1;
2. Propague todos os vetores de entrada do conjunto de treinamento através da rede, comparando o resultado obtido com o valor desejado e calcule um valor de erro global, por exemplo, o erro médio quadrático;
3. Para cada conexão da rede faça:
  - (a) Calcule um novo peso, somando o valor atual do peso com um pequeno delta multiplicado com a “direção de perturbação” daquela conexão;
  - (b) Propague novamente todos os vetores de entrada do conjunto de treinamento através da rede e obtenha um novo valor de erro;
    - i. Se o valor global do erro diminuiu, então selecione uma nova conexão e volte para o passo 3a;
    - ii. Se o valor global do erro aumentou, inverta a “direção de perturbação”, restaure o valor anterior do peso da conexão e volte para o passo 3a; (se em uma segunda iteração o valor do erro global ainda não houver diminuído, reduza o valor da “direção de perturbação” e volte para o passo 3a.)
4. Quando todas as conexões tiverem sido atualizadas, volte para o passo 2 até que o valor do erro global tenha sido reduzido para valores aceitáveis.

Este algoritmo foi implementado em dois sistemas, o primeiro chamado NNT (Neural Network development Tool) para redes diretas e um segundo sistema chamado RNNT (Recurrent Neural Network development Tool), sendo testado para o aprendizado tanto de comportamentos reflexivos em redes neurais diretas como de comportamentos reativos em redes neurais recorrentes. No caso das redes neurais recorrentes, assim como no caso do aprendizado do tipo backpropagation, o conjunto de treinamento deve ser composto pela seqüência temporal de todos os possíveis estados que compõem o AEF.

Para o caso da formiga simulada já apresentado anteriormente, a Tabela 7.5 contém o conjunto de treinamento para ensinar à rede neural recorrente os estados do AEF que descreve o comportamento desejado:

Uma vez que a rede recorrente foi treinada para implementar os estados do AEF, treina-se a rede direta para mapear os valores dos sensores e do estado atual do AEF para os valores corretos dos atuadores correspondentes às saídas do autômato. A

Sensor $u(k)$		Estado $x(k+1)$			
FOOD	NO_FOOD				
+1	-1	-1	-1	-1	(0)
+1	-1	-1	-1	-1	(0)
-1	+1	-1	-1	+1	(1)
+1	-1	-1	-1	-1	(0)
-1	+1	-1	-1	+1	(1)
-1	+1	-1	+1	-1	(2)
+1	-1	-1	-1	-1	(0)
-1	+1	-1	-1	+1	(1)
-1	+1	-1	+1	-1	(2)
-1	+1	-1	+1	+1	(3)
+1	-1	-1	-1	-1	(0)
-1	+1	-1	-1	+1	(1)
-1	+1	-1	+1	-1	(2)
-1	+1	-1	+1	+1	(3)
-1	+1	+1	-1	-1	(4)
+1	-1	-1	-1	-1	(0)
-1	+1	-1	-1	+1	(1)
-1	+1	-1	+1	-1	(2)
-1	+1	-1	+1	+1	(3)
-1	+1	+1	-1	-1	(4)
-1	+1	-1	-1	-1	(0)

Tabela 7.5: Conjunto de treinamento para os estados do AEF que implementa o comportamento da formiga simulada

Sensor $u(k)$		Estado $x(k+1)$	Saída do Atuador
+1	-1	(0)	MOVE
-1	+1	(0)	RIGHT
+1	-1	(1)	MOVE
-1	+1	(1)	RIGHT
+1	-1	(2)	MOVE
-1	+1	(2)	RIGHT
+1	-1	(3)	MOVE
-1	+1	(3)	RIGHT
+1	-1	(4)	MOVE
-1	+1	(4)	RIGHT

Tabela 7.6: Conjunto de treinamento para os valores de saída do AEF que descreve o comportamento da formiga simulada

Tabela 7.6 contém o conjunto de treinamento que descreve este mapeamento para o caso da formiga simulada.

O que se observou através dos experimentos realizados é que muitas vezes, dependendo dos valores iniciais dos pesos das conexões, este algoritmo de aprendizado biologicamente inspirado leva a situações de mínimos locais, em que o valor do erro global é maior do que o aceitável e no entanto ele não consegue mais ser reduzido, não importando o quanto se diminua o valor dos pesos das conexões ou se inverta o valor da “direção de perturbação” de cada conexão.

A solução encontrada, então, é escolher aleatoriamente um novo conjunto de pesos iniciais de recomeçar o algoritmo. Após um certo número de tentativas e erros, o algoritmo termina por selecionar um conjunto de pesos iniciais adequados e o erro global é reduzido a valores aceitáveis.

## 7.7 Programação Evolucionária para a geração de Agentes Autônomos

### 7.7.1 Motivação

Assim como o comportamento animal e o sistema nervoso forneceram inspiração para o desenvolvimento de ferramentas que auxiliam na implementação de AAs mais eficientes, também os mecanismos naturais de evolução servem como inspiração para

a geração da RNA responsável pela operação de um agente. Nesta seção descrevemos um sistema capaz de simular ambientes diferentes onde os AAs são postos para operar. Através de técnicas de programação evolucionária, o sistema gera diferentes redes neurais artificiais que conectam os sensores dos agentes com seus atuadores. Inicialmente, assim como acredita-se tenha ocorrido durante a evolução dos seres vivos sobre a Terra, sistemas nervosos extremamente simples são criados. Estes sistemas nervosos, por suas próprias características topológicas e pequeno número de neurônios possibilita apenas um repertório muito pequeno de comportamentos primitivos. A medida que o ambiente fica mais complexo, comportamentos mais complexos são necessários. Estes, porém, só podem ser implementados pela geração e uso de redes neurais recorrentes. Deste modo, nesta seção introduzimos e descrevemos algumas funções de custo associadas com a complexidade da rede neural, o que conduz a solução de problemas do tipo multi-objetivo, muito difíceis de resolver através de técnicas convencionais [24].

Foram feitas experiências em diferentes ambientes, com vários níveis de complexidade. Os resultados obtidos permitem confirmar a hipótese de que comportamentos mais simples, tais como os comportamentos puramente reflexivos, podem ser implementados por arquiteturas simples de redes neurais estáticas, como por exemplo, as do tipo feedforward. Em outros ambientes, comportamentos mais complexos são necessários para permitir a sobrevivência do agente, como comportamentos do tipo reativo, que persistem e se desenvolvem mesmo depois de terminado o estímulo sensorial disparador da ação. Nestes casos, o sistema gera arquiteturas de redes neurais mais complexas, do tipo recorrente, necessárias para introduzir dinâmica e memória no comportamento do agente.

### 7.7.2 O Sistema EVAG

EVAG, que significa EVolutionary Agents Generator, é um algoritmo evolucionário que procura simular aspectos relevantes do processo evolutivo que se acredita, aconteceu sobre a Terra, cujo objetivo é gerar a rede neural artificial de AA capaz operar ou “sobreviver” em um determinado ambiente.

O sistema é capaz de simular diferentes ambientes onde AAs são postos para operar durante um determinado tempo. Ao final deste tempo, uma função de aptidão é avaliada e, através da utilização de técnicas de Programação Evolucionária, diferentes redes neurais artificiais são geradas conectando os sensores dos agentes aos seus atuadores.

Funções de custo associadas com a complexidade da rede neural são introduzidas. Algumas heurísticas simples são utilizadas para associar uma complexidade com a

arquitetura de rede neural gerada. Por exemplo, é muito mais fácil para o sistema gerar conexões diretas entre os neurônios que compõem a rede do que conexões recorrentes. Do mesmo modo, é mais fácil gerar uma nova conexão do que criar novos neurônios para a rede, e assim por diante. Usando estas heurísticas, o sistema começa sua operação tentando achar arquiteturas de RNAs mais simples. Se estas arquiteturas não podem garantir a operação dos agentes no ambiente, então mutações devem ocorrer a fim de criar redes cada vez mais complexas. Deste modo, o comportamento emergente permite ao agente operar em determinado ambiente e transmitir a arquitetura da rede para novas gerações.

Considera-se que os sensores (neurônios de entrada) são capazes de sentir todos os tipos de objetos presentes no ambiente, tais como comida, parede e veneno. O número de sensores é, então, determinado pelos diferentes tipos de objetos presente no ambiente e o seu valor de ativação é -1 se o objeto não é sentido ou +1 quando o objeto é sentido. Os atuadores (neurônios de saída) permitem ao agente se mover uma célula à frente, virar à direita ou virar à esquerda. A ação a ser executada é determinada pelo maior atuador com maior valor de ativação. Existe também um neurônio de “bias” que é implementado como um neurônio de entrada adicional cuja valor de ativação constante de +1. Todos os neurônios utilizam função de ativação do tipo tangente hiperbólica.

O ambiente é uma superfície celular toroidal bi-dimensional e que pode ser configurada pelo usuário do sistema de modo a conter os seguintes elementos para interação com os agentes:

- comida - elemento que fornece “energia” para que o agente possa se movimentar pelo ambiente;
- veneno - elemento que retira a “energia” do agente; e
- parede - elemento que representa um obstáculo, impedindo que o agente se movimente para a frente.

O agente pode ser configurado para possuir ou não um sensor capaz de detectar a presença de determinado elemento na célula imediatamente em frente à atualmente ocupada por ele.

No instante inicial, os agentes iniciam sua operação no ambiente não possuindo nenhuma conexão entre os seus sensores (neurônios de entrada) e os seus atuadores (neurônios de saída), como pode ser visto na Figura 7.22

Cada geração, ou ciclo de vida, é composto por um número configurável de “instantes de tempo”. Em cada “instante de tempo”, o ambiente, no caso a célula imediatamente em frente ao agente, é sentida pelos sensores do agente e os sinais são

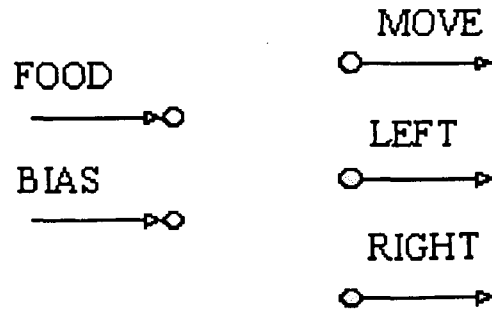


Figura 7.22: Configuração inicial do sistema nervoso do agente no instante em que ele é posto para operar no ambiente

propagados pela rede neural do agente. A ação executada pelo agente corresponde ao neurônio de saída que possuir o maior valor de ativação entre os atuadores.

Se a ação executada for “MOVER UMA CÉLULA À FRENTE”, esta ação poderá conduzir o agente a uma célula contendo alimento, este então é consumido e o nível de energia do agente é elevado. Caso a célula contenha veneno, então o nível de energia do agente será drasticamente reduzido. Se a célula em questão possuir uma parede, o agente permanece na célula anterior e uma unidade de energia é consumida. Caso nenhum destes elementos estiver presente, o agente ocupa a célula e uma unidade de energia é consumida.

Se a ação executada for “VIRAR À DIREITA”, uma unidade de energia é consumida e o agente faz uma volta de  $90^\circ$ . No caso da ação selecionada ser “VIRAR À ESQUERDA”, uma unidade de energia é consumida e o agente faz uma volta de  $-90^\circ$ .

O valor de aptidão é definido como o nível de energia remanescente no agente após um ciclo de vida. O número de agentes é configurável e permanece fixo durante a operação do sistema. Já o número de alimentos e venenos também são configuráveis, sendo que a quantidade de veneno permanece fixa enquanto que a quantidade de alimento vai diminuindo a medida em que for sendo consumido pelos agentes.

Após uma geração, o valor de aptidão dos agentes é verificado e os 50% superiores são escolhidos como “pais” da próxima geração, os 50% restantes “morrem” e servem como substrato para os novos agentes da próxima geração.

A geração de uma descendência envolve três passos:

- cópia dos agentes sobreviventes;
- determinação das mutações que vão ocorrer;

- mutação das cópias.

Uma mutação pode causar a alteração do valor do peso de uma conexão da rede neural de um agente, e é chamada de *Mutação Paramétrica*; ou a mutação pode ocasionar, ainda, o surgimento de novos neurônios ou novas conexões na rede neural de um agente e é chamada de *Mutação Estrutural*.

Como já foi dito anteriormente, a idéia é de que existe uma função de custo associada à complexidade da rede neural, ou seja, é mais “fácil” para a Natureza alterar o valor de uma conexão entre dois neurônios do que criar uma nova conexão que, por sua vez, é mais fácil de ocorrer do que a criação de um novo neurônio. Deste modo, no sistema EVAG esta função de custo é simulada através de distribuições de frequência, ou seja o tipo de mutação é determinado através de funções de probabilidade. A probabilidade de ocorrência de uma *Mutação Paramétrica* é de 80%, enquanto a probabilidade de *Mutações Estruturais* é de 20%. Uma vez que o sistema decide executar uma *Mutação Estrutural*, a probabilidade de gerar uma nova conexão é de 75%, enquanto a chance de criar um novo neurônio é de 25%. Do mesmo modo, a chance de, ao criar uma conexão, criar uma conexão do tipo recorrente, ou seja, para um neurônio da mesma camada ou de uma camada anterior, é de 30%.

É importante ressaltar que estes valores foram obtidos de forma heurística, após uma série de experimentos. Apesar disto, eles parecem ser capazes de criar redes neurais com níveis crescentes de complexidade ao mesmo tempo em que cobrem porções razoáveis do espaço de busca de determinada topologia de rede neural.

### 7.7.3 Experimentos

Nesta seção são descritos três experimentos realizados utilizando o sistema EVAG. O objetivo é mostrar a capacidade do algoritmo para gerar arquiteturas apropriadas de redes neurais artificiais que permitam a um agente operar em um ambiente com um dado grau de complexidade.

#### Ambiente 1

No primeiro experimento, o ambiente selecionado foi o mais simples possível. Uma certa quantidade de alimento foi distribuída de maneira aleatória sobre a superfície toroidal, como mostra a Figura 7.23. Os agentes são, então, colocados para operar neste ambiente. Após duas ou três gerações o sistema encontra uma arquitetura de rede neural capaz de manter a operação do agente neste ambiente. A solução proposta pelo sistema consiste de uma conexão excitatória direta entre o neurônio de

“bias” e o atuador responsável pela ação de “MOVER PARA A FRENTE”, como mostrado na Figura 7.24. Como o ambiente é uma superfície toroidal o comportamento emergente gerado pela rede neural leva o agente a percorrer em linha reta todas as células da superfície coletando os alimentos encontrados pelo caminho.

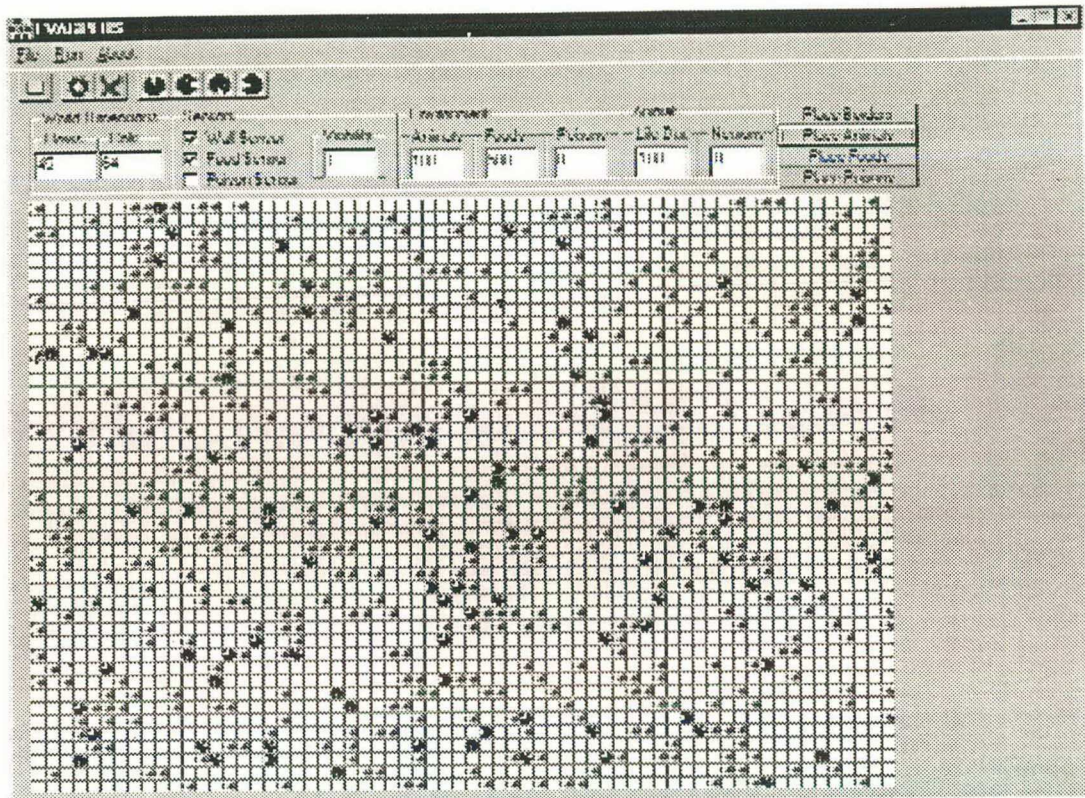


Figura 7.23: Ambiente simples contendo agentes e alimentos aleatoriamente distribuídos

## Ambiente 2

Na segunda experiência, o ambiente selecionado é um pouco mais complexo. Agora além dos alimentos aleatoriamente espalhados pelo ambiente, também elementos de veneno estão presentes neste ambiente, conforme mostra a Figura 7.25.

Quando os agentes começam a operar, o primeiro comportamento que surge após algumas gerações é semelhante ao apresentado no primeiro experimento, com uma conexão excitatória entre o neurônio de “bias” e o atuador de “MOVER PARA A FRENTE”, o que leva os agentes a consumirem, indistintamente, tanto os alimentos como os venenos. Após três ou quatro gerações as mutações que ocorrem nos agentes



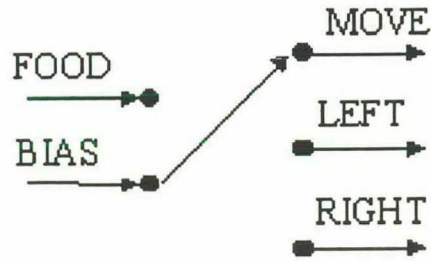


Figura 7.24: Rede Neural Feedforward desenvolvida pelo sistema que permite ao agente percorrer o ambiente coletando alimentos

conduzem a soluções mais otimizadas, como por exemplo:

- o surgimento de agentes com conexões inibitórias entre o sensor de veneno e o atuador de “MOVER PARA A FRENTE” como mostrado na Figura 7.26.a;
- o surgimento de agentes com conexões excitatórias entre o sensor de veneno e o atuador de “VIRAR À ESQUERDA” ou “VIRAR À DIREITA”, como na Figura 7.26.b;
- ou ainda o surgimento de soluções mais complexas, como a encontrada pelo sistema durante uma das execuções, após 30 gerações surgiu um agente cuja rede neural possuía conexões excitatórias entre o sensor de alimento e o atuador de “MOVER PARA A FRENTE” e entre o sensor de veneno e o atuador de “VIRAR À ESQUERDA”, além de uma conexão inibitória entre o sensor de alimento e o atuador de “VIRAR À ESQUERDA”, como mostra a Figura 7.26.c.

Todas estas soluções provocam a emergência do comportamento reflexivo de andar pelo ambiente evitando, ao mesmo tempo, o contato com elementos de veneno.

### Ambiente 3

No terceiro experimento procurou-se comprovar a idéia de que a medida em que o ambiente fica mais complexo, a “sobrevivência” das “criaturas” que nele operam não pode mais ser garantida apenas por um repertório de comportamentos puramente reflexivos. Assim, através de processos evolucionários, deve-se desenvolver um sistema nervoso que permita o surgimento de comportamentos mais complexos, baseados em “memória”, como os comportamentos reativos.

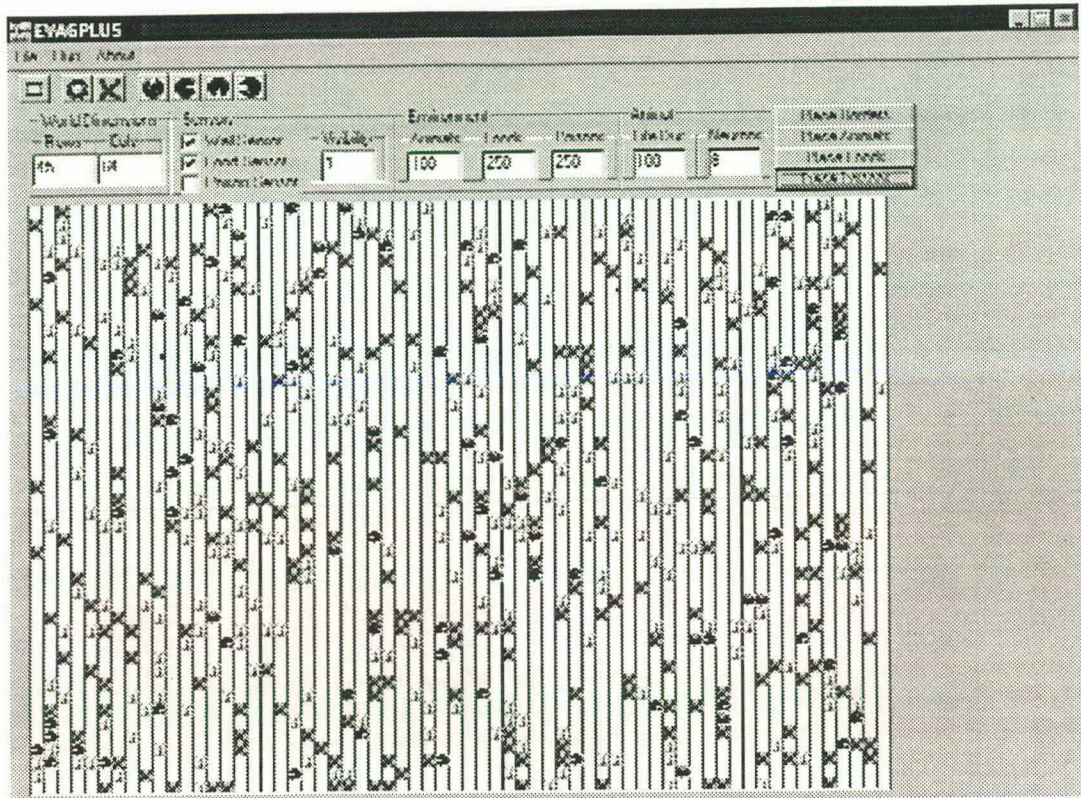


Figura 7.25: Ambiente contendo agentes, alimentos e venenos aleatoriamente distribuídos

Neste experimento, o veneno é removido e as bordas do ambiente são completamente cercadas por paredes, conforme a Figura 7.27. Quando o agente se move para a frente ele, inevitavelmente, acaba batendo na parede ficando imobilizado. Para que o agente possa encontrar alimento e, conseqüentemente, aumente seu nível de energia, o agente deve abandonar a parede e voltar para a parte central do ambiente onde ele tem maiores chances de encontrar alimento. No entanto, abandonar a parede requer um comportamento reativo, no sentido de que uma vez que o agente detecte a parede, ele deve virar a esquerda ou a direita, deixando de detectar a presença da parede, e então ele deve “memorizar” que, no passo anterior, a parede havia sido detectada, virando mais uma vez no mesmo sentido, resultando em uma volta de  $180^\circ$ . Se o agente, agora, voltar a andar para a frente ele se deslocará em direção ao centro do ambiente, com maiores chances de encontrar alimento.

Note-se que este comportamento dinâmico não pode mais ser implementado com uma rede neural feedforward. Assim, uma rede neural recorrente com ciclos entre os neurônios e capaz de modelar este comportamento dinâmico deve ser desenvolvida.

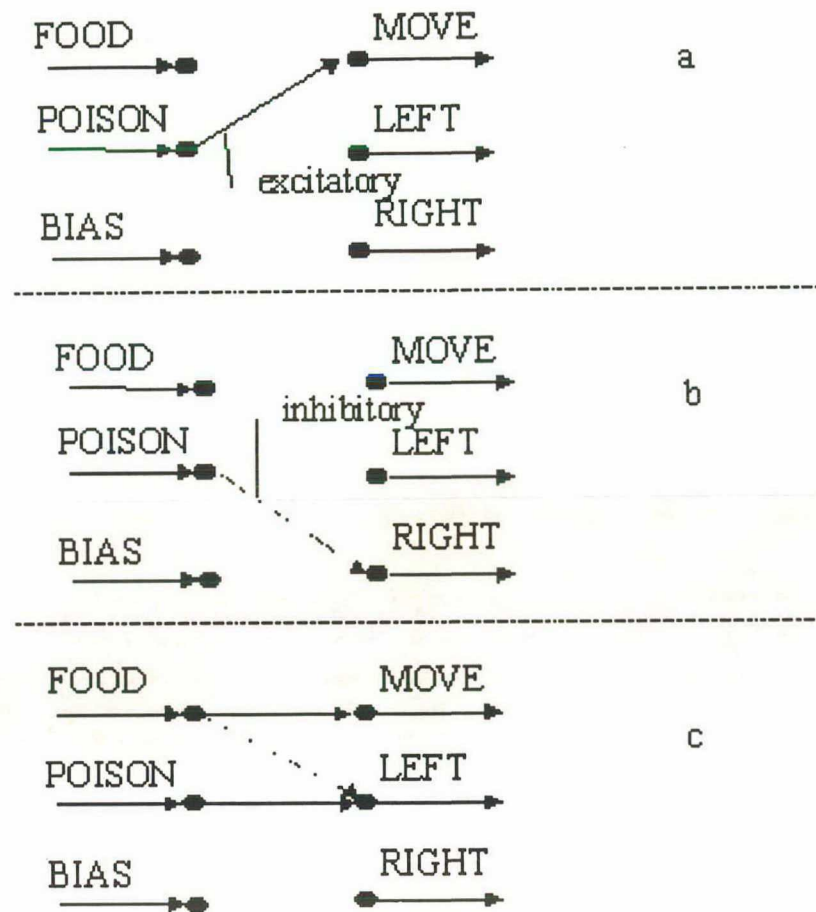


Figura 7.26: Algumas soluções encontradas pelo sistema para percorrer o ambiente coletando alimentos ao mesmo tempo em que evita consumir veneno

Após uma série de experimentos, constatou-se que o sistema EVAG acha soluções para este ambiente após um número médio de 370 gerações. As redes neurais desenvolvidas pelo sistema normalmente implementam o comportamento descrito acima, no entanto, algumas vezes outras soluções são encontradas pelo sistema como, por exemplo, o comportamento de virar 90° ao encontrar uma parede, mover-se uma célula à frente e depois virar novamente de 90°, ou ainda virar 180° ao detectar uma parede e ainda virar 90° após consumir um alimento, percorrendo áreas muito maiores do ambiente. Este comportamento é implementado pela rede neural recorrente mostrada na Figura 7.28 e foi encontrado pelo sistema depois de apenas 30 gerações.

Estes resultados permitem mostrar que as classes de comportamentos estão relacionadas com a arquitetura dos sistemas nervosos e sugere como estes sistemas nervosos se desenvolveram através de mecanismos evolucionários. Deste modo, sugere-se

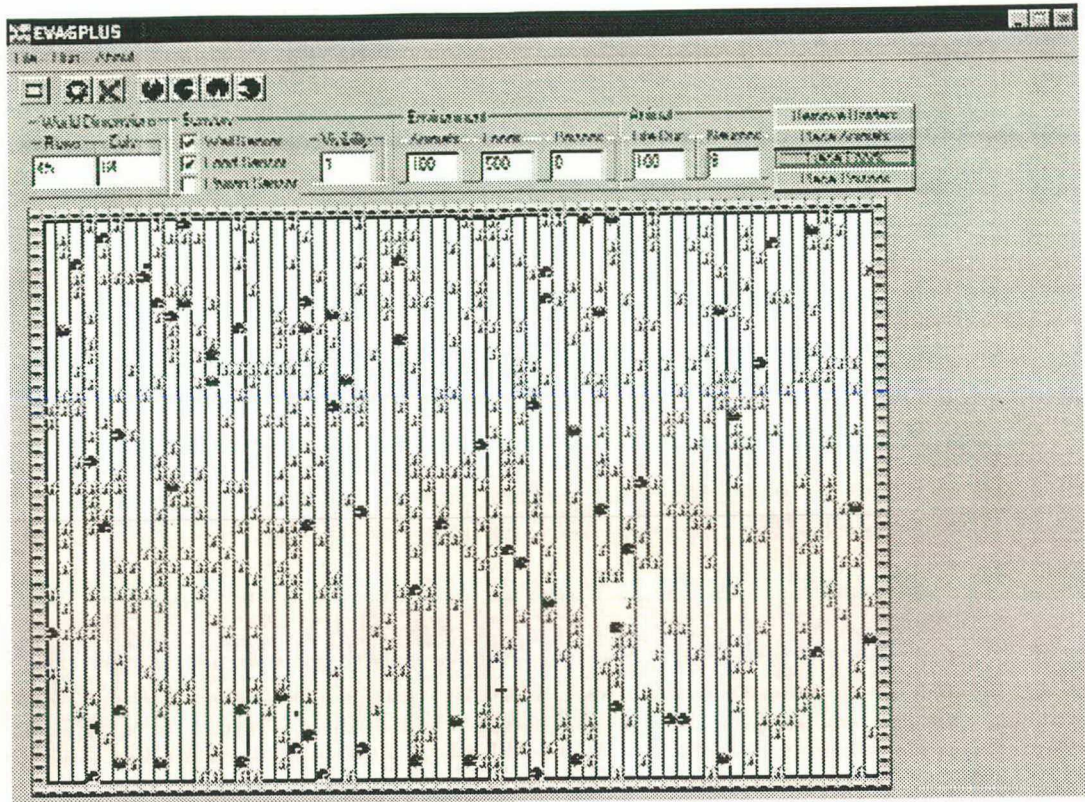


Figura 7.27: Ambiente cercado por paredes contendo agentes e alimentos aleatoriamente distribuídos

que a evolução simulada através de técnicas de programação evolucionária pode ser uma boa ferramenta para o desenvolvimento de AAs destinados a operar em ambientes em que, a princípio, se desconhece qual a complexidade mínima da rede neural que deva equipar estes agentes para sua correta operação no ambiente.

## 7.8 Uma Rede que Aprende Estados Internos

Segundo Angeline [10], um tópico de pesquisa bastante interessante e que tem atraído a comunidade conexionista é desenvolver modelos de redes neurais com capacidade de induzir autômatos finitos baseados apenas em exemplos positivos e negativos de “strings” ou cadeias, ou seja, de cadeias que pertencem e que não pertencem à linguagem reconhecida pelo autômato.

Uma coleção explícita de exemplos positivos e negativos, mostrados na Tabela 7.8, e que colocam dificuldades específicas para a indução da linguagem pretendida é apresentada em Tomita [182]. A caracterização das linguagens propostas

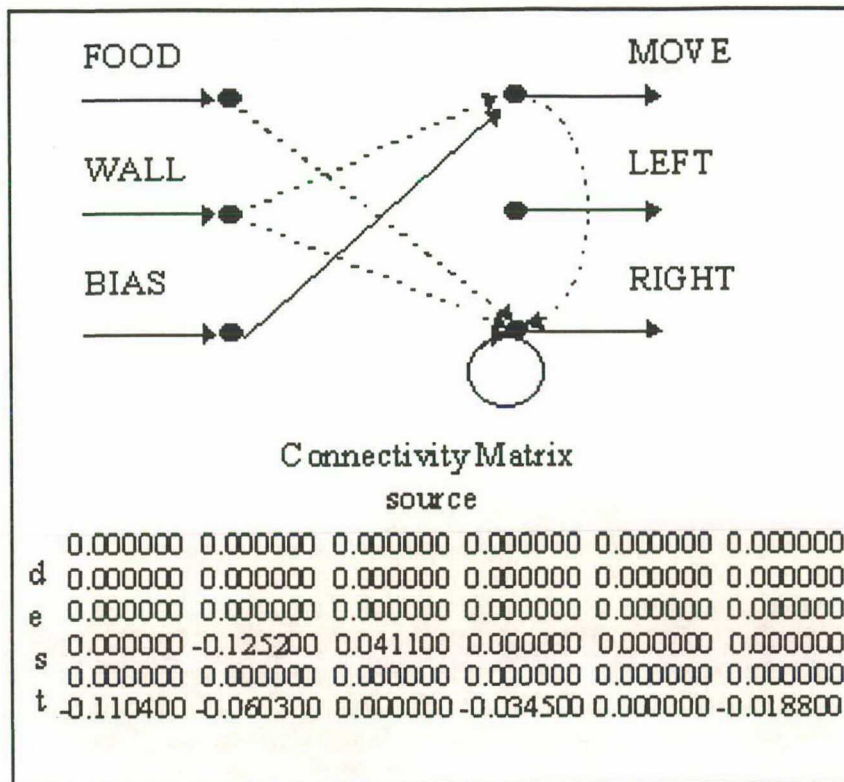


Figura 7.28: Rede Neural Recorrente encontrada pelo EVAG para um agente capaz de operar no ambiente cercado por paredes

por Tomita são apresentadas na Tabela 7.7. Vários estudos nesta área [143], [188], [73], [74] e [198], utilizam estas linguagens como parâmetro para teste de suas implementações. Repare que os conjuntos de treinamento não são balanceados, são incompletos e variam enormemente na sua capacidade de definir a linguagem regular pretendida.

Tomita forneceu também uma coleção explícita de exemplos positivos e negativos que podem ser vistos na Tabela 7.8.

A fim de possibilitar que a própria rede recorrente seja capaz de induzir os estados do autômato que reconhece a linguagem, algumas pequenas modificações foram efetuadas na arquitetura de rede neural composta originalmente proposta para a implementação de comportamentos. Agora, ao invés de ensinar para a rede os estados de um autômato de maneira explícita, fornece-se para a rede exemplos positivos e negativos de cadeias da linguagem que se deseja ensinar, com a respectiva indicação se a cadeia pertence ou não à linguagem desejada.

A partir do sistema RNNT já mostrado na seção 7.6.2, desenvolveu-se o sistema

Linguagem	Descrição
1	$1^*$
2	$(1\ 0)^*$
3	nenhuma cadeia com número ímpar de 0's é permitido depois de uma cadeia com número ímpar de 1's
4	não mais que dois 0's em uma cadeia
5	$(\text{número de 1's} - \text{número de 0's}) \bmod 3 = 0$
6	$0^*1^*0^*1^*$

Tabela 7.7: Caracterização de algumas das linguagens propostas por Tomita [182]

BIONNT (BIOlogical learning Neural Network Tool) para possibilitar o treinamento de redes neurais recorrentes com neurônios de estados na camada de saída utilizando o algoritmo de aprendizado biologicamente inspirado, como pode ser visto na Figura 7.29. A rede possui uma camada de entrada, uma camada intermediária de neurônios e, na camada de saída, existem neurônios que representam valores de saída real da rede e neurônios cuja função é representar os estados do autômato que a rede deve induzir. A saída destes “neurônios de estado” realimentam através de um conjunto de retardos todos os neurônios da camada intermediária, como pode ser visto na Figura 7.30.

Para estes exemplos, cada rede possui um único neurônio de entrada, um único neurônio de saída e um número de neurônios recorrentes configurável pelo usuário e que varia para cada exemplo, como apresentado na Tabela 7.10. Os três valores de entradas lógicas para estas linguagens, 0, 1, e nulo são representados por valores de ativação de -1, 1 e 0 respectivamente. Após a fase de treinamento, procede-se a fase de execução, para verificar se a rede induziu corretamente o autômato pretendido. Nesta fase, caso o valor de ativação do neurônio de saída apresente um valor positivo, considera-se a saída como 0.9; se for negativo, o valor de saída é considerado como -0.9.

Os valores dos “neurônios de estado” não são visíveis externamente à rede e, portanto, a única indicação de que o autômato por ela induzido está correto ou não,

Linguagem	Exemplos Positivos	Exemplos Negativos
1	$\xi$ , 1, 11, 111, 1111, 11111, 111111, 1111111, 11111111	0, 10, 01, 00, 011, 110, 000, 11111110, 10111111
2	$\xi$ , 10, 1010, 101010, 10101010, 10101010101010	1, 0, 11, 00, 01, 101, 100, 1001010, 10110, 110101010
3	$\xi$ , 1, 0, 01, 11, 00, 100, 110, 111, 000, 100100, 110000011100001, 111101100010011100	10, 101, 010, 1010, 110, 1011, 10001, 111010, 1001000, 11111000, 0111001101, 11011100110
4	$\xi$ , 1, 0, 10, 01, 00, 100100, 001111110100, 0100100100, 11100, 010	000, 11000, 0001, 000000000, 00000, 0000, 11111000011, 1101010000010111, 1010010001
5	$\xi$ , 10, 01, 1100, 111, 000000, 0111101111, 100100100	1, 0, 11, 00, 101, 011, 11001, 1111, 00000000, 010111, 10111101111, 1001001001
6	$\xi$ , 1, 0, 10, 01, 11111, 000, 00110011, 0101, 0000100001111, 00100, 011111011111, 00	1010, 00110011000, 0101010101, 1011010, 10101, 010100, 101001, 100100110101

Tabela 7.8: Exemplos positivos e negativos de algumas linguagens investigadas em Tomita [182].  $\xi$  significa a cadeia vazia

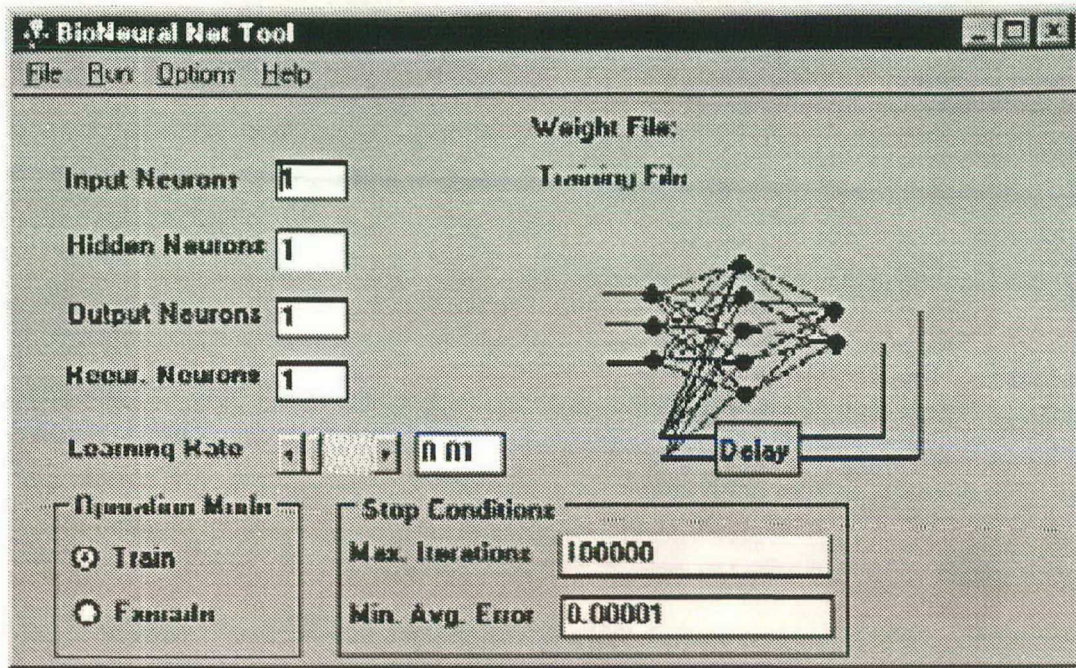


Figura 7.29: Interface de operação do sistema BIONNT

se dá através dos neurônios de saída que são comparados com o valor indicativo se a cadeia apresentada pertence ou não à linguagem.

O algoritmo de aprendizado utilizado neste sistema foi o algoritmo biologicamente inspirado já apresentado anteriormente. Os exemplos de cadeias positivas e negativas são apresentados na entrada da rede na seqüência em que são fornecidos no arquivo de dados preparado pelo usuário, um bit da cadeia de cada vez, e a rede é deixada para funcionar.

O erro para uma dada cadeia de sinais de entrada é computado apenas para a saída final da rede depois de toda a cadeia mais três caracteres de fim-de-cadeia 0 (nulo) terem sido apresentados à rede. A concatenação dos caracteres nulos é utilizada para identificar o fim-de-cadeia e permitir a entrada de cadeia nula como uma entrada válida para a rede. Este método também é usado por [9] e por [188].

Após todos os bits de uma cadeia terem sido apresentados, a saída apresentada pela rede é comparada com a saída desejada (0.9 se o exemplo é positivo, -0.9 se o exemplo é negativo), e um erro é calculado. Depois que todos os exemplos positivos e negativos forem apresentados, um valor global de erro é calculado. A partir daí, os pesos das conexões entre os neurônios são alterados pelo algoritmo, conforme já foi descrito, visando reduzir o valor do erro global a valores aceitáveis.



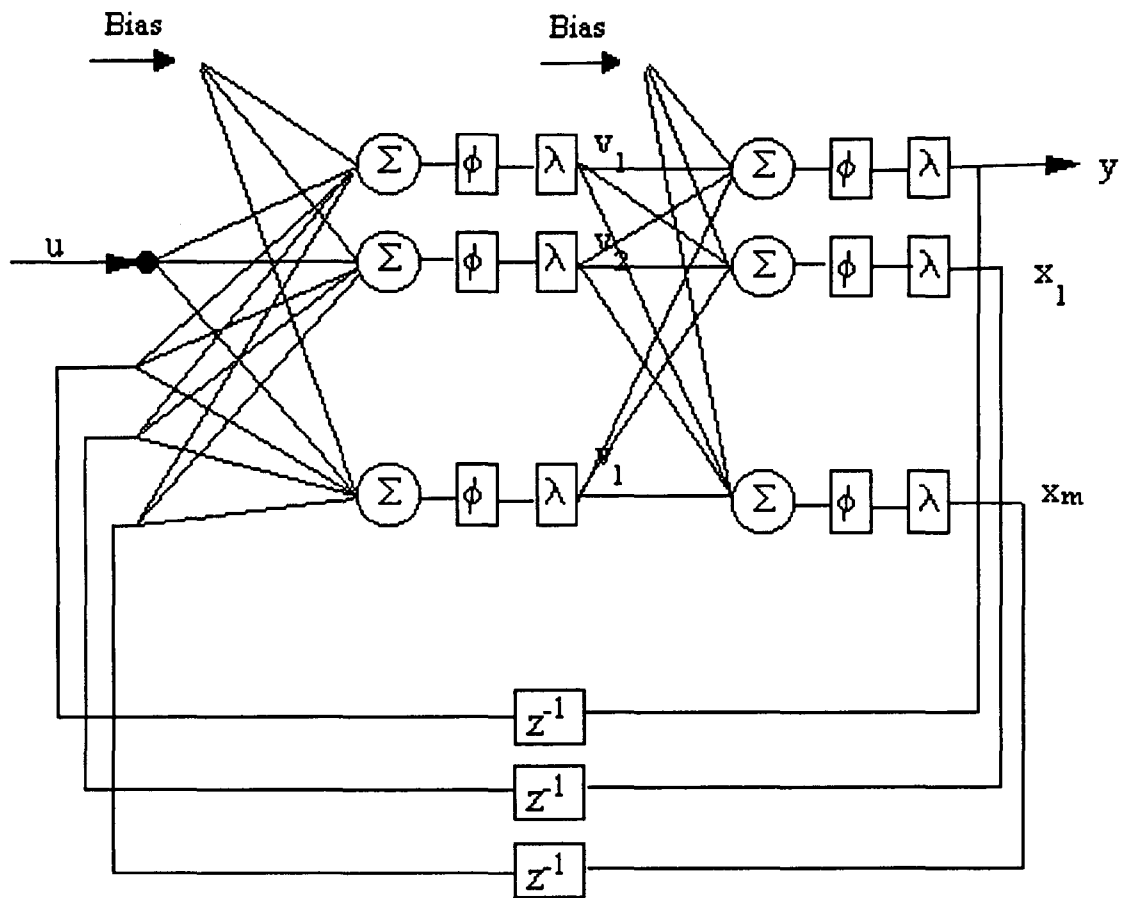


Figura 7.30: Rede Neural Recorrente com camada intermediária e neurônios de estado na camada de saída capaz de aprender linguagens regulares

Também aqui, em função dos valores dos pesos iniciais, o algoritmo pode conduzir a valores de mínimos locais, sendo então necessário escolher novos pesos iniciais para as conexões e recommençar o processo. É importante ressaltar também que a quantidade de neurônios de estado deve ser suficiente para poder representar todos os estados do autômato que reconhece a linguagem apresentada. Caso este número não seja suficiente a rede não será capaz de induzir todos os estados e o erro global também não atingirá valores aceitáveis.

Após os valores dos pesos da rede serem alterados pelo algoritmo de aprendizado de modo que o erro global apresente valores aceitáveis, procede-se a fase de execução; aqui a acuidade da rede é medida como a percentagem de acertos para um conjunto de 50 cadeias aleatórias de caracteres de tamanho 10 ou menor.

Os resultados obtidos pelo sistema BIONNT são apresentados na Tabela 7.10 e alguns gráficos mostrando a convergência da rede durante a fase de treinamento podem ser vistos nas Figuras 7.31-7.36. Estes resultados podem ser comparados

Linguagem	Épocas [188] (média)	Acuidade % [188] (melhor)	Épocas [9]	Acuidade % [9]
1	3033.8	100.0	3975	100.0
2	4522.6	100.0	5400	96.34
3	12326.8	78.31	25050	58.87
4	4393.2	60.92	15775	92.57
5	2137.6	46.21	21475	55.59
6	2969.0	55.74	12200	71.37

Tabela 7.9: Resumo dos resultados de velocidade e acuidade reportados por [188] e [9] para aprendizado das linguagens da Tabela 7.7

com os resultados obtidos por [9] e [188] para as mesmas linguagens e que são resumidos na Tabela 7.9. É importante ressaltar que tanto em [9] como em [188], a tolerância para o valor de saída foi de 0.1, apesar de [9] afirmar que em alguns dos seus experimentos, as suas redes neurais não foram capazes de convergir para este valor de tolerância especificado. No nosso caso, a tolerância considerada é de 0.9, ou seja, considera-se apenas o sinal do valor de ativação do neurônio de saída. Note-se também que o número de épocas necessárias para o treinamento das redes neurais no sistema BIONNT é consistentemente menor do que os apresentados por [9] e [188].

Fato interessante foi a baixa acuidade apresentada pela rede para o reconhecimento da linguagem 1 quando comparada com os resultados apresentados por [9] e [188]. Apesar do erro global obtido durante o treinamento ter chegado a valores mais do que aceitáveis, parece que o autômato induzido pela rede não é capaz de reconhecer totalmente a linguagem 1, ou seja, a capacidade de extrapolação apresentada pela rede foi baixa.

Uma forma de contornar este problema é acrescentar ao conjunto de exemplos do treinamento original as cadeias do conjunto de teste cujos resultados fornecidos pela rede não foram os corretos, procedendo-se então, ao retreinamento da rede. Neste caso, foram necessárias mais 160 épocas de treinamento para que o erro global para este novo conjunto de treinamento caísse a valores aceitáveis. Agora, ao ser novamente apresentada para o conjunto de teste, a acuidade da rede foi de 100%, indicando que os primeiros resultados se deveram a insuficiência de exemplos no conjunto de treinamento.

Apesar de nesta seção, o sistema BIONNT ter sido amplamente utilizado no

Linguagem	Configuração (neurônios de entrada, escondidos, de saída e de estado)	Épocas	Erro Global	Acuidade %
1	1,6,1,4	400	0.0005	76.0
2	1,10,1,6	150	0.001	88.0
3	1,8,1,6	250	0.033	74.0
4	1,8,1,10	92	0.01	58.0
5	1,8,1,10	150	0.014	68.0
6	1,8,1,10	160	0.018	66.0

Tabela 7.10: Resultados de configuração, velocidade, erro mínimo obtido ao final do aprendizado e acuidade das redes recorrentes para aprendizado das linguagens da Tabela 7.7

treinamento de RNAs capaz induzir autômatos para o reconhecimento de linguagens, as mesmas técnicas e conclusões podem ser empregadas para a implementação de comportamentos em agentes autônomos.

Suponha, por exemplo, um agente com apenas um sensor capaz de sentir a presença de algum elemento imediatamente à sua frente em um ambiente celular, codificado da seguinte maneira: -1 - não existe nada na célula à sua frente; +1 - existe um elemento na célula à sua frente. Este agente possui apenas um atuador cuja saída pode ser codificada da seguinte maneira: +1 - vire à direita; -1 - ande para a frente. No ambiente hipotético, um elemento venenoso está sempre colocado "atrás" de um alimento, entretanto o sensor do agente não consegue diferenciar um alimento de um veneno. O comportamento que deve ser ensinado ao agente é o andar para a frente enquanto não detectar nenhum elemento à sua frente e, quando algum elemento for detectado, andar uma casa a frente, para consumir o alimento, mas logo a seguir virar a direita, para evitar o veneno. Um possível conjunto de treinamento para este comportamento pode ser visto na Tabela 7.11. O sistema BIONNT foi usado para treinar uma rede recorrente com 1 neurônio de entrada, 10 neurônios na camada escondida, 1 neurônio de saída e 4 neurônios de estado. Após um período de treinamento de 370 épocas, o erro global chegou a 0.003 e a acuidade da rede para o mesmo conjunto de teste dos exemplos anteriores foi de 96%.

Os sistemas NNT, RNNT e BIONNT que implementam os algoritmos de aprendizado para redes recorrentes, tanto do tipo backpropagation, como o biologicamente inspirado, foram desenvolvidos em Visual Basic para o sistema operacional Windows. Seu objetivo principal, longe de se tornar um produto, é o de avaliar e testar

Sequência de Valores do Sensor	Valor do Atuador
-1	-1
+1	+1
+1 -1	-1
-1 +1	+1
+1 +1	-1
+1 -1 +1	+1
-1 -1 +1	+1
-1 +1 -1	-1
-1 +1 +1	-1
+1 -1 -1	-1
+1 -1 +1	+1
+1 +1 -1	-1
+1 +1 +1	-1
-1 -1 -1 +1	+1
-1 -1 +1 -1	-1
-1 -1 +1 +1	-1
+1 -1 +1 -1	-1
+1 -1 +1 +1	-1
-1 -1 -1 +1 -1 -1 +1	+1
+1 -1 +1 -1 +1	+1
+1 +1 +1 +1 +1 +1 +1 -1 +1	+1
-1 -1 -1 -1 -1 -1 -1 -1 +1	+1

Tabela 7.11: Conjunto de treinamento para implementar o comportamento de andar pelo ambiente evitando elementos de veneno colocados sempre “atrás” dos alimentos

as idéias apresentadas.

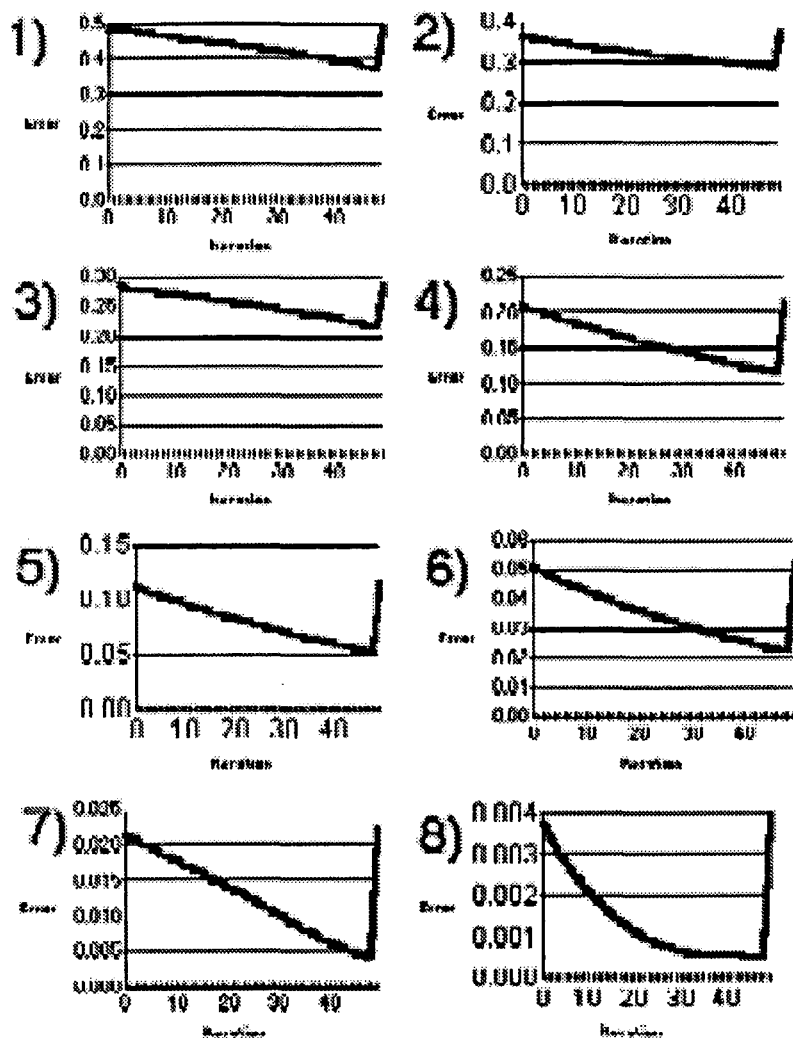


Figura 7.31: Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 1 da Tabela 7.8. Os gráficos são plotados a cada 50 épocas de treinamento

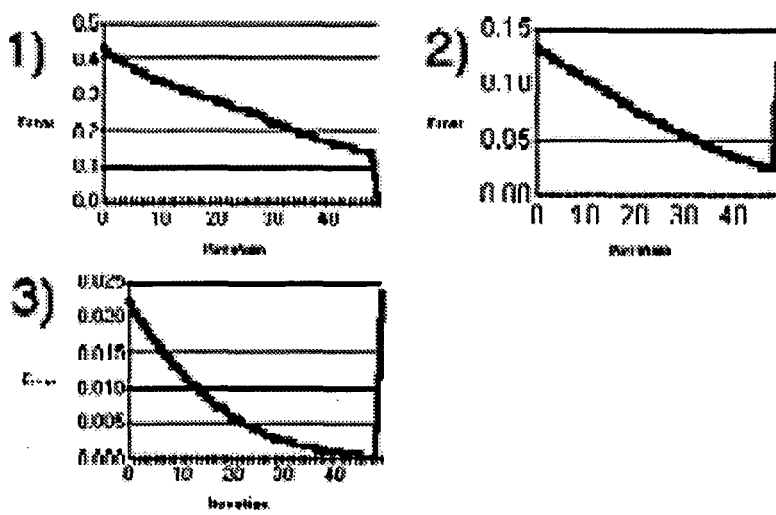


Figura 7.32: Gráficos mostrando a convergência da rede para o apredizado dos exemplos da linguagem 2 da Tabela 7.8

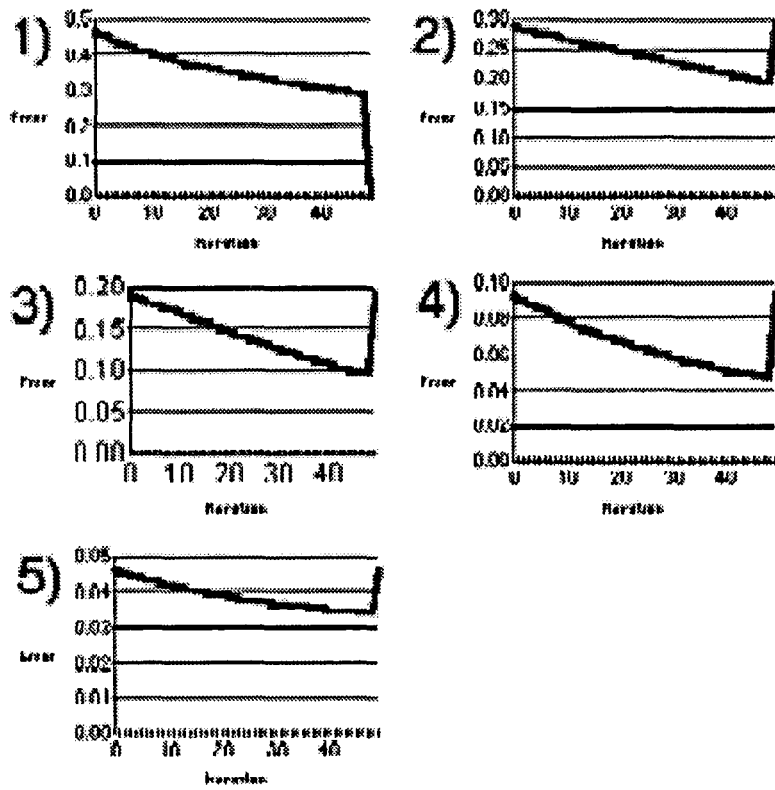


Figura 7.33: Gráficos mostrando a convergência da rede para o aprendizado dos exemplos da linguagem 3 da Tabela 7.8

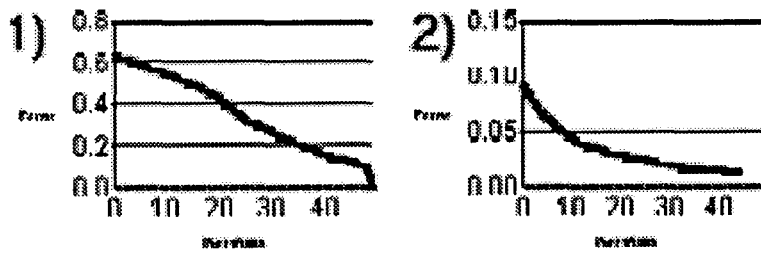


Figura 7.34: Gráficos mostrando a convergência da rede para o aprendizado dos exemplos da linguagem 4 da Tabela 7.8



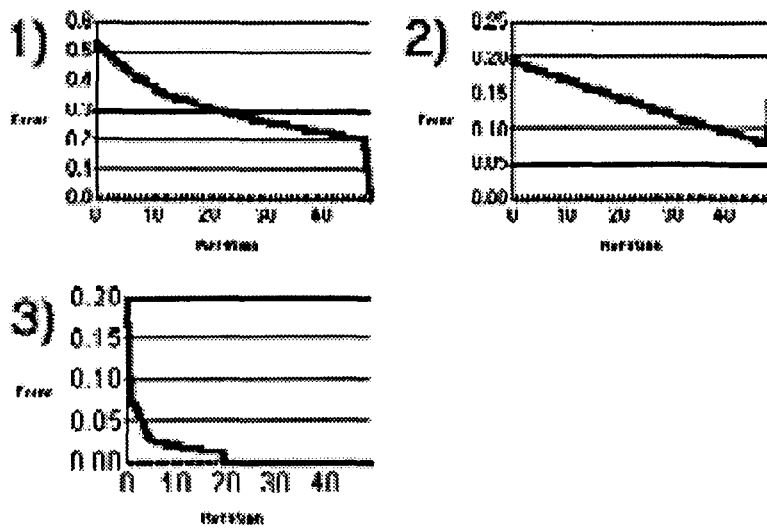


Figura 7.35: Gráficos mostrando a convergência da rede para o aprendizado dos exemplos da linguagem 5 da Tabela 7.8

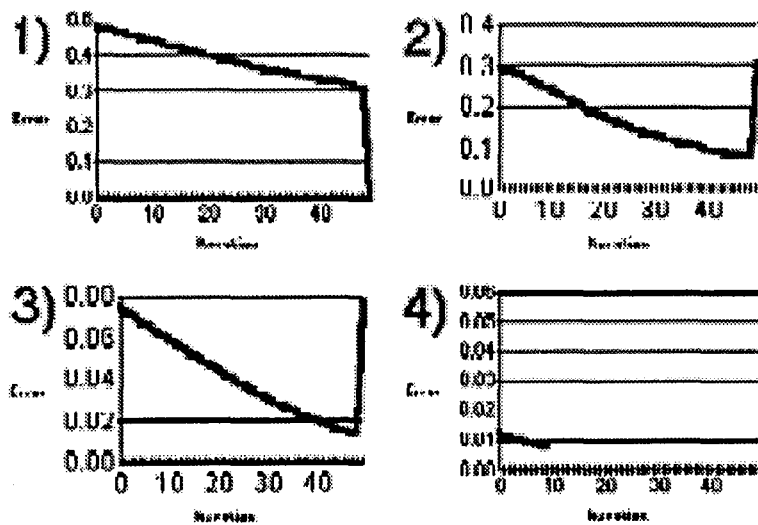


Figura 7.36: Gráficos mostrando a convergência da rede para o aprendizado dos exemplos da linguagem 6 da Tabela 7.8

# Capítulo 8

## Epílogo

### 8.1 Conclusões e Trabalhos Futuros

Este trabalho procurou abordar a problemática de se buscar ferramentas para o desenvolvimento de Agentes Autônomos, procurando fazer emergir aspectos de inteligência através de modelos computacionais inspirados em sistemas naturais. Para isto foram, primeiramente, reunidos uma série de subsídios, através de um levantamento bibliográfico extenso de várias áreas do conhecimento que se supôs serem pertinentes e esclarecedores. Assim, os principais aspectos abordados e que permitem estabelecer as bases sobre as quais se desenvolvem os pontos originais deste trabalho foram:

- Apresentou-se uma nova tendência na conceituação de '*Inteligência*' associada à idéia de sobrevivência e que se afasta dos conceitos tradicionais, altamente antropocêntricos, existentes desde os primórdios das pesquisas na área de Inteligência Artificial. Esta nova conceituação de Inteligência nos parece bem mais adequada quando se trabalha no desenvolvimento de Agentes Autônomos Inteligentes.
- Foram descritas algumas aplicações nas quais a utilização de Agentes Autônomos viria a facilitar a vida do seres humanos. Também foram ressaltadas as características de imprevisibilidade, dinamicidade e periculosidade do ambiente no qual o AA é projetado para operar.
- Ao ressaltar que a performance dos animais para operar e sobreviver nos ambientes descritos é muito superior a dos AAs até hoje desenvolvidos, mostra-se que esta é uma área aberta a uma série de pesquisas e sugere-se que um possível caminho a ser seguido é o de procurar na Natureza paradigmas para a implementação de AAs capazes de operar no mundo real.

- Foram apresentados os principais conceitos encontrados na literatura para AAs e procurou-se chegar a um conceito comum, abrangente e adequado. Também foram apresentadas e analisadas as principais alternativas existentes para a implementação de AAs. Observou-se que todas estas implementações apresentam limitações, porém as mais promissoras parecem ser as implementações baseadas em ‘comportamento reativo’.
- Uma vez justificada a filosofia geral do trabalho de tentar abstrair a dinâmica fundamental dos princípios que regem os fenômenos biológicos e procurar recriar estes fenômenos em um ambiente computacional, o trabalho aprofundou-se bastante em quatro temas considerados fundamentais de serem entendidos e manipulados para, ao serem recriados no ambiente computacional, servir como ferramenta para o desenvolvimento de AAs capazes de operarem a contento no mundo real. Estes temas são: o Comportamento Animal, o Aprendizado, a Evolução e o Sistema Nervoso.
- Procurou-se definir e classificar de maneira clara o Comportamento Animal, apresentando-se uma classificação para os vários tipos de comportamento baseada no aumento crescente de complexidade.
- Mostrou-se que o Autômato de Estados Finitos é uma ferramenta computacional bastante adequada para a descrição de uma grande variedade de classes de comportamentos e que facilita a implementação destes comportamentos em AAs.
- Um dos objetivos principais na tarefa de desenvolvimento de AAs, é poder dotar estes sistemas de um conjunto de comportamentos que os tornem aptos a realizar as tarefas a que foram projetados. Assim como nos animais, as chances de sobrevivência aumentam se estes AAs forem dotados da capacidade de selecionar, alterar e aprimorar seus comportamentos. A esta capacidade chamamos de Aprendizado.
- Assim, neste trabalho estudou-se o conceito de aprendizado tanto do ponto de vista psicológico como do ponto de vista da Inteligência Artificial (IA). Ao comparar-se as duas abordagens verificou-se que elas, de certa forma, são complementares, já que muitos dos mecanismos de aprendizado utilizados em IA, assim como o conceito de Inteligência, têm grande influência antropocêntrica e estão baseados na capacidade de representação simbólica do conhecimento, enquanto que boa parte das formas de aprendizado do ponto de vista psicológico

está ligada à mecanismos para alteração do comportamento de forma condicionada, através de reforços da estimulação sensorial. Por outro lado, não se seguiu as correntes psicológicas mais populares, tais como Piaget, Vygotsky e outros por serem abordagens holísticas do processo de aprendizado, enquanto que neste trabalho nossa abordagem foi reducionista.

- Ao procurar que alterações ocorrem à nível de sistema nervoso em função do aprendizado, viu-se que este é um mecanismo ainda não bem compreendido e ainda aberto a muitas pesquisas. A grosso modo, para os objetivos deste trabalho, trabalha-se com a hipótese que o aprendizado ocorre como uma alteração na força das conexões sinápticas entre os neurônios que compõem o sistema nervoso.
- Tratou-se da questão da evolução descrevendo, inicialmente, os mecanismos naturais da evolução e seleção natural, apresentando hipóteses de como este processo levou ao desenvolvimento do sistema nervoso e à evolução dos comportamentos.
- Assim, apresentou-se a idéia de que a complexidade dos sistemas nervosos nos seres vivos foi determinada através de processos evolucionários que se estenderam por milhões de anos; quanto mais complexo o ambiente se tornava, maior o repertório e a complexidade dos comportamentos necessários para sobreviver naquele ambiente. Assim, mostra-se, também, que a complexidade do comportamento apresentado varia com o grau de complexidade do sistema nervoso central, ou seja, comportamentos simples podem ser implementados com redes neurais simples (poucos neurônios e apenas conexões diretas), enquanto comportamentos complexos necessitam redes neurais mais complexas (muitos neurônios e grande número de conexões formando ciclos).
- Apresentou-se, também, os principais métodos computacionais inspirados nos mecanismos biológicos de evolução, os algoritmos evolucionários, e mostrou-se alternativas de utilização destes métodos para a geração de diferentes arquiteturas de RNAs.
- Ao estudar-se a estrutura biológica responsável pela emergência da inteligência capaz de determinar os comportamentos apresentados pelos animais, primeiramente apresentou-se os modelos biológicos de Sistema Nervoso Central, o funcionamento dos neurônios naturais e a organização interna do cérebro. Em seguida foram apresentados os modelos artificiais de neurônios e de RNAs,

tanto estáticas como dinâmicas, bem como os mecanismos de aprendizado utilizados.

- Relacionou-se as diferentes arquiteturas de redes neurais apresentadas com a capacidade de resolução de diferentes classes de problemas.

Diversos dos aspectos citados acima são áreas ainda bastante recentes de estudo e sua utilização, particularmente para o desenvolvimento de Agentes Autônomos, por si só já constitui aspecto inovador deste trabalho. Além disso, os principais pontos abordados encerrando características de originalidade foram:

- A utilização de Redes Neurais Artificiais para a implementação da entidade cognitiva de Agentes Autônomos é uma abordagem bastante recente. Neste trabalho mostrou-se que ela é uma alternativa bastante eficaz para a implementação de comportamentos e conta, ainda, com as vantagens apresentadas pela abordagem conexionista, quais sejam: alta imunidade ao ruído, tolerância a falhas, representação distribuída, ser 'ensinada' e não programada, etc.
- Ao apresentar-se as várias conceituações e alternativas de implementação de AAs, notou-se a inexistência de uma ferramenta formal que possibilitasse estabelecer parâmetros para a comparação entre as várias abordagens, bem como a necessidade de uma base teórica sólida que permitisse a definição e manipulação formal dos conceitos. Propôs-se, então, a formalização de alguns conceitos básicos fundamentais, tais como Agente, Ambiente, Comportamento, Grau de Autonomia e Algoritmo de Aprendizado, utilizando uma abordagem baseada na Teoria de Geral Sistemas.
- Examinou-se alguns pontos em que os métodos da computação evolucionária não são fiéis aos paradigmas biológicos. Sugeriu-se, então, um mecanismo de geração associando o número de neurônios e criação de conexões recorrentes a uma função de custo, de modo que, ao ser utilizado na geração de estruturas de redes neurais artificiais, assim como na Natureza, gerasse primeiramente alternativas de solução mais simples e econômicas.
- Estudou-se a capacidade de generalização das redes neurais artificiais, demonstrando-se que esta capacidade varia em função da estrutura da rede. Redes neurais diretas com neurônios estáticos são capazes de interpolar, mas não de extrapolar, enquanto que redes neurais recorrentes possuem a capacidade de extrapolação, o que aumenta em muito a sua potencialidade.

- Introduziu-se e fundamentou-se a idéia de que uma série de classes de comportamentos observados podem ser implementados através de diferentes arquiteturas de RNAs, ou seja, comportamentos simples podem ser implementados com redes neurais simples (poucos neurônios e apenas conexões diretas), enquanto comportamentos complexos necessitam redes neurais mais complexas (muitos neurônios e grande número de conexões formando ciclos). Realizou-se alguns experimentos que demonstraram a validade desta hipótese.
- Apresentou-se duas alternativas para o aprendizado em redes neurais recorrentes, uma baseada no algoritmo tradicional de backpropagation e outra biologicamente inspirada no paradigma de aprendizado por reforço. Foram feitos alguns experimentos e analisados vantagens e limitações das duas alternativas.
- Fez-se um análise sobre as potencialidades computacionais das diferentes topologias de RNAs. Mostrou-se que, assim como redes neurais diretas com neurônios estáticos que não possuam camada de neurônios intermediários não são capazes de aprender padrões não-linearmente separáveis e que qualquer rede direta com neurônios estáticos é incapaz de modelar sistemas dinâmicos, também redes neurais recorrentes sem camada intermediária de neurônio são incapazes de aprender certa classe de autômatos finitos.
- A seleção, análise e implementação de um algoritmo evolucionário capaz de gerar redes neurais foi um dos objetivos deste trabalho. Este algoritmo evolucionário possui como características principais: a capacidade de geração de diversas topologias de redes neurais, não só diretas, como também recorrentes; as redes por ele geradas possuem um grau crescente de complexidade, tanto na quantidade de neurônios como na estrutura gerada.
- Por fim, sugere-se uma arquitetura de rede neural que, juntamente com o algoritmo de aprendizado biologicamente inspirado proposto, é capaz de induzir autômatos finitos baseados apenas em exemplos positivos e negativos de 'strings', ou seja, de 'strings' que pertencem e que não pertencem à linguagem reconhecida pelo autômato.

Entretanto, apesar de vários pontos terem sido abordados e resultados concretos apresentados, várias questões permanecem em aberto e muitos estudos e experimentos ainda devem ser realizados em buscas de novos dados e idéias.

- Até este momento, nos experimentos desenvolvidos, implementou-se apenas um comportamento de cada vez. Outro ponto a ser examinado, diz respeito

a como implementar um conjunto de comportamentos, alguns inclusive antagônicos e que agem sobre o mesmo conjunto de atuadores como, por exemplo, ir em direção ao alimento e fugir de um predador.

- Investigar a possibilidade da existência de uma estrutura neural hierárquica em que comportamentos mais complexos ocupam graus mais elevados na hierarquia e podem influenciar e mesmo suprimir os comportamentos mais simples que estão implementados nos níveis inferiores da hierarquia.
- No algoritmo de aprendizado para redes neurais biologicamente inspirado proposto, o treinamento da rede se dá em um estágio distinto. O que se observa na Natureza é que não existe uma separação distinta entre a fase de aprendizado e a fase de execução. Um tópico a ser trabalhado é o desenvolvimento de AAs que possam ir continua e progressivamente aprendendo, mesmo durante o período de operação, de modo a que possam se adaptar a novas situações e mudanças no ambiente.
- Mais experiências devem ser feitas no sentido da utilização da arquitetura neural e do algoritmo de aprendizado biologicamente inspirado para indução dos estados internos. Deve-se investigar como a rede efetua o mapeamento entre os estados induzidos com possíveis comportamentos, através unicamente da relação entre os estímulos recebidos pelos sensores e a ação gerada pelos atuadores, sem que haja necessidade de especificar explicitamente a máquina de estados finitos que descreve o comportamento.
- Utilizando-se a formalização do conceito de Agente Autônomo e de Ambiente baseada na Teoria Geral de Sistemas, estudos comparativos entre as várias alternativas de implementação, usando uma métrica comum pode ser efetuada.
- Sabe-se que sem a capacidade de pensamento simbólico não se pode alcançar estágios de comportamento racional. Uma possibilidade a ser investigada diz respeito a investigar da possibilidade de se implementar processamento simbólico em RNAs e tentar fazer emergir aspectos de comportamento racional em AAs.

A metodologia empregada neste trabalho foi a de desenvolver exemplos simples que visem comprovar e testar as idéias apresentadas, sem a pretensão de desenvolvimento de um produto comercial ou mesmo de um robô real.

Entretanto, como produto final desta tese, obteve-se uma série de ferramentas que podem auxiliar na tarefa de desenvolvimento de AAs com alto grau de autonomia e inteligência.

Uma destas ferramentas é um sistema baseado em algoritmos evolucionários que gera a estrutura e o conjunto de valores de conexões de uma rede neural artificial capaz de implementar o conjunto de comportamentos necessários para operar em um dado ambiente.

Outra ferramenta se constitui de um sistema que, uma vez determinada a estrutura da rede neural artificial, executa um algoritmo de aprendizado biologicamente inspirado capaz de implementar comportamentos em Agentes, seja através da implementação do autômato finito que descreve o comportamento, seja através de exemplos do Comportamento Observável desejado do agente, deixando que a própria rede induza um autômato capaz de implementá-lo.



# Referências Bibliográficas

- [1] Y. S. Abu-Mostafa. The vavnik-chervonenkis dimension? information versus complexity in learning. *Neural Computation*, 3(1):312–317, 1989.
- [2] Y. S. Abu-Mostafa and J. St.Jaques. Information capacity of the hopfield model. *IEEE Trans. on Inform. Theory*, 31(4):461–464, July 1985.
- [3] D. H. Ackley and M. L. Littman. Interactions between learning and evolution. In *Artificial Life II*, p. 487–509. Addison-Wesley, 1992.
- [4] ACM. *Communications of the ACM*, 37(7), July 1994.
- [5] P. E. Agre and D. Chapman. What are plans for? In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, A Bradford Book, p. 17–34. The MIT Press, Cambridge, Massachusetts, 1990.
- [6] J. A. Anderson. Neural-network learning and mark twain’s cat. *IEEE Communications Magazine*, p. 16–23, Sept. 1992.
- [7] R. W. Anderson. Learning and evolution: A quantitative genetics approach. *J. theor. Biol.*, (175):89–101, Aug. 1995.
- [8] T. L. Anderson and M. Donath. Animal behavior as a paradigm for developing robot autonomy. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, A Bradford Book, p. 145–168. The MIT Press, Cambridge, Massachusetts, 1990.
- [9] P. Angeline, G. Saunders, and J. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65, 1994.
- [10] P. J. Angeline. *Evolutionary Algorithms and Emergent Intelligence*. PhD thesis, The Ohio State University, 1993.

- [11] M. A. Arbib and E. G. Manes. *Arrows, structures and functors: The categorical imperative*. Academic Press, Inc., New York, 1975.
- [12] R. C. Arkin. Intelligent robotic systems. *IEEE Expert*, p. 6–8, Apr. 1995.
- [13] J. W. Atmar. *Speculation on the Evolution of Intelligence and Its Possible Realization in Machine Form*. ScD thesis, New Mexico State University, Las Cruces, 1976.
- [14] E. Barnard and L. F. A. Wessels. Extrapolation and interpolation in neural network classifiers. *IEEE Control Systems*, p. 50–53, Oct. 1992.
- [15] J. M. Barreto. Modeling and simulation in biomedicine. Text of the Courses Applications Biomédicales des Méthodes des Simulation I et II, UCL, Belgium.
- [16] J. M. Barreto. Considerações genéricas sobre otimização de sistemas. Relatório Técnico 2, Dept. de Matemática e Informática, Instituto Militar de Engenharia, Rio de Janeiro, 1973.
- [17] J. M. Barreto. Uma definição de otimização de sistemas. *Rev. Circ. Eng. Mil.*, 76(S):179–183, 1976.
- [18] J. M. Barreto. Neural networks learning: A new programming paradigm. In *ACM International Conference: Trends and Directions in Expert Systems*, Florida, USA, Oct. 1990. ACM.
- [19] J. M. Barreto. *Conexionismo e a Resolução de Problemas*. Dissertação para concurso para professor titular, Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, Florianópolis, SC, 1996.
- [20] J. M. Barreto and M. de Neyer. Qualitative and quantitative models of systems. In S. Tzafestas, R. Hanus, and P. Kool, editors, *Mathematical Modelling in Science and Technology*, p. 269–274. J. C. Baltezer AG, Scientific Pub. Co., 1991.
- [21] J. M. Barreto, M. de Neyer, P. Levèfre, and R. Gorez. Qualitative physics versus fuzzy sets theory in modeling and control. In *IECON'91: IEEE International Conference on Industrial Electronics*, p. 1651–1656, Kobe, Japan, oct/nov 1991.
- [22] J. M. Barreto and M. D. Neyer. Qualitative and quantitative models and simulation. Technical report, (RFO/AI/14/UCLLN/90/7, Louvain-la-Neuve, Belgium, 1990.

- [23] J. M. Barreto and R. L. Pagano. The role of qualitative simulation in computer assisted learning. In *ICTE-91: International Conference in Technology and Education*, p. 569–571, Toronto, Canada, May 1991.
- [24] J. M. Barreto, M. Roisenberg, and F. M. de Azevedo. Developing artificial neural networks for autonomous agents using evolutionary programming. In *ASC'98 IASTED International Conference on Artificial Intelligence and Soft Computing*, p. 283–286, Cancun, Mexico, May 1998.
- [25] A. G. Barto. Connectionist learning for control. In W. T. Miller III, R. S. Sutton, and P. J. Werbos, editors, *Neural Networks for Control*, p. 5–58. The MIT Press, Cambridge, Massachusetts, 1990.
- [26] E. B. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1(1):151–160, 1989.
- [27] R. D. Beer. A dynamical systems perspective on autonomous agents. Technical Report CES-92-11, Case Western Reserve University, Department of Computer Engineering and Science, 1992.
- [28] R. D. Beer, H. J. Chiel, and L. S. Sterling. A biological perspective on autonomous agent design. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, A Bradford Book, p. 169–186. The MIT Press, Cambridge, Massachusetts, 1990.
- [29] U. Beyer and F. J. Śmieja. Janus: A society of agents. Technical Report 840, GMD - Gesellschaft für Mathematik und Datenverarbeitung MBH, St. Augustin, Germany, May 1994.
- [30] E. J. W. Boers and H. Kuiper. Biological metaphors and the design of modular artificial neural networks. MSc dissertation, Leiden University, the Netherlands, Departments of Computer Science and Experimental and Theoretical Psychology, Aug. 1992.
- [31] R. P. Bonasso. Underwater experiments using a reactive system for autonomous vehicles. In *Proc. of the Ninth National (USA) Conference on Artificial Intelligence*, p. 794–800, Menlo Park, CA, July 1991. The MIT Press.
- [32] P. Bourgin and F. J. Varela. Introduction: Towards a practice of autonomous systems. In F. J. Varela and P. Bourgin, editors, *Proc. of the First European Conference on Artificial Life*, A Bradford Book, p. xi–xvii, Paris, Dec. 1991. The MIT Press.

- [33] L. M. Brasil. Uma proposta de arquitetura para sistema especialista híbrido e a correspondente metodologia de elicitação/representação do conhecimento. Exame de qualificação para doutorado, UFSC, Engenharia Elétrica, 1996.
- [34] R. A. Brooks. A robust layered control system for a mobile robot. A. I. Memo 864, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Sept. 1985.
- [35] R. A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. A. I. Memo 1091, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Feb. 1989.
- [36] R. A. Brooks. The behavior language; user's guide. A. I. Memo 1227, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Apr. 1990.
- [37] R. A. Brooks. Elephants don't play chess. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, A Bradford Book, p. 3–16. The MIT Press, Cambridge, Massachusetts, 1990.
- [38] R. A. Brooks. Artificial life and real robots. In F. J. Varela and P. Bourguine, editors, *Proc. of the First European Conference on Artificial Life*, A Bradford Book, p. 3–10, Paris, Dec. 1991. The MIT Press.
- [39] R. A. Brooks. Intelligence without reason. A. I. Memo 1293, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Apr. 1991.
- [40] J. Buhmann and H. Kuhnel. Complexity optimized data clustering by competitive neural networks. *Neural Computation*, 5(3):75–88, 1993.
- [41] T. H. Bullock, R. Orkand, and A. Grinnell. *Introduction to Nervous System*. Freeman, San Francisco, CA, 1977.
- [42] A. B. Bulsari and H. Saxén. A recurrent neural network model. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks*, number 2, p. 1091–1094. North-Holland, Amsterdam, 1992.
- [43] T. Burrows and M. Niranjan. The use of feed-forward and recurrent networks for system identification. Technical Report CUED F-INFENG TR158, Cambridge University Engineering Dept., Cambridge U., Cambridge, UK, Dec. 1993.

- [44] J. Carbonell and P. Langley. Machine learning. In Shapiro, editor, *Encyclopedia of Artificial Intelligence*, number 3. John Wiley, New York, 1987.
- [45] E. B. Carne. *Artificial Intelligence Techniques*. Spartan Books, Washington, DC, 1965.
- [46] M. Casey. Computation dynamics in discrete-time recurrent neural networks. In *Proceedings of the Annual Research Symposium*, volume 3, p. 78–95, UCSD, La Jolla, CA, 1993. Institute for Neural Computation.
- [47] P. M. Churchland. *Matter and conciousness: a contemporary introduction to the philosophy of mind*. A Bradford Book. The MIT Press, Cambridge, Massachusetts, second edition, 1984.
- [48] A. Cleeremans, D. Servan-Schreiber, and J. McClelland. Finite state automata and simple recurrent neural networks. *Neural Computation*, 1(3):372–381, 1989.
- [49] P. R. Cohen and E. A. Feigenbaum, editors. *Learning and Inductive Inference*. Number 3. Addison-Wesley, Reading, Massachusetts, Sept. 1986.
- [50] F. Córdova et al. Alternativas de automatización para el guiado autónomo de vehículos cargadores frontales en una mina subterránea. *Automática e Innovación*, 1(2):67–73, 1994.
- [51] N. E. Cotter. The stone-weierstrass theorem and its application to neural networks. *IEEE Transactions on Neural Networks*, 1(4):290–295, 1990.
- [52] G. Cybenko. Continuous valued networks with two hidden layers are sufficient. Technical report, Tufts University, Department Science Department, 1988.
- [53] S. S. da Costa Botelho e D. Barone. Implementação do condicionamento comportamental am robôs móveis autônomos. In *III Congresso Brasileiro de Redes Neurais*, Curitiba, PR, Out. 1995.
- [54] J. B. da Mota Alves et al. Ficção, realidade e expectativa de robôs inteligentes baseados em comportamento. In *Anais do 1. Simpósio Brasileiro de Automação Inteligente*, p. 145–154, Rio Claro, SP, 1993.
- [55] J. da Silva Dias. Treinamento híbrido de redes neurais para processamento de informações biomédicas. Exame de qualificação para doutorado, UFSC, Engenharia Elétrica, 1996.

- [56] C. Darwin. *On the Origin of Species*. Grolier Enterprises Corp., Danbury, Connecticut, 1859.
- [57] L. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
- [58] M. S. Dawkins. *Explicando o Comportamento Animal*. Editora Manole, São Paulo, Brasil, 1989.
- [59] F. M. de Azevedo. *Contribution to the Study of Neural Networks in Dynamical Expert Systems*. PhD thesis, Institut d'Informatique, FUNDP, Namur, Belgium, 1993.
- [60] A. B. de Holanda Ferreira. *Minidicionário da Língua Portuguesa*. Editora Nova Fronteira, Rio de Janeiro, 1988. 2a. Edição.
- [61] V. G. Dethier and E. Stellar. *Comportamento Animal*. Textos básicos de biologia moderna. Edgard Blücher, Ed. da Universidade de São Paulo, São Paulo, Brasil, 1973.
- [62] R. J. M. et al. The capacity of hopfield associative memory. *IEEE Trans. on Inform. Theory*, 33(4):461–482, 1987.
- [63] J. A. Fabro e F. Gomide. Controle neural/nebuloso auto-organizável de veículos autônomos. In *III Congresso Brasileiro de Redes Neurais*, Curitiba, PR, Out. 1995.
- [64] Y. Fang and T. Sejnowski. Faster learning for dynamic recurrent backpropagation. *Neural Computation*, 2:270–273, 1990.
- [65] R. J. Firby. An investigation into reactive planning in complex domains. In *Proc. 6th National Conference on Artificial Intelligence*, p. 202–206, San Mateo, CA, July 1987. Morgan Kaufmann.
- [66] D. B. Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence*. IEEE Press, New York, 1995.
- [67] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966.
- [68] J. A. Freeman and D. M. Skapura. *Neural Networks: algorithms, applications, and programming techniques*. Computation and Neural Systems Series. Addison-Wesley, Reading, Massachusetts, 1992.

- [69] R. Garcia, F. M. de Azevedo, and J. M. Barreto. Genetic algorithms in the optimal choice of neural networks for signal processing. In *38 IEEE Midwest Symposium on Circuits and Systems*, Rio de Janeiro, Brasil, Ago. 1995.
- [70] M. P. Georgeffand and A. L. Lansky. Reactive reasoning and planning. In *Proc. 6th National Conference on Artificial Intelligence*, p. 677–682, San Mateo, CA, July 1987. Morgan Kaufmann.
- [71] C. Giles and B. Horne. Representation and learning in recurrent neural network architectures,. In K. Narendra, editor, *Proceedings of the Eighth Yale Workshop on Adaptive and Learning Systems*, Center for Systems Science, Dunham Laboratory, Yale University New Haven, CN, 1994.
- [72] C. Giles, G. Kuhn, and R. Williams. Dynamic recurrent neural networks: Theory and applications. *IEEE Transactions on Neural Networks*, 5(2):153–156, 1994.
- [73] C. L. Giles et al. Higher order recurrent networks & grammatical inference. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, number 2, p. 380–387. Morgan Kaufmann, San Mateo, CA, 1990.
- [74] C. L. Giles et al. Extracting and learning an unknown grammar with recurrent neural networks. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, number 4, p. 317–324. Morgan Kaufmann, San Mateo, CA, 1992.
- [75] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.
- [76] A. A. Gorni. Redes neurais artificiais - uma abordagem revolucionária em inteligência artificial. *Microsistemas*, 13(133):14–25, Jul. 1994.
- [77] R. A. Grupen et al. Distributed control representation for manipulation tasks. *IEEE Expert*, p. 9–14, Apr. 1995.
- [78] A. Guyton. *Textbook of Medical Physiology*. W. B. Sanders Company, Philadelphia, 1976.
- [79] S. Haykin. *Neural Networks - A Comprehensive Foundation*. IEEE Press, New York, 1994.
- [80] D. O. Hebb. *The Organization of Behavior*. John Willey, New York, 1949.

- [81] D. O. Hebb. *Psicologia*, volume 1. Livraria Atheneu, Rio de Janeiro, Brasil, 1979. 3a. Edição.
- [82] R. Hecht-Nielsen. *Neurocomputers*. Addison-Wesley, Reading, Massachusetts, 1990.
- [83] J. Heitkoetter and D. Beasley (eds.). Guide to evolutionary computation (faq in comp.ai.genetic). <http://www.cefetpr.br/encore/FAQ>, 1996.
- [84] B. R. Hergenhahn. *An Introduction to Theories of Learning*. Prentice-Hall, Englewood Cliffs, New Jersey, second edition, 1982.
- [85] T. Hessburg and M. Tomizuka. Fuzzy logic control for lateral vehicle guidance. *IEEE Control System*, p. 55–63, Aug. 1994.
- [86] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [87] B. Irie and S. Miyake. Capabilities of three-layered perceptrons. In *Proceedings of the IEEE International Conference on Neural Networks*, number Vol. I, p. 641–648, San Diego, CA, 1988. IEEE.
- [88] D. Jefferson et al. Evolution as a theme in artificial life: The genesys/tracker system. In C. Langton, C. Taylor, J. Farmer, and S. Rasmussen, editors, *Artificial Life II*, p. 549–578. Addison-Wesley, Reading, Massachusetts, 1992.
- [89] R. Johnson (ed.). The software agents mailing list faq. <http://www.sml.com/research/tcl/lists/agents-list.html>, 1996.
- [90] L. P. Kaelbling. *Learning in Embedded Systems*. A Bradford Book. The MIT Press, Cambridge, Massachusetts, 1993.
- [91] L. P. Kaelbling and S. J. Rosenchein. Action and planning in embedded agents. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, A Bradford Book, p. 35–48. The MIT Press, Cambridge, Massachusetts, 1990.
- [92] M. Kaiser et al. Using machine learning techniques in real-world mobile robots. *IEEE Expert*, p. 37–45, Apr. 1995.
- [93] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in Mathematical System Theory*. McGraw-Hill, New York, 1969.



- [94] N. Karunanithi, R. Das, and D. Whitley. Genetic cascade learning for neural networks. In L. D. Whitley and J. D. Schaffer, editors, *Proc. of the International Workshop on Combinations of Genetic Algorithms and Neural Networks COGANN'92*, Los Alamitos, CA, 1992. IEEE Computer Society Press.
- [95] G. Kechriotis, E. Zervas, and E. Manolakos. Using recurrent neural networks for adaptive communication channel equalization. *IEEE Transactions on Neural Networks*, 5(2):267–278, 1994.
- [96] G. A. Kimble. *Hilgard and Marquis' conditioning and learning*. Prentice-Hall, Englewood Cliffs, New Jersey, second edition, 1961.
- [97] G. Kiss. Autonomous agnets, ai and chaos theory. In J. A. Meyer and S. W. Wilson, editors, *From Animals to Animats*. The MIT Press, Cambridge, Massachusetts, 1991.
- [98] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, third edition, 1989.
- [99] B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.
- [100] J. R. Koza. Evolution and co-evolution of computer programs to control independently-acting agents. In *Proceedings First International Conference on Simulation of Adaptive Behavior*, p. 366–375, Paris, 1990. The MIT Press.
- [101] J. R. Koza. Evolution of subsumption using genetic programming. In F. J. Varela and P. Bourguine, editors, *Proc. of the First European Conference on Artificial Life*, A Bradford Book, p. 110–119, Paris, Dec. 1991. The MIT Press.
- [102] U. Kramer. On the application of fuzzy sets to the analysis of the system driver-vehicle-environment. *Automatica*, 21(1):101–107, 1985.
- [103] C. Langton et al. *Artificial Life II*. Addison-Wesley, Reading, Massachusetts, 1991.
- [104] A. Laprade and G. Lambert-Torres. Controle de um veículo utilizando a teoria dos conjuntos nebulosos. In *Anais do 1. Simpósio Brasileiro de Automação Inteligente*, p. 166–174, Rio Claro, SP, 1993.
- [105] A. Leite, J. B. da Mota Alves, e M. F. Resende. Sistema para depuração de um robô móvel autônomo inteligente. In *Anais do 1. Simpósio Brasileiro de Automação Inteligente*, p. 175–184, Rio Claro, SP, 1993.

- [106] L. Lin. Programming robots using reinforcement learning and teaching. In *Proc. of the Ninth National (USA) Conference on Artificial Intelligence*, p. 781–786, Menlo Park, CA, July 1991. The MIT Press.
- [107] A. Lindenmayer and G. Rozenberg, editors. *Automata, Languages, Development*, Amsterdam, 1976. North-Holland Publishing Company.
- [108] R. K. Lindsay. Artificial evolution of intelligence. *Contemp. Psychology*, 13(3):113–116, 1966.
- [109] R. Liscano et al. Using a blackboard to integrate multiple activities and achieve strategic reasoning for mobile-robot navigation. *IEEE Expert*, p. 24–36, Apr. 1995.
- [110] K. Lorenz. The comparative method in studying innate behaviour patterns. In *Symposium of the Society for Experimental Biology*, number 4, p. 221–268, 1950.
- [111] T. B. Ludermir. Learnability and computability in artificial neural networks. In *1º SBAI Simpósio Brasileiro de Automação Inteligente*, p. 1–9, Rio Claro, SP, 1993.
- [112] P. Maes. Situated agents can have goals. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, A Bradford Book, p. 49–70. The MIT Press, Cambridge, Massachusetts, 1990.
- [113] P. Maes. Learning behavior networks from experience. In F. J. Varela and P. Bourguine, editors, *Proc. of the First European Conference on Artificial Life*, A Bradford Book, p. 48–58, Paris, Dec. 1991. The MIT Press.
- [114] P. Maes. Modeling adaptive autonomous agents. [pattie@media.mit.edu](mailto:pattie@media.mit.edu), MIT, USA, 1995.
- [115] S. Mahadevan and J. Connell. Automatic programming of behaviour-based robots using reinforcement learning. *Artificial Intelligence*, (55):311–365, 1992.
- [116] A. Mamdani. Foundation for intelligent physical agents. <http://www.raleigh.ibm.com/iagfipa.html>, 1996.
- [117] A. Manning. Evolution of behavior. In J. L. MacGaugh, editor, *Psychobiology*, p. 1–52. Academic Press, New York, 1971.

- [118] A. Manning. *An Introduction to Animal Behavior*. Addison-Wesley, Reading, Massachusetts, third edition, 1979.
- [119] E. Mayr. Evolution. *Scientific American*, p. 2-13, Sept. 1978.
- [120] E. Mayr. *Toward a New Philosophy of Biology: Observations of an Evolutionist*. Belknap Press, Cambridge, Massachusetts, 1988.
- [121] J. L. McClelland, D. E. Rumelhart, and PDP Group. *Parallel Distributed Processing*, volume 2, Psychological and Biological Models. The MIT Press, Cambridge, Massachusetts, 1986.
- [122] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged locomotion. *Computer Graphics*, 24(4):29-38, 1990.
- [123] M. D. Mesarovic and Y. Takahara. *General Systems Theory: Mathematical Foundations*. Academic Press, 1975.
- [124] J. B. Messenger. *Nervos, Cérebro e Comportamento*, volume 22 of *Temas de Biologia*. Ed. da Universidade de São Paulo, São Paulo, Brasil, 1980.
- [125] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolutionary Programs*. Springer-Verlag, Berlin, 1992.
- [126] M. L. Minsky. *The Society of Mind*. Simon and Schuster, New York, 1985.
- [127] M. L. Minsky and S. A. Papert. *Perceptrons: an introduction to computational geometry*. The MIT Press, Cambridge, Massachusetts, 1969.
- [128] M. L. Minsky and S. A. Papert. *Perceptrons: an introduction to computational geometry - Expanded Edition*. The MIT Press, Cambridge, Massachusetts, 1988.
- [129] D. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In *Proc. of the Eleventh International Joint Conference on Artificial Intelligence*, p. 762-767, San Mateo, CA, 1989. Morgan Kaufmann Publishers Inc.
- [130] J. F. Montgomey, A. H. Fagg, and G. A. Bekey. The USC AFV-I: A behavior-based entry in the 1994 international aerial robotic competition. *IEEE Expert*, p. 16-22, Apr. 1995.

- [131] J. L. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. L. Moody and S. J. Hanson, editors, *Neural Information Processing Systems*, number 4. Morgan Kaufmann Publishers Inc., 1992.
- [132] H. Muhlenbein. Parallel genetic algorithms and neural networks as learning machines. In D. J. Evans, G. R. Joubert, and H. Liddell, editors, *Proc. of the International Conference Parallel Computing '91*, p. 91–103, London, UK, 3-6 sep 1991. North-Holland.
- [133] H. Muhlenbein and J. Kindermann. *The Dynamics of Evolution and Learning - Towards Genetic Neural Networks*. German National Research Center for Computer Science, 1989.
- [134] P. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions in Neural Networks*, 1(1):4-27, 1990.
- [135] C. L. Nascimento Jr. *Artificial Neural Networks in Control and Optimization*. PhD thesis, Control Systems Centre, University of Manchester, Manchester, UK, 1994.
- [136] S. M. Nassar. *Sistema Estatístico Inteligente para apoio a Pesquisas Médicas*. Tese de doutorado, Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica, Florianópolis, SC, Dez. 1995.
- [137] C. W. Omlin and C. L. Giles. Constructing deterministic finite-state automata in recurrent neural networks. Technical Report CS 94-3, Rensselaer Polytechnic Institute Computer Science, Troy, NY, Revised April 1995.
- [138] D. Parker. Learning logic. Invention Report S81-64, File 1, Office of Technology Licensing, Stanford University, 1982.
- [139] M. J. Patel and U. Schnepf. Concept formation as emergent phenomena. In F. J. Varela and P. Bourguine, editors, *Proc. of the First European Conference on Artificial Life*, A Bradford Book, p. 11–20, Paris, Dec. 1991. The MIT Press.
- [140] J. Piaget. *Adaptation and intelligence - organic selection and phenocopy*. The University of Chicago Press, 1980.

- [141] P. Piggott and A. Sattar. Reinforcement learning of iterative behavior with multiple sensors. *Journal of Applied Intelligence*, (4):351–365, 1994.
- [142] J. B. Pollack. Book review - perceptrons: An introduction to computational geometry, expanded edition - marvin l. mynsky and seymour a. papert. pollack.perceptrons.ps PostScript file at cheopo.cis.ohio-state.edu.
- [143] J. B. Pollack. The induction of dynamical recognizers. *Machine Learning*, (7):227–252, 1991.
- [144] V. W. Porto, D. Fogel, and L. Fogel. Alternative neural networks training methods. *IEEE Expert*, 10(3):16–22, June 1995.
- [145] A. T. R. Pozo. *Um Sistema de Ensino Inteligente Via Sociedade de Multiagentes Aplicado ao Diagnóstico de Epilepsia*. Tese de doutorado, Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica, Florianópolis, SC, Fev. 1995.
- [146] D. L. Prado. New learning algorithm for training multilayered neural networks that uses genetic-algorithm techniques. *Electronics Letters*, 28(16):1560–1561, 1992.
- [147] N. Rashevsky. The principle of adequate design. In R. Rosen, editor, *Foundations of Mathematical Biology vol III: Supercellular Systems*, p. 143–175. Academic Press, New York and London, 1973.
- [148] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologische Evolution [Evolutionary Strategy: Optimization of Technical Systems According to the Principles of Biological Evolution]*. Frommann Holzboog Verlag, Stuttgart, Germany, 1973.
- [149] J. L. Ribeiro Filho, P. C. Treleaven, and C. Alippi. Genetic-algorithm programming environments. *IEEE Computer*, p. 28–43, June 1994.
- [150] E. Rich and K. Knight. *Inteligência Artificial*. Makron Books, São Paulo, Brasil, 1993. 2a. Edição.
- [151] M. Roisenberg, J. M. Barreto, e F. M. de Azevedo. Generalization capacity in neural networks - the ballistic trajectory learning case. In *II Congresso Brasileiro de Redes Neurais*, Curitiba, PR, Out. 1995.

- [152] M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Biological inspirations in neural network implementations of autonomous agents. In D. L. Borges and C. A. A. Kaestner, editors, *Advances in artificial intelligence : proceedings / 13th Brazilian Symposium on Artificial Intelligence*, number 1159 in lecture notes in computer science; Lecture notes in artificial intelligence, p. 211–220. Springer-Verlag, Berlin, Oct. 1996.
- [153] M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. A neural network that implements reactive behaved autonomous agents. In *AEN'96 IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, Honolulu, Hawaii, Aug. 1996.
- [154] M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Specialization versus generalization in neural network learning for ballistic interception movement. In *MELECON'96 IEEE Mediterranean Elettrotechnical Conference*, p. 627–630, Bari, Italy, May 1996.
- [155] M. Roisenberg, J. M. Barreto, e F. M. de Azevedo. Feedforward and recurrent neural networks complexity power - a comparison based on a concrete example. In *III Congresso Brasileiro de Redes Neurais*, p. 1–6, Florianópolis, SC, Julho. 1997.
- [156] M. Roisenberg, J. M. Barreto, e F. M. de Azevedo. Modeling behaviors with artificial neural networks. In *WRI'97 Workshop de Robótica Inteligente*, p. 34–45, Brasília, DF, Agosto. 1997.
- [157] M. Roisenberg, J. M. Barreto, e F. M. de Azevedo. Uma proposta de modelização para agentes autônomos baseada na teoria de sistemas. In *3º SBAI Simpósio Brasileiro de Automação Inteligente*, p. 500–507, Vitória, ES, Set. 1997.
- [158] M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. Neural network classification based on the problem. In *IJCNN IEEE International Joint Conference on Neural Networks*, Anchorage, Alaska, May 1998.
- [159] M. Roisenberg, J. M. Barreto, and F. M. de Azevedo. On a formal concept of autonomous agents. In *AI'98 IASTED International Conference on Applied Informatics*, Garmisch-Paterkirchen, Germany, Feb. 1998.
- [160] R. Rosen. Is there a unified mathematica biology? In R. Rosen, editor, *Foundations of Mathematical Biology vol III: Supercellular Systems*, p. 361–393. Academic Press, New York and London, 1973.

- [161] D. E. Rumelhart, J. L. McClelland, and PDP Group. *Parallel Distributed Processing*, volume 1, Foundations. The MIT Press, Cambridge, Massachusetts, 1986.
- [162] D. E. Rumelhart, B. Widrow, and M. A. Lehr. The basic ideas in neural networks. *Communications of the ACM*, 37(3):86–92, Mar. 1994.
- [163] H. Saxén and B. Saxén. A tool for modeling, simulation and prediction using feedforward and recurrent neural networks. In *Anais do I Simpósio Brasileiro de Redes Neurais*, p. 55–60, Caxambu, MG, Ago. 1994.
- [164] E. E. Scalabrin, L. Vanderberghe, H. de Azevedo, e J. A. Barthès. A generic model of cognitive agent to develop open systems. In *SBIA '96 XIIIth Brazilian Symposium on Artificial Intelligence*, p. 61–70, Curitiba, PR, Out. 1996.
- [165] R. J. Schalkoff. *Artificial Intelligence: An Engineering Approach*. McGraw-Hill, New York, 1990.
- [166] E. Schrödinger. *What is life?* Cambridge University Press. Combined reprint with *Mind and Matter* 1989, 1944.
- [167] J. T. Schwartz. *The new connectionism: developing relationships between neuroscience and artificial intelligence*. 1989.
- [168] H.-P. Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, UK, 1981.
- [169] A. SIGART. The first international conference on autonomous agents '97 call for papers. <http://www.doc.mmu.ac.uk/aa97.html>, 1996.
- [170] P. K. Simpson. *Artificial Neural Systems - foundations, paradigms, applications and implementations*. Neural Networks: Research and Applications. Pergamon Press, Elmsford, NY, 1990.
- [171] F. J. Śmieja and U. Beyer. Janus: A robot manipulator system implemented on a blackboard architecture. Technical Report 839, GMD - Gesellschaft für Mathematik und Datenverarbeitung MBH, St. Augustin, Germany, May 1994.
- [172] J. M. Smith. Evolution of behavior. *Scientific American*, p. 92–101, Sept. 1978.
- [173] T. Smithers. Taking eliminative materialism seriously: A methodology for autonomous systems research. In F. J. Varela and P. Bourguine, editors, *Proc.*

- of the First European Conference on Artificial Life*, A Bradford Book, p. 31–40, Paris, Dec. 1991. The MIT Press.
- [174] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. The Systems Programming Series. Addison-Wesley, Reading, Massachusetts, 1984.
- [175] M. Srinivas and L. M. Patnaik. Genetic algorithms: A survey. *IEEE Computer*, p. 17–26, June 1994.
- [176] L. Steels. Towards a theory of emergent functionality. In J. A. Meyer and S. W. Wilson, editors, *From Animals to Animats*. The MIT Press, Cambridge, Massachusetts, 1991.
- [177] M. Sugeno, M. F. Griffin, and A. Bastian. Fuzzy hierarchical control of an unmanned helicopter. Presented at IEEE Int'l Conf. Fuzzy Systems; disponível através de ftp anonymous da máquina cs.arizona.edu no directório /japan/kahaner.reports com o nome copt-fuz.93, 1993.
- [178] G. Taga et al. Generation and coordination of bipedal locomotion through global entrainment. In *Proceedings on International Symposium on Autonomous Decentralized Systems*, 1993.
- [179] J. Tanomaru. Motivação, fundamentos e Aplicações de algoritmos Genéticos. In *III Congresso Brasileiro de Redes Neurais*, Curitiba, PR, Out. 1995.
- [180] W. H. Thorpe. *Learning and instinct in animals*. Harvard University Press, Cambridge, Massachusetts, second edition, 1963.
- [181] N. Tinbergen. *The Study of Instinct*. Oxford University Press, 1951.
- [182] M. Tomita. Dynamic construction of finite automata from examples using hill-climbing. In *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, p. 105–108, Ann Arbor, Michigan, 1982.
- [183] F.-S. Tsung. *Modeling Dynamical Systems with Recurrent Networks*. PhD thesis, University of California at San Diego, San Diego, CA, 1994.
- [184] J. Vaario. *An Emergent Modeling Method for Artificial Neural Networks*. Doctor dissertation of engineering, Aeronautic and Astronautic Engineering Course, The University of Tokyo, Japan, 1993.



- [185] F. J. Vico and F. Sandoval. Use of genetic algorithms in neural networks definition. In A. Prieto, editor, *Proc. of the International Workshop on Artificial Neural Networks IWANN'91*, p. 196–203, Granada, Spain, 17-19 Sep 1991. Springer-Verlag.
- [186] L. von Bertalanfy. *General Systems Theory*. Penguin Books, Middlesex, 1968.
- [187] R. A. Wallace. *The Ecology and Evolution of Animal Behavior*. Goodyear Publishing Company, Santa Monica, California, second edition, 1979.
- [188] R. L. Walthous and G. M. Kuhn. Induction of finite-state automata using second-order recurrent networks. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, number 4, p. 309–316. Morgan Kaufmann, San Mateo, CA, 1992.
- [189] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.
- [190] B. Webb and T. Smithers. The connection between ai and biology in the study of behavior. In F. J. Varela and P. Bourguine, editors, *Proc. of the First European Conference on Artificial Life*, A Bradford Book, p. 421–428, Paris, Dec. 1991. The MIT Press.
- [191] P. J. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, Cambridge, MA, 1974.
- [192] P. J. Werbos. Backpropagation through time: What it does and how to do it. *Proceeding of the IEEE*, 78(10):1550–1560, Oct. 1990.
- [193] N. Wiener. *Cybernetics*. The MIT Press, Massachusetts, 1945.
- [194] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.
- [195] D. Wood. Grammar and l forms: An introduction. In G. Goos and J. Hartmanis, editors, *Lecture Notes in Computer Science*, Berlin, 1980. Springer-Verlag.
- [196] M. Wooldridge and N. R. Jennings, editors. *Intelligent Agents - Theories, Architectures, and Languages*. Number 890 in Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, Jan. 1995.
- [197] L. Zadeh and C. Desoer. *Linear System Theory, A State Space Approach*. McGraw-Hill, New York, 1963.

- [198] Z. Zeng, R. M. Goodman, and P. Smith. Learning finite state machines with self-clustering recurrent network. *Neural Computation*, 5(6):976–990, 1993.
- [199] D. Zisper. A subgrouping strategy that reduce complexity and speeds up learning in recurrente networks. *Neural Computation*, 1(4):552–558, 1989.