

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

EDUARDO NUNES FERREIRA BASTOS

**Uma Rede Neural Auto-Organizável
Construtiva para Aprendizado Perpétuo de
Padrões Espaço-Temporais**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Paulo Martins Engel
Orientador

Porto Alegre, agosto de 2007

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Bastos, Eduardo Nunes Ferreira

Uma Rede Neural Auto-Organizável Construtiva para Aprendizado Perpétuo de Padrões Espaço-Temporais / Eduardo Nunes Ferreira Bastos. – Porto Alegre: PPGC da UFRGS, 2007.

137 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2007. Orientador: Paulo Martins Engel.

1. Ciência cognitiva corporificada. 2. Redes neurais artificiais. 3. Auto-organização. 4. Representação temporal. 5. Aprendizado construtivo. I. Engel, Paulo Martins. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cesar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^a. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenadora do PPGC: Prof^a. Luciana Porcher Nedel

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Dedico este trabalho à minha família e à minha namorada Karina
por estarem presente em todos os momentos da minha vida

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
LISTA DE TABELAS	14
LISTA DE ALGORITMOS	15
RESUMO	16
ABSTRACT	17
1 INTRODUÇÃO	18
1.1 Contexto do trabalho	18
1.2 Motivação principal	21
1.3 Objetivo geral e objetivos específicos	22
1.4 Posicionamento, justificativa e relevância	22
1.5 Contribuições relevantes	24
1.6 Estrutura da dissertação	24
2 FUNDAMENTOS DE CIÊNCIAS COGNITIVAS	26
2.1 Introdução	26
2.2 Abordagens tradicionais das ciências cognitivas	27
2.2.1 Cognitivismo	27
2.2.2 Conexionismo	28
2.2.3 Projeto de agentes de <i>software</i> tradicionais	28
2.3 Abordagens contemporâneas das ciências cognitivas	29
2.3.1 Teoria Ecológica	29
2.3.2 Teoria da Autopoiese	29
2.3.3 Teoria da Enação	30
2.3.4 Teoria Psicológica Construtivista	30
2.3.5 Teoria da Seleção de Grupos Neurais	30
2.3.6 Projeto de agentes de <i>software</i> contemporâneos	33
2.4 Considerações do capítulo	40
3 FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS	41
3.1 Introdução	41
3.2 O neurônio biológico	42
3.3 O neurônio artificial	43

3.4	A rede neural de múltiplos neurônios	45
3.5	A rede neural auto-organizável de Kohonen	47
3.5.1	Processo de competição	49
3.5.2	Processo de cooperação	49
3.5.3	Processo de adaptação	50
3.5.4	Métricas de precisão e preservação de topologia	51
3.5.5	Propriedades do mapa de características	51
3.6	Considerações do capítulo	52
4	TRABALHOS CORRELATOS	54
4.1	As redes neurais auto-organizáveis temporais	54
4.1.1	Processamento de seqüências temporais: definições e propriedades	55
4.1.2	Memórias de curto prazo	55
4.1.3	Time-Delayed Self-Organizing Map (TDSOM)	64
4.1.4	Differentiator-Integrator Neuron (DIN)	64
4.1.5	Temporal Kohonen Map (TKM)	64
4.1.6	Recurrent Self-Organizing Map (RSOM)	65
4.1.7	Contextual Self-Organizing Map (CSOM)	65
4.1.8	Sequential Activation Retention and Decay NETwork (SARDNET)	66
4.1.9	Adaptive Resource Allocating Vector Quantization (ARAVQ)	66
4.1.10	Resumo	67
4.2	As redes neurais auto-organizáveis construtivas	68
4.2.1	Growing Cell Structures (GCS)	69
4.2.2	Incremental Grid Growing (IGG)	70
4.2.3	Growing Grid (GG)	70
4.2.4	Growing Neural Gas (GNG)	71
4.2.5	Adaptive Resonance Theory (ART)	72
4.2.6	Resumo	72
4.3	Considerações do capítulo	73
5	PROPOSTA DE REDE NEURAL ARTIFICIAL	74
5.1	Introdução	74
5.2	Estrutura geral da rede neural	75
5.3	Funcionamento da rede neural	79
5.3.1	Processo de inicialização	80
5.3.2	Processo de competição	81
5.3.3	Processo de cooperação	81
5.3.4	Processo de adaptação	82
5.3.5	Processo de inclusão e remoção de neurônios	83
5.3.6	Processo de inclusão, adaptação e remoção de arestas	86
5.4	Resumo do algoritmo	87
5.5	Notação matemática	87
5.6	Análise paramétrica	88
5.7	Inspiração biológica	90
5.8	Discussão	91
5.9	Considerações do capítulo	93

6	RESULTADOS EXPERIMENTAIS	95
6.1	Metodologia	95
6.2	Quantização vetorial	96
6.2.1	Distribuição bidimensional de dados	96
6.2.2	Distribuição não-linearmente separável de dados	101
6.2.3	Distribuição multimodal de dados	102
6.2.4	Distribuição não-estacionária de dados	103
6.2.5	Distribuição não-uniforme de dados	105
6.3	Segmentação de seqüências temporais	106
6.3.1	Distribuição espaço-temporal de dados em forma de “8”	106
6.3.2	Distribuição espaço-temporal de dados em forma de circunferência	113
6.4	Considerações do capítulo	116
7	CONCLUSÕES E TRABALHOS FUTUROS	117
7.1	Sobre o trabalho	117
7.2	Trabalhos futuros	118
	REFERÊNCIAS	120
	APÊNDICE A FERRAMENTA DE SIMULAÇÃO ROBÓTICA	130
A.1	Introdução	130
A.2	Ambiente de simulação	131
A.3	Interface de programação de aplicativos	133
A.4	Robôs	133
A.5	Sensores robóticos	134
A.6	Efetuadores robóticos	137
A.7	Considerações do apêndice	137

LISTA DE ABREVIATURAS E SIGLAS

ADALINE	Adaptive Linear Element	46
AHIGG	Adaptive Hierarchical Incremental Grid Growing	70
API	Application Programming Interface	133
ARAVQ	Adaptive Resource Allocating Vector Quantization	66
ART	Adaptive Resonance Theory	47
ART	Adaptive Resonance Theory	72
BPTT	Backpropagation Through Time	21
CasCor	Cascade-Correlation	21
CHL	Competitive Hebbian Learning	71
CPN	Counterpropagation	47
CR	Conditioned Response	38
CSOM	Contextual Self-Organizing Map	65
CS	Conditioned Stimuli	38
DAC	Distributed Adaptive Control	137
DASH	Dynamic Adaptive Self-Organising Hybrid Model	72
DIN	Differentiator-Integrator Neuron	64
DNA	Deoxyribonucleic Acid	26
DOF	Degree of Freedom	137
EBA	Extended Braitenberg Architecture	137
GCS	Growing Cell Structures	69
GDCS	Growing Dynamical Cell Structure	69
GG	Growing Grid	70
GHSOM	Growing Hierarchical Self-Organizing Map	71
GNG	Growing Neural Gas	71
GPL	General Public License	130
GSOM	Growing Self-Organizing Map	70

GTSOM	Growing Temporal Self-Organizing Map	74
GWR	Grow When Required	71
HiGS	Hierarchical Growing Cell Structures	69
HSOM	Habituating Self-Organizing Map	75
IA	Inteligência Artificial	18
IC	Inteligência Computacional	24
IGG	Incremental Grid Growing	70
LTM	Long-Term Memory	105
MADALINE	Multiple ADALINE	46
MSOM	Merging Self-Organizing Map	66
NG	Neural Gas	71
PSOM	Plastic Self-Organizing Map	72
RNA	Redes Neurais Artificiais	20
RSOM	Recurrent Self-Organizing Map	65
RTRL	Real-Time Recurrent Learning	21
SARDNET	Sequential Activation Retention and Decay NETWORK	66
SMC	Sensory-Motor Coordination	137
SOM	Self-Organizing Map	47
SOMSD	Self-Organizing Map for Structured Data	66
STM	Short-Term Memory	55
TDSOM	Time-Delayed Self-Organizing Map	64
TKM	Temporal Kohonen Map	64
TNGS	Theory of Neuronal Group Selection	30
TreeGCS	Tree Growing Cell Structures	70
UR	Unconditioned Response	38
US	Unconditioned Stimuli	38

LISTA DE FIGURAS

Figura 1.1:	A trilogia do agente de <i>software</i> inteligente idealizado.	19
Figura 1.2:	A trilogia da rede neural artificial idealizada.	20
Figura 2.1:	A visão tradicional do processamento de informação	28
Figura 2.2:	A visão contemporânea do processamento de informação	33
Figura 3.1:	Elementos básicos de um neurônio biológico	42
Figura 3.2:	Unidade de limiar linear de McCulloch e Pitts	43
Figura 3.3:	Exemplos de funções de ativação. À esquerda, função linear. No centro, função rampa. À direita, função sigmóide (tangente hiperbólica).	45
Figura 3.4:	Perceptron de múltiplas camadas de Frank Rosenblatt	46
Figura 3.5:	Elementos básicos de um mapa auto-organizável de Kohonen. Um padrão de entrada x de dimensão m é fornecido para a rede no instante t . Dentre todos os l neurônios da rede, um deles é considerado o vencedor v . A vizinhança h_{vi} é dada pelos neurônios mais próximos ao neurônio vencedor. Apenas o vencedor e a sua vizinhança têm os pesos sinápticos modificados, de forma que fiquem ainda mais próximos do padrão de entrada.	48
Figura 4.1:	Representação gráfica de uma memória de atraso k	56
Figura 4.2:	Função <i>kernel</i> da memória de atraso unitário ($k = 1$).	57
Figura 4.3:	Representação gráfica de uma memória de traço exponencial.	58
Figura 4.4:	Função <i>kernel</i> da memória de traço exponencial para $\lambda = 0.6, 0.7, 0.8, \text{ e } 0.9$	58
Figura 4.5:	Representação gráfica da memória Gama.	59
Figura 4.6:	Função <i>kernel</i> da memória Gama para $\tau = 6$ e $\lambda = 0.3, 0.5, \text{ e } 0.7$	59
Figura 4.7:	Representação gráfica da memória diferenciadora-integradora.	60
Figura 4.8:	Função <i>kernel</i> da memória diferenciadora-integradora para $\mu = 0.0$ e $\eta = 0.1$; $\mu = 0.2$ e $\eta = 0.4$; $\mu = 0.6$ e $\eta = 0.4$; $\mu = 1.0$ e $\eta = 0.4$	61
Figura 4.9:	Representação gráfica da unidade de habituação	62
Figura 4.10:	(A) Sinal temporal de entrada $x(t)$ aplicado às memórias de curto prazo. (B) Resposta de uma memória de linhas de atraso de parâmetro $\tau = 3$. (C) Resposta de uma memória de traço exponencial de parâmetro $\lambda = 0.9$. (D) Resposta de uma memória gama de parâmetros $\tau = 6$ e $\lambda = 0.4$. (E) Resposta de uma memória diferenciadora-integradora de parâmetros $\mu = 0.6$ e $\eta = 0.4$. (F) Resposta de uma unidade de habituação de parâmetros $\alpha = 0.1$ e $\beta = 0.9$	63

Figura 5.1:	Representação gráfica da unidade de memorização.	76
Figura 5.2:	Representação gráfica da unidade de habituação.	77
Figura 5.3:	Elementos básicos da rede neural proposta.	78
Figura 5.4:	Mapeamento Φ implementado pela rede neural proposta.	79
Figura 5.5:	Ajuste de pesos sinápticos na rede neural proposta. Os números reais indicam os valores das sinapses de habituação de cada neurônio da rede. À esquerda, configuração da rede antes da atualização sináptica. No centro, um padrão \mathbf{z} é apresentado à rede e o primeiro e segundo neurônios vencedores, v e s , respectivamente, são selecionados. À direita, os vetores de pesos \mathbf{w}^z do neurônio vencedor v e da vizinhança topológica N_v são ajustados em direção ao padrão \mathbf{z} , as sinapses de habituação w^h do neurônio vencedor v e da vizinhança topológica N_v são atualizadas em direção ao ponto de máximo habituação (valor zero) e as sinapses de habituação dos demais neurônios da rede são modificadas rumo ao estado inicial de não habituação (valor 1).	83
Figura 5.6:	Inserção de um novo neurônio na rede neural proposta. À esquerda, configuração da rede antes da inserção. No centro, um padrão espaço-temporal \mathbf{z} é apresentado à rede e o primeiro e segundo neurônios vencedores, v e s , respectivamente, são selecionados. À direita, um novo neurônio de índice r é adicionado na rede entre o neurônio vencedor v e o padrão espaço-temporal \mathbf{z} e a conexão existente entre v e s é realocada.	84
Figura 5.7:	Remoção de neurônios na rede neural proposta. Os números reais indicam os valores das sinapses de habituação de cada neurônio. À esquerda, configuração da rede antes da remoção de neurônios. No centro, os neurônios que têm o valor da sinapse de habituação próximo de 1 são removidos da rede (e.g., $w_i^h \geq 0.95$). À direita, os neurônios que não possuem nenhuma conexão sináptica são removidos da rede (i.e., $N_i = \emptyset$).	85
Figura 5.8:	Inserção, atualização e remoção de conexões sinápticas na rede neural proposta. Os números inteiros representam os contadores de idade de cada conexão sináptica da rede. À esquerda, configuração da rede em um instante de tempo arbitrário. No centro, uma nova conexão é adicionada na rede entre os neurônios vencedores v e s e o contador de idade dessa conexão é feito igual a zero. Ademais, as idades de todas as conexões existentes entre o neurônio vencedor v e a sua vizinhança topológica N_v , exceto neurônio s , são incrementadas em uma unidade. À direita, as conexões sinápticas que possuem uma idade maior que o limiar i_{max} são removidas da rede (e.g., $i_{max} = 50$).	86
Figura 5.9:	À esquerda a triangulação de Delaunay e à direita o diagrama de Voronoi resultante para uma distribuição arbitrária de 20 pontos.	89
Figura 6.1:	Arquitetura simplificada da rede aplicada à tarefa de quantização vetorial.	96

Figura 6.2:	Simulação de uma distribuição de dados em forma de quadrado com $\rho = 0.75$ e $\delta = 0.1$. Ao todo são 1000 amostras uniformemente distribuídas dentro da área em cinza claro. (a) Distribuição de densidade dos dados de entrada. (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.	97
Figura 6.3:	Simulação de uma distribuição de dados em forma de círculo com $\rho = 0.8$ e $\delta = 0.1$. No total, são 1000 amostras uniformemente distribuídas sobre a área em cinza claro. (a) Distribuição de densidade dos dados de entrada. (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.	97
Figura 6.4:	Simulação de uma distribuição de dados em forma de triângulo com $\rho = 0.85$ e $\delta = 0.1$. Ao todo são 1000 amostras uniformemente distribuídas ao longo da área em cinza claro. (a) Distribuição de densidade dos dados de entrada. (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.	98
Figura 6.5:	Falha de ordenação topológica observada em uma rede de Kohonen convencional com 25 neurônios dispostos regularmente em 5 linhas \times 5 colunas. (a) Distribuição dos dados de entrada (1000 amostras). (b) e (c) Valores iniciais e finais, respectivamente, dos pesos sinápticos da rede. O arranjo de neurônios encontra-se em uma configuração metaestável e, neste caso, a proximidade espacial dos neurônios é violada.	99
Figura 6.6:	(a) Número de neurônios em função do parâmetro de plasticidade. (b) Número de neurônios em função do parâmetro de novidade. (c) Erro médio quadrado em função do parâmetro de plasticidade. (d) Erro médio quadrado em função do parâmetro de novidade.	100
Figura 6.7:	Simulação de uma distribuição de dados em forma de duas espirais com $\rho = 0.7$ e $\delta = 0.1$. (a) Distribuição de densidade de duas espirais concêntricas. Ao todo são 960 amostras. (b) Condição da rede após 10 épocas de treinamento.	102
Figura 6.8:	Simulação de uma distribuição multimodal de dados com $\rho = 0.85$ e $\delta = 0.1$. (a) Distribuição de entrada (1000 amostras distribuídas em quatro regiões desconexas). (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.	102
Figura 6.9:	Identificação de agrupamentos com $n_{max} = 5$. (a) Modelo gerador com cinco agrupamentos. O símbolo \times representa o centróide de um determinado agrupamento. (b) Os cinco protótipos calculados pela rede neural se aproximam dos cinco centróides do modelo gerador. (c) Diagrama de Voronoi resultante.	103
Figura 6.10:	Simulação de uma distribuição não-estacionária de dados com $\rho = 0.8$ e $\delta = 0.1$. Ao todo são 1000 amostras uniformemente distribuídas dentro das áreas em cinza claro. (a) Configuração inicial do mapa de características. (b) A distribuição de dados é drasticamente modificada e se inicia um processo de inclusão e migração de neurônios. (c) Todos os neurônios são realocados para regiões de alta densidade probabilística.	103

- Figura 6.11: (a) Erro de quantização da rede (EMQ=0.00559). (b) Número de neurônios em função do tempo. O número mínimo de neurônios permitido na rede é fixado em 2. O número máximo de neurônios é governado pelo parâmetro n_{max} e, neste caso, vale 300. 104
- Figura 6.12: Valor do peso da sinapse de habituação de um neurônio arbitrário em função do tempo. O neurônio é adicionado à rede no instante de tempo 11. A distribuição de dados é alterada após 5 épocas. Neste momento, o neurônio se torna não-representativo para a distribuição de densidade e é removido da rede no instante de tempo 5542. 104
- Figura 6.13: Simulação de uma distribuição não-estacionária de dados com $\rho = 0.8$ e $\delta = 0.1$. Em particular, o parâmetro da taxa de sensibilização, α , foi fixado em 0.0. No total, são 1000 amostras uniformemente distribuídas sobre as áreas em cinza claro. (a–c) Configuração do mapa antes, durante e depois da distribuição sofrer uma alteração. . . 105
- Figura 6.14: Simulação de uma distribuição não-uniforme de dados com $\rho = 0.85$, $\delta = 0.1$. (a) Distribuição dos dados de entrada. Ao todo são 1000 amostras distribuídas sobre as áreas em cinza: 900 dentro do retângulo da esquerda e apenas 100 ao longo do retângulo da direita. (b) Alocação de mais recursos (i.e., neurônios) para a região com maior densidade de amostras. (c) Diagrama de Voronoi resultante. 106
- Figura 6.15: Ilustração da seqüência temporal em forma de “8”. O círculo fechado (‘•’) simboliza o ponto inicial da seqüência ($t = 0$), a seta indica o sentido do fluxo (da esquerda para a direita) e o círculo aberto (‘o’) representa os dois pontos ambíguos da seqüência temporal ($t = 50$ e $t = 150$), alocados na mesma posição espacial (0.5, 0.5). 106
- Figura 6.16: Simulação de uma distribuição de dados em forma de “8” com $\rho = 0.7$, $\delta = 0.1$, $\mu = 0.7$, $\eta = 0.7$, $\alpha = 0.01$, $\beta = 0.9$ e $\gamma = 0.7$. (a) Condição da rede após 10 épocas de aprendizagem. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01001). (d) Atividade na saída da rede na última época. Os símbolos \blacktriangledown e \blacktriangle indicam os instantes $t = 50$ e $t = 150$, respectivamente, representados pelos neurônios de índice 5 e 14, respectivamente. 107
- Figura 6.17: Simulação de uma distribuição de dados em forma de “8” com 20 neurônios dispostos num mapa de Kohonen unidimensional circular de parâmetros $\sigma = 1.5$ e $\eta = 1$. (a) Condição da rede após 10 épocas de treinamento. (b) Heurística usada para decrescer monotonicamente o valor de η com o transcorrer do treinamento. (c) Erro de quantização da rede (EMQ=0.019118). (d) Atividade na saída da rede na última época. 109
- Figura 6.18: Freqüências de convergência e de classificação de cada rede neural auto-organizável na tarefa de segmentação temporal. A barra da esquerda indica a porcentagem de convergência dos neurônios protótipos, ou seja, a freqüência de representação estável da seqüência temporal. A barra da direita aponta a porcentagem de acerto na classificação da seqüência temporal, isto é, a freqüência de identificação da posição de ambigüidade. 110

- Figura 6.19: Erro médio quadrado de quantização de cada rede neural auto-organizável na tarefa de segmentação temporal. A linha vertical sobre cada barra representa a margem de erro, isto é, o desvio padrão para mais e para menos em torno da média. 110
- Figura 6.20: Simulação de uma distribuição de dados em forma de “8” com ruído de distribuição uniforme no intervalo $[-0.1, 0.1]$. (a) Condição da rede após 10 épocas. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01872). (d) Atividade na saída da rede na última época. 111
- Figura 6.21: Simulação de uma distribuição de dados em forma de “8” com ruído de translação. (a) Condição da rede após 10 épocas de aprendizagem. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01891). (d) Atividade na saída da rede na última época. 112
- Figura 6.22: Simulação de uma distribuição de dados em forma de “8” com ruído de rotação. (a) Condição da rede após 10 épocas de aprendizagem. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01803). (d) Atividade na saída da rede na última época. 113
- Figura 6.23: Ilustração da seqüência temporal definida sobre uma circunferência de raio igual a 0.5 e centro igual a (0.5, 0.5). No total, são 50 amostras uniformemente distribuídas ao longo da circunferência. O símbolo \bullet representa o ponto inicial da seqüência e a seta indica o sentido de rotação do fluxo: (a) horário ($t = \{0..249; 500..749\}$) ou (b) anti-horário ($t = \{250..499; 750..999\}$). 114
- Figura 6.24: Simulação de uma distribuição de dados em forma de circunferência com $\rho = 0.7$, $\delta = 0.1$, $\mu = 0.6$, $\eta = 0.4$, $\alpha = 0.05$, $\beta = 0.9$, $\gamma = 0.7$. (a) Condição da rede após 10 épocas de aprendizagem. O símbolo \blacktriangledown indica os neurônios ativos durante o sentido de rotação horário. O símbolo \blacktriangle aponta os neurônios ativos durante o sentido de rotação anti-horário. (b) Espaço de representação gerado pela unidade de memorização Ψ . Todos os 50 pontos são ambíguos. Os símbolos \blacktriangledown e \blacktriangle representam os pontos iniciais da seqüência temporal nos sentidos de rotação horário e anti-horário, respectivamente. 114
- Figura 6.25: Identificação de dois agrupamentos distintos na distribuição de dados. Os neurônios posicionados no eixo $z = 1.0$ mapeiam o sentido de rotação horário. Os neurônios alocados no eixo $z = -1.0$ mapeiam o sentido de rotação anti-horário. 115
- Figura 6.26: (a) Neurônios inseridos entre a primeira e a quinta época (rotação horária). (b) Atividade na saída da rede na quinta época. (c) Neurônios adicionados entre a quinta e a décima época (rotação anti-horária) (d) Atividade na saída da rede na décima época. 116
- Figura 6.27: Simulação de uma distribuição de dados em forma de circunferência com uma rede TKM de parâmetro $\lambda = 0.8$. (a) Condição da grade após 10 épocas de aprendizagem. (b-c) Atividade na saída da rede na quinta época e na décima época, respectivamente. Note que todos os neurônios estão ativos em ambos os sentidos de rotação. 116

LISTA DE TABELAS

Tabela 2.1:	Abordagens cognitivas para o estudo da inteligência	27
Tabela 2.2:	Visão geral dos principais paradigmas de aprendizagem	37
Tabela 3.1:	Inspiração biológica das redes neurais. Adaptado de: (BARONE, 2003, p. 120).	47
Tabela 4.1:	Resumo das redes neurais auto-organizáveis temporais	68
Tabela 4.2:	Resumo das redes neurais auto-organizáveis construtivas	73
Tabela 5.1:	Resumo das funções da rede neural proposta.	87
Tabela 5.2:	Resumo das variáveis da rede neural proposta.	88
Tabela 5.3:	Resumo dos parâmetros da rede neural proposta.	88
Tabela 5.4:	Resumo das principais características da rede neural proposta.	94
Tabela 6.1:	Relação do parâmetro de novidade com a gênese da rede	101
Tabela 6.2:	Resumo dos desempenhos de segmentação da seqüência temporal “8”	108

LISTA DE ALGORITMOS

Algoritmo 3.1:	Algoritmo de Retropropagação de Erros (<i>Backpropagation</i>) . . .	46
Algoritmo 3.2:	Algoritmo do Mapa Auto-Organizável de Kohonen	49
Algoritmo 5.1:	Algoritmo da rede neural proposta.	87

RESUMO

O presente trabalho propõe um novo modelo de rede neural artificial voltado a aplicações robóticas, em especial a tarefas de *natureza espaço-temporal* e de *horizonte infinito*. Este modelo apresenta três características que o tornam único e que foram tomadas como guia para a sua concepção: *auto-organização*, *representação temporal* e *aprendizado construtivo*. O algoritmo de aprendizagem auto-organizada incorpora todos os mecanismos que são básicos para a auto-organização: *competição global*, *cooperação local* e *auto-amplificação seletiva*. A rede neural é suprida com propriedades dinâmicas através de uma *memória de curto prazo*. A memória de curto prazo é inserida na estrutura da rede por meio de *integradores* e *diferenciadores*, os quais são implementados na camada de entrada da rede. Nesta abordagem existe uma evidente separação de papéis: a rede é responsável pela não-linearidade e a memória é responsável pelo tempo. A construção automática da arquitetura da rede neural é realizada de acordo com uma *unidade de habituação*. A unidade de habituação regula o *crescimento* e a *poda* de neurônios. O procedimento de inclusão, adaptação e remoção de conexões sinápticas é realizado conforme o método de *aprendizado hebbiano competitivo*. Em muitos problemas práticos, como os existentes na área da robótica, a auto-organização, a representação temporal e o aprendizado construtivo são fatores imprescindíveis para o sucesso da tarefa. A grande dificuldade e, ao mesmo tempo, a principal contribuição deste trabalho consiste em integrar tais tecnologias em uma arquitetura de rede neural artificial de maneira eficiente. Estudos de caso foram elaborados para validar e, principalmente, determinar as potencialidades e as limitações do modelo neural proposto. Os cenários abrangeram tarefas simples de classificação de padrões e segmentação temporal. Os resultados preliminares obtidos demonstraram a eficiência do modelo neural proposto frente às arquiteturas conexionistas existentes e foram considerados bastante satisfatórios com relação aos parâmetros avaliados. No texto são apresentados, também, alguns aspectos teóricos das ciências cognitivas, os fundamentos de redes neurais artificiais, o detalhamento de uma ferramenta de simulação robótica, conclusões, limitações e possíveis trabalhos futuros.

Palavras-chave: Ciência cognitiva corporificada, redes neurais artificiais, auto-organização, representação temporal, aprendizado construtivo.

A growing self-organizing neural network for lifelong learning of spatiotemporal patterns

ABSTRACT

The present work proposes a new artificial neural network model suitable for robotic applications, in special to *spatiotemporal tasks* and *infinite horizon tasks*. This model has three characteristics which make it unique and are taken as means to guide its conception: *self-organization*, *temporal representation* and *constructive learning*. The algorithm of self-organizing learning incorporates all the mechanisms that are basic to the self-organization: *global competition*, *local cooperation* and *selective self-amplification*. The neural network is supplied with dynamic properties through a *short-term memory*. The short-term memory is added in the network structure by means of *integrators* and *differentiators*, which are implemented in the input layer of the network. In this approach exists an evident separation of roles: the network is responsible for the non-linearity and the memory is responsible for the time. The automatic construction of the neural network architecture is carried out taking into account habituation units. The habituation unit regulates the *growing* and the *pruning* of neurons. The procedure of inclusion, adaptation and removal of synaptic connections is carried out in accordance with *competitive hebbian learning* technique. In many practical problems, as the ones in the robotic area, self-organization, temporal representation and constructive learning are essential factors to the success of the task. The great difficulty and, at the same time, the main contribution of this work consists in the integration of these technologies in a neural network architecture in an efficient way. Some case studies have been elaborated to validate and, mainly, to determine the potentialities and the limitations of the proposed neural model. The experiments comprised simple tasks of pattern classification and temporal segmentation. Preliminary results have shown the good efficiency of the neural model compared to existing connectionist architectures and they have been considered sufficiently satisfactory with regard to the evaluated parameters. This text also presents some theoretical aspects of the cognitive science area, the fundamentals of artificial neural networks, the details of a robotic simulation tool, the conclusions, limitations and possible future works.

Keywords: Embodied cognitive science, artificial neural networks, self-organization, temporal representation, constructive learning.

1 INTRODUÇÃO

1.1 Contexto do trabalho

As ciências cognitivas, ou ciências da cognição, consistem em um amplo programa de pesquisa que tem por objeto a *inteligência*, desde sua estrutura formal até seu substrato biológico. As ciências cognitivas envolvem várias disciplinas, dentre as quais pode-se citar a psicologia, a filosofia, a física, a lingüística e a antropologia. Cada disciplina contribui, conforme suas perspectivas, para compreender, conceber e pensar o conhecimento e o saber humanos (DUPUY, 1996). O estudo das ciências cognitivas possui duas correntes científicas identificáveis que se intercalam em períodos da História. Uma corrente, denominada *da objetividade* (ou *tradicional*), assume uma separação entre sujeito e meio e, portanto, o saber e conhecer humanos são vistos como representações internas realizadas pelo sujeito de um mundo externo. A outra corrente, chamada *da não-objetividade* (ou *contemporânea*), assume o sujeito como parte do meio, e o saber e conhecer humanos surgem das interações do sujeito com o mundo do qual ele faz parte.

De acordo com os fundamentos teórico-conceituais de cada uma das vertentes das ciências da cognição, a Inteligência Artificial¹ (IA) busca projetar e desenvolver mecanismos de *software* e/ou *hardware* que apresentem comportamento inteligente. Em sua vasta maioria, os desenvolvimentos têm se embasado em abordagens tradicionais, alcançando relativo sucesso. Isso se deve em parte a algumas inadequações das abordagens tradicionais quanto à suposta inteligência dos mecanismos artificiais construídos (e.g., divisão do processamento da informação em sentir-pensar-agir). Por outro lado, as abordagens contemporâneas, dentre as quais se destaca a cognição situada e corporificada (COSTA; ROCHA, 2005; DOWNING, 2005), apresentam um caminho alternativo e aparentemente mais coerente para se desenvolver os agentes de *software* cognitivos (SANTOS, 2003).

Um agente de *software* (ou agente de *hardware*)² é composto basicamente por *sensores*, *efetadores* e por uma *entidade cognitiva* que realiza o processamento de informação perceptiva e a tomada de decisão motora (RUSSELL; NORVIG, 2004, p. 6). Em particular, um agente de *software* inteligente é um sistema computacional que opera em ambientes dinâmicos e imprevisíveis e possui atributos que o diferenciam de meros “programas computacionais”, tais como operar sob controle autônomo, perceber seu ambiente, per-

¹A Inteligência Artificial é a parte da Ciência da Computação cuja ênfase está no estudo e desenvolvimento de entidades inteligentes, isto é, sistemas de *software* ou *hardware* que exibem características associadas à inteligência de uma forma geral, como compreensão da linguagem, aprendizado e raciocínio.

²A palavra *agente* vem do latino *agere*, que significa fazer. No âmbito da Inteligência Artificial, o significado do termo agente não é partilhado por todos os investigadores da área. No entanto, a maioria dos pesquisadores define agente como uma entidade – *software* ou *hardware* – imersa em um ambiente e dotada de sensores e efetadores apropriados à execução das tarefas para as quais foi projetada.

sistir por um período de tempo prolongado, adaptar-se a mudanças, ser capaz de assumir metas de outros, etc. (FRANKLIN; GRAESSER, 1997). No presente trabalho, três aspectos são considerados imprescindíveis no projeto de um agente de *software* inteligente: *autonomia*, *localidade* e *corporificação* (PFEIFER; IIDA; BONGARD,). O diagrama da Figura 1.1 mostra graficamente a trilogia³ do agente de *software* inteligente idealizado.

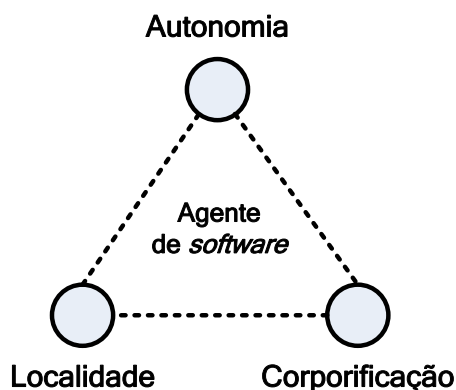


Figura 1.1: A trilogia do agente de *software* inteligente idealizado.

A *autonomia* consiste na capacidade do agente de governar a si próprio durante a execução de determinadas tarefas. O agente de *software* deve ser capaz de funcionar com pouquíssima interferência, supervisão ou instrução externa. O grau de autonomia de um agente está relacionado à sua habilidade de decidir por si só como relacionar os dados dos sensores com os comandos dos efetadores em seus esforços para atingir objetivos, satisfazer motivações, etc (PFEIFER; SCHEIER, 2001). Uma das principais características do comportamento autônomo decorre da necessidade do agente de se adaptar, eventualmente se auto-organizar, de forma a contornar situações imprevistas, como naturalmente ocorre na interação com um meio externo dinâmico, cujos parâmetros são variantes no tempo.

A *localidade* compreende na maneira com que o agente de *software* adquire e utiliza a informação perceptiva. O princípio da localidade afirma que a inteligência de um agente de *software* está situada no meio e não em algum modelo formal do mundo construído em suas estruturas internas. Portanto, um agente de *software* deve adquirir a informação sobre o meio ambiente através de seus próprios sensores e a utilizar no processo de percepção, ao invés de usar deduções baseadas em alguma representação simbólica do mundo.

A *corporificação* constitui na *realização* do agente de *software* como um sistema físico ou computacional (ZIEMKE, 2003). O produto da corporificação é um corpo, o qual confere ao agente sensores e efetadores para mediar a interação com o meio ambiente. De acordo com Florian (2003), a corporificação não é necessariamente dada pela materialidade, como a apresentada em animais ou robôs físicos, mas por uma relação dinâmica com o ambiente. Por esse motivo, a pesquisa também pode ser realizada com ambientes de simulação genuinamente computacionais.

A área da robótica é um campo de estudos fascinante e ao mesmo tempo desafiador. Robôs físicos ou agentes de *software* inteligentes têm cada vez mais o papel de agilizar e facilitar a realização de tarefas cotidianas – por exemplo, transporte de materiais, combate a incêndios, desativação de explosivos, participação em cirurgias e prestação de serviços

³A palavra trilogia significa o estudo de três entidades que se interrelacionam e formam um conjunto. Por exemplo, trilogia pode ser um conjunto de três obras literárias vinculadas entre si e que se unem em um tema comum, ou simplesmente uma obra dividida em três partes.

em geral (RUSSELL; NORVIG, 2004, p. 870–912). Hoje em dia, a pesquisa em redes neurais artificiais⁴ (RNA) tem se apresentado como uma alternativa viável para a solução de problemas complexos no campo da robótica.

As redes neurais artificiais representam um *tema multidisciplinar* com raízes na neurociência, matemática, estatística, física, ciência da computação e engenharia. A sua habilidade para *aprender* a partir de exemplos as tornou ferramentas inestimáveis em aplicações tão diversas como modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinais e controle (ENGEL, 2001). De forma resumida, uma rede neural artificial compreende um sistema não-linear do tipo *caixa-preta*, pois fornece uma estrutura com parâmetros distribuídos que podem ser ajustados através de algoritmos de treinamento eficientes. A classe de modelos que pode ser implementada por uma rede neural depende essencialmente da arquitetura específica da rede considerada (ENGEL, 2001). No presente trabalho, três aspectos fundamentais são ponderados no projeto de uma rede neural: *auto-organização*, *representação temporal* e *aprendizado construtivo*. O diagrama da Figura 1.2 mostra graficamente a trilogia da rede neural idealizada.

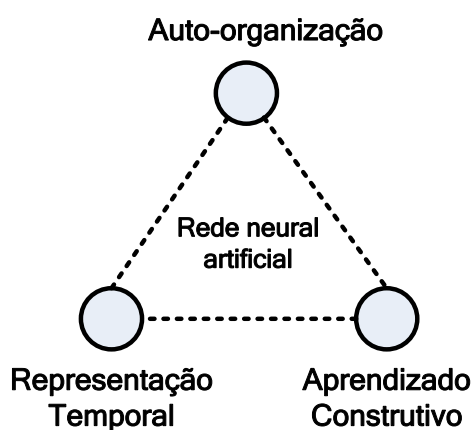


Figura 1.2: A trilogia da rede neural artificial idealizada.

O conceito de *auto-organização* de sistemas tem suas primeiras formulações matemáticas propostas com o advento da Cibernética (ASHBY, 1970; PESSOA JR., 2001). Atualmente, há muitas teorias que apóiam a hipótese de auto-organização de redes neurais. Em comum a todas as teorias, há princípios e mecanismos básicos que regem a auto-organização – por exemplo, a *competição global*, a *cooperação local* e a *auto-amplificação seletiva* (HAYKIN, 2001, p. 430–432). Um sistema é dito auto-organizável se a sua estrutura é modificada em função de sua experiência e de seu relacionamento com o meio ambiente, *sem* o auxílio de um professor externo ou um crítico para supervisionar a aprendizagem.

A *representação temporal* constitui um ingrediente essencial do processo de aprendizagem. É através da incorporação do tempo na operação de uma rede neural que ela se torna capaz de solucionar diversas tarefas cognitivas encontradas na prática, como a visão, a fala e o controle motor (HAYKIN, 2001, p. 686). No entanto, o processamento de seqüências temporais é uma tarefa desafiadora para a qual ainda não há uma abordagem unificada (BARRETO; ARAÚJO, 2001, p. 5). Isto se deve em parte à diversidade de propriedades dos sinais temporais. Basicamente, há duas formas de incluir o tempo na operação

⁴As redes neurais artificiais são também referidas na literatura como *neurocomputadores*, *redes conexionistas*, *processadores paralelamente distribuídos*, etc. (HAYKIN, 2001). Em todo o texto desta dissertação, utiliza-se o termo *redes neurais*; ocasionalmente o termo *redes neurais artificiais* ou *redes* é usado.

de uma rede neural: através de *laços de realimentação (recorrência)* ou por meio de *memórias de curto prazo*. Em geral, a memória de curto prazo é a peça-chave para o projeto de redes neurais, pois converte informação temporal em espacial, possibilitando que redes estáticas convencionais sejam empregadas em tarefas que envolvam o processamento de padrões espaço-temporais. A memória pode ser aplicada *externa* ou *internamente* à rede e comumente é codificada por *linha de atrasos de tempo, diferenciadores e/ou integradores*.

O *aprendizado construtivo* é uma técnica que possibilita incluir e remover neurônios na estrutura da rede neural⁵. Os métodos de aprendizagem construtiva se enquadram em duas categorias principais (HAYKIN, 2001, p. 245): (i) pelo *crescimento* da rede, começando com uma arquitetura de rede bastante simples e então adicionando um novo neurônio ou uma nova conexão sináptica quando a rede não for capaz de satisfazer as especificações de projeto; e (ii) pela *poda* da rede, iniciando com uma arquitetura de rede de dimensão elevada e então podando-a pela redução ou eliminação de certos neurônios ou pesos sinápticos de uma forma seletiva e ordenada.

Dentro destas concepções, a hipótese de trabalho é a de que as três características supracitadas – auto-organização, representação temporal e aprendizado construtivo –, quando combinadas adequadamente em uma rede neural artificial, viabilizam a resolução de *tarefas de horizonte infinito*⁶ e de *dimensão espaço-temporal*. Esta pesquisa, em particular, se posiciona nesse contexto e se propõe a definir um modelo de rede neural auto-organizável, temporal e construtivo voltado a aplicações de robótica.

1.2 Motivação principal

O presente trabalho tem como foco principal aplicações de robótica. Os problemas encontrados na área da robótica são, via de regra, bastante complexos, pois envolvem grande dimensionalidade de dados e demandam processamento multisensorial em tempo real. Os dados são espaço-temporais e apresentam redundância e perturbações por ruído. Além disso, a interação com o meio ambiente pode levar a situações nunca vistas e, portanto, é necessário constante aprendizado. Logo, para um sistema computacional lidar com problemas dessa natureza, é necessário criar uma *representação temporal* ou causal do ambiente e possuir um mecanismo de *aprendizado não-supervisionado, construtivo e em tempo real* (do inglês *on-the-fly learning*).

Os algoritmos de aprendizado tradicionais para processamento temporal, tais como o BPTT (do inglês *Backpropagation Through Time*) (WERBOS,) e o RTRL (do inglês *Real-Time Recurrent Learning*) (WILLIAMS; ZIPSER, 1989), são computacionalmente complexos e têm poder limitado para a memorização de eventos, o que limita consideravelmente a gama de problemas temporais que podem ser solucionados (HOCHREITER; SCHMIDHUBER, 1997). No entanto, as limitações impostas por tais redes recorrentes podem ser contornadas se os dados de entrada forem acompanhados de informação temporal útil, geralmente, na forma de *memórias de curto prazo*. Entende-se por útil uma representação que permita a generalização de conceitos, possibilite a rápida adaptação a mudanças na dinâmica do sistema e seja econômica sob o ponto de vista computacional. Em geral, com o emprego de uma memória de curto prazo, a tarefa de encontrar representações internas se torna mais simples e os problemas dinâmicos tratáveis podem ser mais complexos, como os existentes na robótica.

Os algoritmos de aprendizado com topologia variável, entre eles o CasCor (do in-

⁵As redes neurais com estrutura dinâmica são também chamadas de *redes ontogênicas* (FIESLER, 1994).

⁶Tarefas de horizonte infinito são também referenciadas como tarefas contínuas (SUTTON; BARTO, 1998).

glês *Cascade-Correlation*) (FAHLMAN; LEBIERE, 1990), as redes da família ART (do inglês *Adaptive Resonance Theory*) (CARPENTER; GROSSBERG, 2003) e o GNG (do inglês *Growing Neural Gas*) (FRITZKE, 1995b), modificam a estrutura da rede neural durante o processo de treinamento até que uma solução de projeto adequada seja apresentada. Contudo, a maioria das redes construtivas não é apropriada para lidar com sinais variantes no tempo (i.e., sinais não-estacionários) e nem com aplicações que necessitam de aprendizado constante e em tempo real (e.g., tarefas de horizonte infinito). As principais causas para essas duas dificuldades são a ausência de uma memória de curto prazo e o modo de inserção de novos neurônios que, em geral, é baseado em interações múltiplas de uma constante de tempo predefinida pelo projetista da rede.

De acordo com o exposto, até onde se sabe, não há na literatura especializada da área algoritmos de aprendizado de natureza auto-organizável que integrem a representação temporal e o aprendizado construtivo. Dessa forma, existe uma carência muito grande por uma fundamentação teórica, uma arquitetura, uma metodologia de criação e ferramentas que estejam definidas e difundidas no meio acadêmico. Isso implica um amplo campo de pesquisas a ser explorado e é neste hiato identificado que este trabalho de pesquisa retira a sua motivação principal e pretende se desenvolver.

1.3 Objetivo geral e objetivos específicos

O objetivo geral do presente trabalho é projetar uma rede neural auto-organizável, temporal e construtiva que possa ser empregada na resolução de determinados problemas práticos de agentes de *software* inteligentes – por exemplo, tarefas robóticas de *natureza espaço-temporal* e de *horizonte infinito*. A rede é denominada GTSOM (do acrônimo em inglês para *Growing Temporal Self-Organizing Map*). A partir dessa meta inicial, a rede neural a ser desenvolvida deve ser planejada para atender os seguintes requisitos:

- seja de natureza não-supervisionada para ser utilizada em tarefas de tempo real;
- adote uma representação implícita do tempo para processar informação temporal;
- possua uma estrutura dinâmica para variar no decorrer do tempo o número de neurônios e a quantidade de conexões sinápticas;
- tenha somente parâmetros constantes no tempo para funcionar em uma única fase de operação (i.e., sem distinção entre treinamento e atuação);
- seja parametrizável para se adaptar às necessidades específicas de cada aplicação;
- seja uma solução econômica do ponto de vista computacional; e
- permita ser incorporada à arquitetura de controle (i.e., entidade cognitiva) de um agente de *software* inteligente.

1.4 Posicionamento, justificativa e relevância

Em muitos problemas práticos, como os existentes na robótica, a auto-organização, a representação temporal e o aprendizado construtivo são fatores importantes e, até mesmo, imprescindíveis para o sucesso da aplicação. A grande dificuldade reside em integrar tais tecnologias em uma arquitetura de rede neural de maneira eficiente, tal que a resolução de determinadas tarefas contínuas de dimensão espaço-temporal seja possível e viável.

Um sistema auto-organizável é aquele que tende a melhorar espontaneamente seu desempenho ao longo do tempo enquanto persegue os seus objetivos. Esta formulação inclui o caso especial no qual o objetivo é alcançar um alto grau de organização (*ordem*) de entidades relevantes a partir de um baixo nível de organização (*caos*). Em particular, em uma rede neural artificial, muitas interações locais originalmente aleatórias entre neurônios vizinhos de uma rede podem se fundir em estados de ordem global e, em última instância, levar a um comportamento coerente na forma de padrões espaciais ou ritmos temporais (HAYKIN, 2001, p. 430). Dessa forma, a estabilidade de uma rede neural está fortemente relacionada com o seu poder de auto-organização.

A representação temporal deve ser uma propriedade das mais relevantes de um sistema inteligente, seja ele biológico ou artificial, haja vista que muitas das tarefas desempenhadas por seres humanos e animais abrangem tomadas de decisão e respostas comportamentais a estímulos espaço-temporais – por exemplo, a visão, a fala e o controle motor. Atualmente, já existem teorias bem-desenvolvidas para resolver problemas dinâmicos quando o sistema sob estudo ou o mecanismo físico subjacente de interesse é linear. No entanto, quando o sistema ou o mecanismo físico for não-linear, a tarefa se torna mais difícil. É nestas situações que as redes neurais artificiais têm o potencial de fornecer uma solução viável e econômica sob o ponto de vista computacional (HAYKIN, 2001, p. 710).

Em uma rede neural artificial, a representação do conhecimento é definida pelos valores assumidos pelos seus parâmetros livres (i.e., pesos sinápticos e bias) (HAYKIN, 2001, p. 51). Portanto, o conhecimento adquirido pela rede é limitado pelo número existente de conexões sinápticas. Não obstante, a maioria das redes neurais tem uma arquitetura específica e estática que, além de limitar o conhecimento que pode ser adquirido, dificulta a resolução de determinadas tarefas. Isso se deve ao fato de que, redes com dimensão muito reduzida podem não apresentar *flexibilidade* suficiente para tratar alguns problemas, enquanto redes sobre-dimensionadas geralmente apresentam pouca capacidade de *generalização*. A utilização de procedimentos de *validação cruzada* (HAYKIN, 2001, p. 239-244) ou *técnicas de regularização* (GIROSI; JONES; POGGIO, 1995) diminuem os problemas em relação ao sobre-treinamento da rede e conseqüentemente amenizam os efeitos das redes sobredimensionadas, mas estas ainda continuam a exigir um esforço computacional excessivo e desnecessário. Logo, algoritmos capazes de determinar automaticamente a arquitetura da rede neural são extremamente desejáveis.

A *construção automática* da arquitetura de uma rede neural retira o ônus do projetista em ter que especificar *a priori* a conectividade e a topologia da rede, torna possível resolver tarefas que exijam aprendizado em tempo real e oferece uma boa solução para o *dilema da estabilidade-plasticidade* (HAYKIN, 2001, p. 30). Uma forma muito estudada de determinação automática da dimensão de uma rede neural são os chamados *métodos de poda*, cuja idéia é iniciar com uma arquitetura de dimensão elevada e ir retirando unidades ou conexões até que se chegue a uma dimensão adequada. No entanto, estes métodos sempre trabalham com um esforço computacional adicional, exceto no final do processo de poda, e além disso, se existir mais de uma arquitetura adequada para a solução, a rede sempre vai convergir para a de maior dimensão. Alternativamente, *métodos construtivos* são uma ótima opção por começarem com uma arquitetura de rede de dimensão reduzida e irem adicionando unidades intermediárias e/ou conexões até que uma solução adequada seja apresentada. Hoje em dia, a pesquisa científica se concentra no desenvolvimento de técnicas híbridas de crescimento e poda da rede neural.

O presente trabalho se torna relevante ao disseminar um referencial teórico-prático no meio acadêmico, contribuindo para a formação de pesquisadores nesse contexto trans-

disciplinar. Além disso, e mais importante, este trabalho de cunho exploratório e eminentemente prático contribui ao propor um modelo neural baseado no trinômio auto-organização, representação temporal e aprendizado construtivo.

1.5 Contribuições relevantes

Em particular, este trabalho traz os seguintes avanços e contribuições técnicas para a área da Inteligência Computacional⁷ (IC):

1. Revisão dos fundamentos teórico-conceituais da área das ciências cognitivas e síntese dos princípios de projeto para agentes de *software* inteligentes;
2. Revisão do estado da arte na utilização de redes neurais artificiais não-supervisionadas no processamento de sinais temporais;
3. Revisão do estado da arte em técnicas de aprendizagem construtivas para redes neurais artificiais não-supervisionadas;
4. Implementação de uma ferramenta de simulação robótica para viabilizar a modelagem, o desenvolvimento e a simulação de agentes de *software* inteligentes que estejam em conformidade com o trinômio autonomia, localidade e corporificação;
5. Proposição de um modelo de rede neural auto-organizável, temporal e construtivo, desenvolvido originariamente para tarefas robóticas de natureza espaço-temporal e de horizonte infinito;
6. Aplicação do modelo neural sugerido em alguns problemas computacionais, tais como classificação adaptativa de padrões e segmentação de seqüências temporais.

É importante ressaltar que apesar do modelo neural proposto não se enquadrar efetivamente na visão contemporânea das ciências cognitivas, o modelo permite vislumbrar a realização de expansões qualitativamente interessantes no tocante à integração sensorio-motora, assim como será melhor discutido no capítulo de conclusões e trabalhos futuros.

1.6 Estrutura da dissertação

Este capítulo teve o objetivo de introduzir o leitor no contexto dos temas abordados nesta dissertação através da exposição das motivações, objetivos, justificativa e contribuições relevantes. O restante deste trabalho encontra-se organizado da seguinte forma:

- **Capítulo 2: Fundamentos de Ciências Cognitivas.**

Este capítulo descreve os fundamentos teórico-conceituais da área das ciências cognitivas, bem como introduz os princípios de projeto de agentes de *software*. A discussão é conduzida em duas direções distintas, conforme os princípios, modelos e métodos de investigação de cada uma das vertentes das ciências da cognição.

⁷A Inteligência Computacional (IC) reúne uma série de abordagens e técnicas que tenta modelar a *inteligência* através da representação numérica, contínua do *conhecimento* (ENGEL, 2005).

- **Capítulo 3: Fundamentos de Redes Neurais Artificiais.**

Este capítulo apresenta uma breve revisão dos principais fundamentos de redes neurais artificiais, definindo uma terminologia coerente para o restante do volume. Nesse sentido, são descritos os componentes básicos de um neurônio artificial e de uma rede neural de múltiplos neurônios, com destaque para a rede neural auto-organizável baseada em grades sugerida por Teuvo Kohonen.

- **Capítulo 4: Trabalhos Correlatos.**

Este capítulo apresenta uma visão abrangente das pesquisas relacionadas ao tema desta dissertação. Inicialmente, trabalhos que tratam da questão temporal em redes neurais auto-organizáveis são discutidos. Em seguida, trabalhos que lidam com a dimensão construtiva em redes neurais auto-organizáveis são descritos.

- **Capítulo 5: Proposta de Rede Neural Artificial.**

Este capítulo detalha a proposta de rede neural auto-organizável, temporal e construtiva. Inicialmente, dá-se uma visão geral da estrutura da rede que está sendo sugerida. Em seguida, discorre-se sobre o seu funcionamento, com especial ênfase nos processos de inclusão, remoção e adaptação de neurônios e conexões sinápticas. Por fim, apresentam-se as principais propriedades da rede, realiza-se uma análise sobre seus parâmetros e argumenta-se sobre sua plausibilidade biológica.

- **Capítulo 6: Resultados Experimentais.**

Este capítulo relata os resultados de diferentes experimentos preliminares de avaliação aplicados à rede neural proposta, considerando tarefas simples de classificação de padrões e segmentação de seqüências temporais. Para fins de validação da rede neural proposta, também são apresentadas comparações qualitativas e quantitativas com outros trabalhos similares.

- **Capítulo 7: Conclusões e Trabalhos Futuros.**

Este capítulo relata as conclusões gerais e considerações finais do trabalho, apresenta as limitações verificadas do modelo neural proposto e discute possíveis desdobramentos e direções para o encaminhamento de pesquisas futuras.

- **Apêndice A: Ferramenta de Simulação Robótica.**

Este apêndice apresenta a ferramenta de simulação robótica desenvolvida durante a elaboração desta dissertação. A ferramenta de simulação robótica proporciona uma plataforma simples, flexível e de fácil uso para modelagem, prototipação e simulação de agentes de *software* situados e corporificados.

2 FUNDAMENTOS DE CIÊNCIAS COGNITIVAS

Este capítulo tem como objetivo contextualizar a área das ciências cognitivas, apresentando os fundamentos teórico-conceituais e os princípios de projeto de agentes de *software* de cada uma das vertentes das ciências cognitivas. Não se pretende exaurir este tema, mas apenas incitar uma reflexão e apresentar algumas idéias que serão úteis ao desenvolvimento deste trabalho. Alguns livros-texto interessantes que abordam o assunto de forma teórica são (DUPUY, 1996) e (SANTOS, 2003). De maneira mais prática, Pfeifer e Scheier (2001) guiam o desenvolvimento de sistemas inteligentes.

2.1 Introdução

Nos primórdios da vida na Terra, toda a informação usada para resolver problemas dos seres primitivos estava armazenada no DNA (acrônimo de ácido desoxirribonucléico). Com o aparecimento do *sistema nervoso*, e do *cérebro* em especial, a memória dos eventos ocorridos durante a existência de um organismo passou a ser utilizada como *base de conhecimento* para solucionar novos problemas. Nesse momento, houve um tremendo salto evolutivo, pois o comportamento de um organismo poderia ser modificado durante a sua vida sem alteração de seu *código genético*. Dessa forma, os seres vivos que possuíam cérebro tinham uma ferramenta que incrementava e adaptava o seu conhecimento e seu comportamento de uma forma muito mais dinâmica (FLORES; SCHÜLER,).

Desde então, um conjunto de esforços interdisciplinares para melhor compreender o funcionamento do sistema nervoso vem sendo realizado por especialistas de diferentes áreas das *ciências da cognição* (CONNORS; BEAR; PARADISO, 2002). Por exemplo, os psicólogos encaram o sistema nervoso como um objeto desconhecido mas capaz de produzir comportamento e consciência, e assim dedicam-se ao estudo de suas propriedades. Já os neurobiólogos celulares o vêem como um conjunto de células que se tocam através de finos prolongamentos, formando trilhões de complexos circuitos intercomunicantes. Por outro lado, eletrofisiologistas estudam apenas os sinais elétricos produzidos pelas células nervosas. Alternativamente, neuroquímicos se concentram nas reações químicas que ocorrem entre as moléculas existentes dentro e fora das células nervosas. Na visão dos computólogos, o sistema nervoso é concebido como uma fonte inesgotável de inspiração para a solução dos mais variados problemas, e assim dão ênfase ao estudo dos princípios e mecanismos que o regem para aplicá-los à construção de sistemas computacionais.

Logo, as ciências cognitivas constituem em um amplo programa de pesquisa que tem por objeto a *inteligência*, desde sua estrutura formal até seu substrato biológico. A disciplina tem sua origem, possivelmente, no movimento cibernético ocorrido por volta de 1950, em uma tentativa de edificar uma ciência geral sobre o funcionamento da mente. As ciências cognitivas são caracterizadas por uma variedade de abordagens teóricas, mode-

los e métodos de investigação. Na literatura, comumente agrupa-se este arsenal teórico-conceitual em *abordagens tradicionais* e *abordagens contemporâneas*, de acordo com seus princípios *objetivistas* e *não objetivistas*, respectivamente. A Tabela 2.1 apresenta de forma esquemática um paralelo entre estas duas abordagens.

Tabela 2.1: Abordagens cognitivas para o estudo da inteligência

Classe	Paradigma	Descrição
Tradicional (objetividade)	Objetivismo	Baseado na estrutura funcional do modelo formal da “máquina de Turing”
	Conexionismo	Baseado na estrutura funcional do modelo neural de McCulloch-Pitts
Contemporânea (não-objetividade)	Teoria Ecológica	Co-dependência entre a percepção e a ação. Atribui um papel ativo para a ação
	Teoria da Autopoiese	Determina dois domínios fenomênicos da existência de seres vivos (estrutura e interações)
	Teoria da Enação	Define a cognição como uma ação incorporada, dando ênfase à inseparabilidade dos processos sensoriais e motores
	Teoria Psicológica Construtivista	Estabelece que a cognição seja paulatinamente construída na interação do sujeito com o meio, conforme um equilíbrio entre processos de assimilações e acomodações
	Teoria da Seleção de Grupos Neurais	Explica a cognição através de um modelo selecionista sobre a organização, o desenvolvimento e o funcionamento do cérebro

2.2 Abordagens tradicionais das ciências cognitivas

As abordagens baseadas nos princípios da objetividade, referidas doravante como *tradicionais*, representam o paradigma dominante das ciências cognitivas e se enquadram basicamente em dois programas de pesquisa: o *cognitivismo* e o *conexionismo*.

Em síntese, tanto a abordagem cognitivista quanto a conexionista se apóiam na visão de um mundo existente, com características determinadas e *independentes do observador*. O observador tem um papel completamente passivo no processo de perceber o mundo ao seu redor e, portanto, o saber e conhecer humanos são vistos como *representações* ou modelos internos da realidade externa. Como consequência, ambos os paradigmas caminham em direção ao representacionismo do mundo externo e ao processamento de estímulos do ambiente, seja na forma simbólica quanto sub-simbólica.

2.2.1 Cognitivismo

O *cognitivismo* tem sua origem na estrutura funcional do consagrado modelo da máquina de Turing (1936). Por isto, o conhecimento é representado por um conjunto de símbolos e regras que têm uma correspondência direta e explícita com os objetos do mundo real. A cognição ou aprendizagem é a manipulação dos símbolos conforme suas propriedades, relacionamentos e regras aplicáveis. O comportamento inteligente é obtido através de processos que manipulam satisfatoriamente os símbolos e as regras.

2.2.2 Conexionismo

O *conexionismo* está apoiado sobre o modelo neural de McCulloch-Pitts (1943). Neste caso, o conhecimento é representado de forma sub-simbólica por um conjunto de pesos sinápticos entre unidades funcionais concebidas em analogia aos neurônios naturais. A cognição ou aprendizagem consiste na modificação coerente destes pesos sinápticos. O comportamento inteligente é a resposta satisfatória a um estímulo externo, obtido através de um comportamento emergente da rede neural artificial.

2.2.3 Projeto de agentes de *software* tradicionais

No âmbito da Inteligência Artificial, um agente de *software* cognitivista ou conexio-nista é visto como um sistema seqüencial composto por *entrada*, *processamento* e *saída* (vide Figura 2.1). Inicialmente, dados oriundos do ambiente são obtidos através de sensores e, em um processo compreendido como a *percepção*, mapeados para representações internas. Em um segundo momento, há um processamento central efetuado sobre esta representação, que pode ser realizado de inúmeras formas e compreender inúmeras etapas. Por fim, o sistema gera uma resposta ao ambiente por intermédio de efetadores.

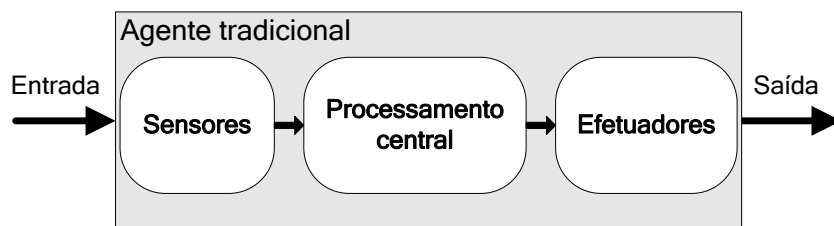


Figura 2.1: A visão tradicional do processamento de informação

A divisão do processamento nestas três fases implica a compreensão da percepção como um processo separado e anterior à ação, na suposição da cognição como central e distinta dos processos sensorio-motores, e na ação como um processo passivo e dependente dos demais. Deste modo, há um fluxo de "informação" unidirecional indo da percepção, passando pela cognição e, finalmente, pela ação (PFEIFER; SCHEIER, 1994).

Em virtude do representacionismo e da separação em sentir-pensar-agir das abordagens clássicas das ciências da cognição, inúmeros problemas fundamentais são inseridos na modelagem de agentes de *software*, entre eles:

- como estabelecer as representações internas a partir de dados sensoriais e mantê-las em sintonia com um mundo externo que está continuamente em mudança. Esse problema foi originalmente descrito por McCarthy e Hayes em 1969 e é referenciado como o *problema do enquadramento* (do inglês *the frame problem*);
- como relacionar os símbolos com os objetos do mundo real, já que grande parte do conhecimento necessário é tácito e, portanto, não explicitável e nem tampouco redutível a seqüência de fatos e regras. Esse problema foi primeiramente discutido por Steven Harnad em 1990 e é conhecido como o *problema da fundamentação do símbolo* (do inglês *the symbol-grounding problem*);
- como interagir com um mundo externo em tempo real, uma vez que isto é dificilmente alcançado postulando a existência de um módulo centralizador; e

- como categorizar os objetos presentes no meio, haja vista que um objeto pode conduzir a inúmeros padrões de entrada diferentes dependendo do ângulo de visão relativo ao objeto, de condições de luminosidade do ambiente, de ruído associado a sensores e efetadores, e assim por diante. Esse problema já foi estudado por diversos pesquisadores (e.g., Rodney Brooks) e é denotado por problema da localidade e da corporificação (do inglês *the embodiment and situatedness problem*).

A inquietude demonstrada por parte da comunidade científica com relação às questões levantadas propiciou o surgimento de abordagens alternativas para a compreensão da cognição e, conseqüentemente, da percepção e da ação (PFEIFER; SCHEIER, 1994; MÖLLER, 1999; HURLEY, 2001). A próxima seção discorre sobre cada uma destas abordagens.

2.3 Abordagens contemporâneas das ciências cognitivas

As abordagens baseadas nos princípios da não-objetividade, doravante referidas como *contemporâneas*, não representam o mundo em que habitam mas o criam dinamicamente a partir da interação com ele. Por isto, também são caracterizadas como abordagens holísticas, ecológicas, situacionistas ou não-representacionistas. Dentre as abordagens contemporâneas proeminentes, as principais são apresentadas nas subseções seguintes. Apesar das particularidades inerentes a cada abordagem contemporânea, todas compartilham bases ontológicas¹ e princípios epistemológicos² semelhantes. A cognição é concebida em uma circularidade de processos sensoriais e motores, e a percepção e a ação constituem processos simultâneos e inseparáveis.

2.3.1 Teoria Ecológica

A *Teoria Ecológica*, elaborada por James Gibson em 1979, estabelece um marco dentro da história evolutiva das pesquisas científicas a respeito da cognição. Em primeiro lugar, porque pioneiramente enfatiza a *co-dependência* entre a percepção e a ação. Em segundo lugar, porque atribui um *papel ativo* para a ação. Na concepção de Gibson, a percepção não é vista como um estado mental isolado do mundo físico, pelo contrário, o significado da percepção encontra-se na interação contínua entre sujeito e ambiente.

De acordo com Gibson, muitas questões sobre como a informação visual é “construída” internamente pelo sujeito poderiam ser substituídas por questões que tratam das fontes de informação no ambiente, determinantes para o comportamento do sujeito. O conceito de informação reconcilia dois aspectos da relação sujeito-ambiente: os inseparáveis conceitos de *invariantes* e *affordances*. A noção de invariantes é baseada na idéia de que padrões de energia que estimulam os sentidos contêm informações que especificam o meio ambiente. O conceito de *affordances* faz referência ao sujeito, expressando as possibilidades de ação oferecidas pelo meio ambiente. Para maiores informações sobre a Teoria Ecológica de Gibson, consulte (OLIVEIRA, 2005).

2.3.2 Teoria da Autopoiese

A *Teoria da Autopoiese* de Maturana Romecin e Varela (1997, 2001), referenciada mais recentemente por *Biologia do Conhecer*, define o sujeito sob o ponto de vista biológico, destacando o *domínio de sua dinâmica estrutural interna* e o *domínio de suas*

¹ Ontologia é o ramo da filosofia referente à natureza do ser.

² Epistemologia é o ramo da filosofia que procura explicar o fenômeno do conhecer.

relações e interações como totalidade. O sujeito é visto como um sistema dinâmico determinado estruturalmente. Na interação contínua com o ambiente, são modificadas as suas estruturas internas a fim de se manter em congruência circunstancial com o meio. É justamente a conservação dessa congruência que sustenta a existência do ser.

Maturana Romecin e Mpodozis (1997) analisaram um experimento descrito por Roger Sperry a respeito da rotação dos olhos de anfíbios. Nesse experimento foi demonstrado que os animais “recuperam” a visão mas, no entanto, apresentam um variação na conduta de captura de presas com um desvio de ângulo igual ao do giro efetuado nos olhos. Com base nisso, os autores argumentaram a respeito do fenômeno da percepção, pois ao modificar a estrutura do organismo altera-se, por conseguinte, a sua percepção.

2.3.3 Teoria da Enação

A *Teoria da Enação*, desenvolvida por Varela, Thompson e Rosch (2003), entende a cognição como uma *ação incorporada*. O termo incorporada pretende enfatizar dois aspectos principais: a dependência intrínseca da cognição e dos tipos de experiências possibilitadas pelas inúmeras capacidades sensório-motoras que surgem devido ao fato de se ter um corpo; e o contexto biológico, psicológico e cultural nos quais tais capacidades sensório-motoras encontram-se embutidas. Já o emprego do termo ação dá ênfase à inseparabilidade dos processos sensoriais e motores – a percepção e a ação – na cognição.

Em síntese, Varela, Thompson e Rosch (2003) resumem a sua teoria nos seguintes termos: (1) a percepção consiste em ação perceptivamente orientada e (2) as estruturas cognitivas emergem de padrões sensório-motores recorrentes que, por sua vez, possibilitam a ação ser perceptivamente orientada.

2.3.4 Teoria Psicológica Construtivista

A *Teoria Psicológica Construtivista*, proposta por Piaget (1979), define o conhecimento como fruto de ações mútuas indissociáveis entre sujeito e meio. Nesta concepção, não há ênfase nas experiências adquiridas (empirismo) e nem na bagagem hereditária (inatismo), mas sim na relação e interação entre esses dois elementos.

O desenvolvimento do conhecimento construtivista de Piaget tem base nos mecanismos conhecidos como *assimilação* e *acomodação*. A assimilação constitui em uma tentativa de integrar as novas experiências do sujeito em suas estruturas cognitivas previamente construídas. Já a acomodação consiste na modificação progressiva de suas estruturas cognitivas, uma vez que a organização do sujeito não é capaz de lidar com uma determinada situação que o meio impõe. Portanto, o sujeito está a todo instante realizando “assimilações” e “acomodações”, na busca por um equilíbrio cognitivo.

Esta concepção construtivista da inteligência foi reconhecidamente trazida ao campo científico da Inteligência Artificial no trabalho de Drescher (DRESCHER, 1991). A IA Construtivista, como é denominada na literatura, constitui em um ramo alternativo dentro da IA Simbólica, e, em linhas gerais, engloba todo os trabalhos desta ciência que fazem referência à Teoria Psicológica Construtivista. Recentemente, vários esforços foram realizados na concepção de agentes de *software* segundo este paradigma, como, por exemplo, em Wazlawick (1993), Muñoz (1999) e Perotto (2004).

2.3.5 Teoria da Seleção de Grupos Neurais

Nas últimas décadas, experiências científicas em seres humanos e animais identificaram certas áreas do sistema nervoso que são especializadas em determinadas funções

mentais (LENT, 2002, p .24). Mais especificamente, as pesquisas realizadas por Gerald Edelman demonstraram que a cognição é resultado da atividade coordenada de *grupos neurais* dispostos em diferentes regiões cerebrais (EDELMAN, 1987, 1989, 1995, 2004). Foram descobertos grupos neurais sensíveis a cores, ao movimento de estímulos luminosos, tons sonoros, cheiros, ritmos respiratórios e uma lista inumerável de funções.

Edelman engajou-se na pesquisa da estrutura e da função cerebral, propondo em 1978 uma teoria científica do cérebro (EDELMAN, 1978). Esta teoria, denominada *Teoria da Seleção de Grupos Neurais* (ou, abreviadamente, *Darwinismo Neural*), é um *modelo seletcionista* sobre a organização, o desenvolvimento e o funcionamento do cérebro. A idéia é que o cérebro classifica, categoriza e adapta-se ao mundo através de um processo de seleção entre *grupos neurais*. Este processo é análogo àquele que opera na seleção natural de seres vivos, mas seu substrato e mecanismo são evidentemente diferentes.

Um *grupo neural* compreende uma coleção de neurônios intimamente ligados, da ordem de centenas a milhares, os quais disparam concomitantemente a determinados estímulos. De modo geral, as conexões sinápticas entre os neurônios de um grupo neural são constituídas “filogeneticamente”, ou seja, são herdadas geneticamente (transgeracionais). Há, contudo, um processo de seleção darwiniana que dá o ajuste fino (microscópico) por meio da eliminação de neurônios e sinapses. Isto assegura uma variabilidade ontogenética necessária para lidar com a imprevisibilidade do ambiente.

O sistema nervoso possui bilhões de grupos neurais, cada qual com sua pequeníssima especialidade. Por exemplo, um grupo neural para realizar movimentos com o braço para a esquerda, outro para a direita, um para reconhecer objetos da cor vermelha, outro para reconhecer formas triangulares, e assim por diante. Reeke e Edelman (1984, p .186) sugerem que exista um equilíbrio adequado entre a especificidade e generalidade na função de um grupo neural, de modo que variados grupos neurais possam responder razoavelmente bem a qualquer estímulo. Esse fenômeno, denominado *degeneração*, assegura que todo e qualquer estímulo é necessariamente reconhecido por pelo menos um grupo neural.

Existe um número maciço de circuitos fechados entre os neurônios de cada grupo neural. Por essa razão, o estado de um neurônio é dependente do estado de todos os demais neurônios com os quais está conectado, direta ou indiretamente. Devido a essa dependência circular entre os neurônios, o comportamento de um grupo neural é altamente não-linear. Através das experiências vividas por cada ser vivo, os grupos neurais são seletivamente fortalecidos ou enfraquecidos, de acordo com a quantidade de estímulos recebidos ou, mais simplesmente, conforme o seu uso. Este processo é modulado pela atribuição, de diferentes *valores*³, aos estímulos recebidos. À medida que o cérebro se desenvolve, ele abandona certos circuitos inúteis em proveito de conexões repetidas com sucesso, curtidas e recurtidas por uma aprendizagem bem sucedida e recompensada (EDELMAN, 1995).

Os grupos neurais competem uns com os outros, a fim de serem selecionados e se tornarem parte integrante de um *circuito funcional*. Através da *amplificação seletiva*, uma variedade de circuitos funcionais são “epigeneticamente” esculpidos, isto é, através da combinação de influências genéticas e ambientais, o que acaba por originar *mapas neurais locais*. Um mapa neural pode ser definido como um conjunto interconectado de grupos neurais com funcionalidades semelhantes. O sistema nervoso possui inúmeros mapas neurais, cada qual com uma determinada especialidade e localização. Por exemplo, um mapa neural da visão específico para reconhecer cores⁴, que é composto por três

³Para maiores detalhes sobre os sistemas de valores presentes nos seres vivos, consulte a Seção 2.3.6.5.

⁴O termo “reconhecer cores” conota que a ativação de um determinado grupo neural proporciona, no observador, uma sensação que ele descreve, por exemplo, como azul. Dessa maneira, a cor não é uma caracte-

grupos neurais, cujas funcionalidades são reconhecer a cor vermelha, a verde e a azul, respectivamente. Cabe salientar que a maioria das conexões sinápticas entre os grupos neurais é constituída “ontogeneticamente”, ou seja, mediante aprendizagem.

Existe também certa circularidade das conexões entre os grupos neurais. Assim, a frequência de oscilação de cada grupo neural depende das conexões recebidas dos outros grupos do mesmo mapa. Essa propriedade implica uma não-linearidade no comportamento dos mapas neurais, da mesma natureza que a não-linearidade presente no comportamento dos grupos neurais, todavia, em certo sentido, em um nível acima.

Os diversos mapas neurais que vão surgindo também podem estar conectados entre si por intermédio de ligações maciçamente paralelas e recíprocas. Este processo de estabelecer conexões sinápticas entre mapas funcionalmente diferentes origina os *mapas globais*. Por exemplo, um mapa global composto por dois mapas neurais, cujas funções são reconhecer as cores (vermelha, verde, azul, etc.) e as formas (triangular, retangular, circular, etc.) de objetos. Dessa forma, um ser vivo é capaz de “perceber”, por exemplo, um objeto azul de forma triangular. Por meio das conexões reentrantes, os mapas coordenam entre si o aparecimento de importantes funções comportamentais, como *memórias especializadas e capacidades conceituais*.

Devido à reentrância das conexões entre os mapas neurais, há uma co-dependência entre seus estados, isto é, um mapa depende do estado do mapa com o qual está conectado e vice-versa. Vê-se aqui, novamente, a presença da não-linearidade no comportamento do mapa global, porém em um nível mais alto que no mapa local. Resumindo, os mapas globais são constituídos pelas conexões sinápticas reentrantes entre neurônios pertencentes a múltiplos mapas locais e “representam” uma experiência como um todo ao correlacionar categorizações específicas.

Como foi descrito anteriormente, os grupos neurais correlacionam os neurônios que os compõem, os mapas neurais correlacionam os grupos neurais que os compõem, e finalmente, os mapas globais correlacionam os mapas neurais que os compõem. Isto ocorre pois no sistema nervoso existem “apenas” neurônios que se correlacionam entre si, formando circuitos entre diferentes regiões cerebrais. Em outras palavras, os grupos neurais, mapas locais e mapas globais são apenas *abstrações* feitas na linguagem por um observador de regiões funcionais do sistema nervoso. Pode-se notar que uma mudança de estado de um neurônio sensitivo, por exemplo, pode disparar, ao mesmo tempo, mudanças na ativações dos grupos neurais, mapas neurais e mapas globais que os compõem. É por este motivo que o que é sentido, percebido e feito surge simultaneamente. Pode-se dizer então que, no sistema nervoso, a percepção surge no mesmo instante da ação.

Nos últimos anos, uma série de esforços foram realizados para estender a Teoria da Seleção de Grupos Neurais à *consciência* (EDELMAN, 1989, 1995, 2004; SETH; BAARS, 2005). De acordo com Gerald Edelman, as funções cerebrais superiores são constituídas por uma tríade básica composta da percepção, da memória e da aprendizagem, concebidas através da noção de *categorização*. Gerald Edelman consolidou a sua teoria com simulações computacionais que mostram a verossimilhança de sua conjectura inicial. Para uma descrição pormenorizada e em sequência cronológica de todas as simulações computacionais desenvolvidas por Gerald Edelman, consulte Bastos (2005, p. 21–39).

2.3.6 Projeto de agentes de *software* contemporâneos

No âmbito da Inteligência Artificial, um agente de *software* contemporâneo é visto como um sistema dinâmico, onde a percepção e a ação constituem processos simultâneos e inseparáveis (vide Figura 2.2). Existe, então, uma substituição da metáfora do processamento de informação, advinda do paradigma tradicional das ciências cognitivas, pela metáfora da integração sensorio-motora.

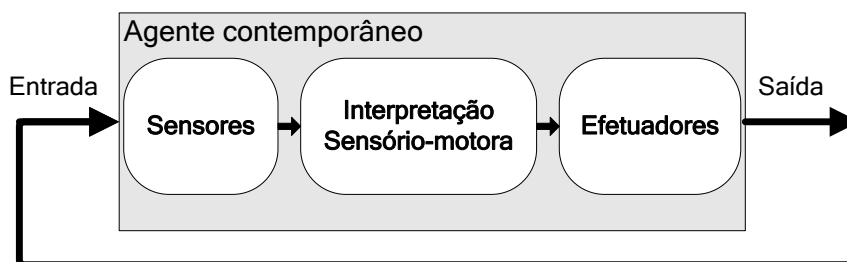


Figura 2.2: A visão contemporânea do processamento de informação

A seguir são descritos os princípios de projeto empregados na construção de agentes de *software* contemporâneos. Cabe salientar que ainda não existe nenhum agente que implemente todos estes princípios (FLORIAN, 2003). No entanto, estes princípios resumizam e tornam explícitos os conhecimentos adquiridos na área até o presente momento. Esse novo ramo de pesquisa dentro da Inteligência Artificial tem sido apresentado sob diversas denominações, dentre elas: *New Artificial Intelligence*, *Nouvelle Artificial Intelligence*, *Situated Activity*, *Situated Cognition*, *Embodied Cognition*, e *Behavior-Based Robotics*. Para maiores detalhes, consulte (CLANCEY, 1997; LUNGARELLA; PFEIFER; SANDINI, 2003; PFEIFER; IIDA; BONGARD, ; KRICHMAR; EDELMAN, 2006).

2.3.6.1 Princípio de um agente absoluto

De acordo com Pfeifer e Scheier (2001, p. 304-306), o projeto de um agente de *software* sempre envolve a definição de três componentes que são rigorosamente interconectados e mutuamente interdependentes:

1. Fixação de um nicho ecológico onde o agente irá atuar. A natureza de ambientes que um agente pode habitar varia significativamente. Nenhum agente pode adaptar-se, fisicamente e cognitivamente, para lidar com todas as variações possíveis.
2. Estabelecimento de comportamentos desejados ou tarefas a serem cumpridas;
3. Determinação do agente propriamente dito. A concretização de um agente deve priorizar três aspectos principais: *autonomia*, *localidade* e *corporificação*. Apesar de certa objeção por alguns pesquisadores da área, existe consenso da maioria de que a corporificação não é necessariamente dada pela materialidade, como a apresentada em animais ou robôs físicos, mas por uma relação dinâmica com o ambiente (FLORIAN, 2003). Assim, a pesquisa em inteligência artificial pode ser realizada com ambientes de simulação genuinamente computacionais⁵.

⁵O Apêndice A apresenta o ambiente de simulação robótica desenvolvido no âmbito deste trabalho.

2.3.6.2 *Princípio da redundância, do equilíbrio ecológico e do projeto econômico*

Um dos modelos mais adequados sobre a evolução da inteligência animal foi desenvolvido por Daniel Dennett (1996). Segundo ele, quanto mais precisa for a informação e quanto maior for a região de onde uma criatura coleta informação, mais provável será o seu sucesso. No início da evolução das espécies, apareceu o olfato que é uma forma de obter informação do ambiente próximo. Posteriormente, surgiram a audição e a visão permitindo a aquisição de informação de alta qualidade a distâncias maiores. A percepção de eventos a grande distância no espaço contribui, eventualmente, para a predição de eventos a grandes distâncias no tempo. Por exemplo, a visão permite à presa antever o ataque de um predador. Dennett (1996) classificou os seres vivos, de acordo com os seus processos de manipulação da informação, na seguinte hierarquia de complexidade: *darwiniana*, *skinneriana*, *popperiana* e *gregoriana*. Recentemente, João Paulo Schüler (2002) realizou uma recontextualização do domínio das criaturas de Dennett para o domínio de *software*.

No entanto, conforme as abordagens contemporâneas da cognição, não basta um agente de *software* possuir variados mecanismos de obtenção de informação, é necessário incorporar *redundância* nos dispositivos sensoriais. Em outras palavras, os sensores devem estar posicionados no agente de tal forma que exista sobreposição espacial nas informações adquiridas. A redundância promove correlações e associações entre as informações obtidas por diferentes modalidades sensitivas. Estas correlações ajudam o agente a reduzir drasticamente a incerteza do ambiente e a prever eventos (PFEIFER; SCHEIER, 2001, p. 446–455). Os sistemas visual e tátil, por exemplo, fornecem informação espacial do ambiente, apesar de se basearem em diferentes processos físicos. Duplicar meramente os componentes sensoriais não conduz a uma redundância útil, pois mesmo que um agente possua dez câmeras de vídeo de alta definição não conseguirá operar em um ambiente completamente escuro, diferentemente de um agente que tenha um sistema tátil em adição a um sistema visual. Além da redundância, deve existir um *equilíbrio* da complexidade do agente (sistemas sensório, motor e de controle) com a complexidade de seu ambiente de tarefa. Um sistema de controle extremamente complexo é desnecessário se o agente ou o ambiente são extremamente simples (PFEIFER; SCHEIER, 2001, p. 455–463). Um agente com um sistema de controle demasiadamente elaborado, via de regra, tem problemas no aprendizado em decorrência do excesso de ajustes do modelo. Por outro lado, um agente com um sistema de controle muito simples pode ter dificuldades em adaptar-se às circunstâncias ambientais. Em suma, a física da interação agente-ambiente e as restrições impostas pelo nicho ecológico devem ser exploradas (PFEIFER; SCHEIER, 2001, p. 435–445), tornando *econômico* o projeto do agente sob o ponto de vista computacional.

2.3.6.3 *Princípio de processos paralelos fracamente acoplados*

Agentes de *software* devem possuir diversos comportamentos a fim de cumprirem determinadas tarefas. Alguns comportamentos são compatíveis, entretanto, outros são mutuamente exclusivos. Por isso, uma decisão deve ser tomada para selecionar, a cada momento, aquelas ações que são coerentes com o contexto atual do agente e do ambiente.

A solução mais direta é assumir que existe um módulo ou representação interna para cada categoria do comportamento observado. Por exemplo, postular a existência de um módulo interno de seguir paredes quando se observa um rato andando rente a paredes. A maioria das abordagens computacionais propõe o uso de arquiteturas hierárquicas para realizar uma decomposição funcional (e.g., Newell e Simon, 1963; Fikes and Nilsson, 1971). A idéia é que uma funcionalidade global (e.g., seguir paredes) é decomposta

em subfuncionalidades diferentes que, por sua vez, podem ser segregadas novamente em módulos e submódulos (e.g., detecção de parede, mover-se rente a uma parede). A noção de hierarquias implica um processo centralizado e seqüencial e, portanto, não tolerante a falhas: quando um módulo é removido ou deixa de funcionar, a operação do sistema como um todo fica afetada. Isso não reflete o que realmente ocorre em sistemas naturais, além de gerar inúmeros problemas de projeto (PFEIFER; SCHEIER, 2001).

Um mecanismo que vem despontando como uma nova opção para essa problemática é assumir um número elevado de *processos heterogêneos, paralelos e fracamente acoplados* que são conectados ao aparato sensório-motor do agente (PFEIFER; SCHEIER, 2001, p. 327–376). Estes processos não necessitam de um supervisor; ou seja, o controle é descentralizado e distribuído. A arquitetura de controle pode ser construída de forma gradual, com a adição de novos processos, assim como acontece na evolução biológica. Além disso, a união de todos os processos promove a emergência de novos comportamentos, os quais não foram previstos no projeto. Por exemplo, o ato de seguir paredes é compreendido como a união de processos básicos que, juntos, na interação com o meio ambiente, produzem o comportamento desejado.

A *arquitetura de subsunção* foi originalmente proposta por Rodney Brooks (1986). Nessa arquitetura, os comportamentos são organizados em níveis de acordo com uma *hierarquia de subordinação*. Os níveis inferiores são prioritários em relação aos comportamentos dos níveis superiores, dado que as ações de níveis mais altos representam comportamentos mais abstratos e elaborados. Como vários comportamentos podem ser acionados em paralelo, um mecanismo de controle seleciona, entre os diferentes comportamentos possíveis, aquele que é considerado ótimo em cada momento. Um número considerável de robôs foi construído utilizando variações desta arquitetura, dentre eles: Myrmix, Ghengis, Herbert e Hannibal (PFEIFER; SCHEIER, 2001, p. 199–225).

Dimitrios Lambrinos e Christian Scheier (1995) desenvolveram uma arquitetura inspirada nos *veículos de Braitenberg*. O esquema básico desta arquitetura consiste em associar diretamente os sensores do agente com processos internos (e.g., andar-para-frente). Esta associação pode ser ponderada por algum valor, a fim de realizar uma modulação da importância de um sensor sobre um determinado processo. Os processos internos, por sua vez, podem estar conectados a outros processos internos bem como a processos motores (e.g., motor-esquerdo, motor-direito). O comportamento a ser executado pelo agente é simplesmente obtido pelo somatório linear da saída dos processos motores.

2.3.6.4 Princípio da coordenação sensório-motora

Nesta última década, o estudo da *integração sensório-motora* tem sido um tópico de pesquisa bastante ativo na área das ciências cognitivas (PFEIFER; SCHEIER, 2001, p. 377–434). Existe consenso entre os pesquisadores da área que, em organismos biológicos, as informações oriundas de múltiplos sentidos (visão, audição, tato, etc.) são integradas e, diretamente, mapeadas sobre um conjunto apropriado de comandos motores (músculos e glândulas). Este processo sensório-motor, quando aplicado a agentes de *software*, conduz a inúmeros benefícios e simplificações de projeto, conforme descrito em (PFEIFER; SCHEIER, 1994, 1997; NOLFI; PARISI, 1999; NOLFI, 2002).

Pfeifer e Scheier (1997) demonstraram que, através da *coordenação sensório-motora*, agentes de *software* podem estruturar suas percepções e, por meio disto, induzir *regularidades* que significativamente simplificam a aprendizagem. Dados sensoriais não são apenas adquiridos mas, sobretudo, gerados e correlacionados. A correlação reduz a alta *dimensionalidade* presente nos dados obtidos do ambiente que, por sua vez, capacita o

agente a fazer associações entre diferentes modalidades sensitivas (BOEKHORST; LUNGARELLA; PFEIFER, 2003). A coordenação sensório-motora possibilita, ainda, resolver um dos maiores desafios dos roboticistas: a *categorização* de objetos físicos (SCHEIER; PFEIFER, 1995; SCHEIER; LAMBRINOS, 1996; LAMBRINOS; SCHEIER, 1996).

Qualquer agente de *software* deve ser capaz de realizar distinções entre diferentes tipos de objetos; ou seja, deve ter a competência de categorizar os objetos físicos presentes no meio. Realizar distinções é um processo bastante rotineiro e natural para todos os seres vivos. Contudo, ainda constitui em uma árdua tarefa para os robôs. Uma das principais razões para essa problemática é que um objeto pode conduzir a inúmeros padrões diferentes e ambíguos, dependendo do ângulo de visão do agente, de condições de luminosidade do ambiente, de ruído associado a sensores e efetadores, e assim por diante.

Tradicionalmente, o problema do reconhecimento de objetos tem sido investigado pelo emprego de modelos clássicos da Inteligência Artificial, onde a categorização é definida em termos de um mapeamento de estímulos sensoriais em representações internas (e.g., Kruschke, Jan. 1992). Em muitas arquiteturas de redes neurais artificiais emprega-se uma camada de entrada, para codificar as características dos objetos, e uma camada de saída, que representa as categorias propriamente ditas. Tipicamente, através de métodos de aprendizagem supervisionada, o objetivo é aprender uma associação entre ativações na camada de entrada e ativações correspondentes na camada de saída.

Clark e Thornton (1997) introduziram o conceito de *problemas do tipo 2* para denotar conjuntos de dados para os quais o mapeamento da entrada para a saída não pode ser extraído por algoritmos de aprendizagem ou procedimentos estatísticos. De forma análoga, sempre que o mapeamento pode ser aprendido diretamente a partir de dados de entrada, tem-se um *problema do tipo 1*. Frequentemente, problemas do tipo 2 podem ser transformados em problemas do tipo 1. Contudo, a transformação apropriada não é conhecida *a priori*. Então, a questão principal no aprendizado de categorias pode ser resumida em como realizar essa conversão. Uma estratégia eficiente que pode ser empregada para resolver isto é explorar a coordenação sensório-motora (NOLFI; PARISI, 1999).

Pfeifer e Scheier (1997) propuseram uma série de modelos de agentes de *software* para explorar a coordenação sensório-motora. No *modelo SMC I*, o agente aprende a discernir entre objetos circundando em torno deles. Este comportamento exploratório permite diferenciar os objetos por meio de invariâncias que são identificadas (e.g., diferença na velocidade angular). No *modelo SMC II*, a categorização é realizada por uma associação reentrante entre as modalidades sensitivas da visão e do tato. Esses dois trabalhos são amparados pelo estudo conduzido por Bushnell e Boudreau (1993) sobre o desenvolvimento de capacidades motoras em crianças recém nascidas. Diante dos resultados obtidos, os autores sugerem que a aprendizagem é drasticamente simplificada quando as crianças manipulam objetos em frente de suas faces a uma distância fixa.

2.3.6.5 Princípio da aprendizagem

Os seres vivos têm a capacidade de aprender. O *aprendizado*, do ponto de vista da neurociência, ocorre através de mudanças estruturais nas conexões sinápticas entre os neurônios. Estas alterações podem ser realizadas de variadas formas, o que acaba por gerar inúmeros tipos de aprendizagem. O presente trabalho não discute em detalhes cada uma das abordagens existentes, entretanto, dá ênfase aos aspectos relevantes à pesquisa em inteligência artificial. A Tabela 2.2 apresenta um esboço geral dos principais paradigmas de aprendizagem, a saber: não-associativo e associativo.

Tabela 2.2: Visão geral dos principais paradigmas de aprendizagem

Classe	Paradigma	Descrição	Regra(s)
Não associativo	Habituação	Aprendizagem baseada na redução da intensidade de resposta a estimulações repetitivas	Regra de Stanley, Regra de Wang & Arbib
	Sensibilização	Aprendizagem baseada no aumento da intensidade de resposta a estimulações repetitivas	–
Associativo	Supervisionado	Aprendizagem baseada em informação <i>a priori</i> (ou <i>a posteriori</i>) sobre as relações causa-efeito entre as ações do agente e o estado do ambiente	regra Delta, regra de retropropagação de erros (<i>backpropagation</i>), arquitetura Outstar
	Não-supervisionado	Aprendizagem baseada somente em correlações de dados de entrada, sem nenhum tipo de conhecimento <i>a priori</i>	regra de Hebb, regra de Kohonen
	Auto-supervisionado	Aprendizagem baseada em recompensas/punições resultantes do comportamento efetuado pelo agente no ambiente	Q-learning, variações de Hebb

O *aprendizado não-associativo* descreve a mudança comportamental que ocorre ao longo do tempo em resposta a um único tipo de estímulo. Assim, a repetição de um estímulo proporciona o aprendizado sobre suas propriedades. Dois tipos de aprendizado que se enquadram nessa categoria são a *habituação* e a *sensibilização*.

A *habituação* é uma forma simples de aprendizado em que um estímulo neutro é repetido inúmeras vezes. Na primeira vez, o estímulo é uma novidade e evoca uma determinada reação no organismo. No entanto, à medida que o estímulo é repetido evoca cada vez menos resposta. Por fim, após um período de tempo, o estímulo passa a ser ignorado provavelmente porque não agrega nenhum significado ao organismo. Um exemplo familiar é a percepção do barulho de carros em uma avenida movimentada que, após algum tempo, pode diminuir e até mesmo não ser mais notado. Do ponto de vista neural, a depressão da atividade sináptica entre interneurônios e neurônios motores parece ser o mecanismo geral de habituação (LENT, 2002).

A *sensibilização* é um processo de aprendizado pelo qual um organismo aprende a ter uma resposta mais intensa a estímulos que previamente não evocavam nenhum tipo de reação. Em outras palavras, um organismo aprende a reconhecer um estímulo com propriedades nocivas, respondendo eficazmente a ele e a outros que a ele se assemelhem, mesmo que inócuos. É o que ocorre, por exemplo, depois que uma pessoa ouve um forte barulho, como em uma batida de carro ou em uma explosão. Nesse caso, pequenos ruídos passam a assustá-la, sendo que se esses ruídos não fossem antecidos pelo barulho, nem seriam percebidos. Acredita-se que essa forma de aprendizado ocorra devido a um mecanismo de facilitação presente no sistema nervoso de animais (LENT, 2002).

O *aprendizado associativo* refere-se à associação de eventos, onde ocorre a ligação de estímulos que anteriormente não estavam relacionados. Existem duas formas de aprendizagem associativa: por *condicionamento clássico* e por *condicionamento instrumental*.

O aprendizado por *condicionamento clássico* foi inicialmente descrito, em situações

experimentais, pelo russo Ivan Pavlov. O cientista ensinou cães a salivar quando ouviam o som de campainha, ao parear o estímulo neutro da campainha a uma recompensa alimentar. A idéia é que os cães, ao sentir a presença de um alimento por meio do olfato ou da visão, começam a salivar. O experimento consistiu em tocar uma sineta enquanto os cães eram alimentados. Mais tarde, após algumas repetições, Pavlov constatou que em pouco tempo os cães respondiam com salivação ao soar da campainha, que passou a ser um estímulo condicionado e a provocar o reflexo da salivação mesmo sem a presença da comida. Ao descrever esta experiência, Pavlov denominou a salivação sob a influência do alimento de reflexo não-condicionado (UR); o próprio alimento de estímulo não-condicionado (US); a salivação sob a influência da campainha de reflexo condicionado (CR); e a campainha de estímulo condicionado (CS). Pavlov chamou esta modalidade de aprendizagem de reflexo condicionado. No contexto de técnicas de aprendizagem de máquina, este paradigma se enquadra na definição do *aprendizado supervisionado*.

A base do *condicionamento instrumental* (ou operante), designado pelo psicólogo Burrhus Frederic Skinner, é a associação entre o estímulo e o comportamento do animal. A resposta está relacionada à probabilidade de ocorrência de uma recompensa ou punição. A pesquisa desenvolvida por Thorndike demonstrou que os animais aprendem conforme as conseqüências de suas ações; ou seja, todo comportamento animal é influenciado por seus efeitos, sejam eles benéficos ou maléficos. No experimento realizado por Thorndike, um cão (ou um gato) deve aprender uma seqüência de ações específicas para escapar de uma caixa ("*puzzle box*"). Assim, o animal inicia timidamente emitindo uma variedade de respostas que estão no seu repertório natural de comportamentos. O treinador pode então selecionar uma resposta para reforçar. Devido ao reforço, esta resposta é emitida com maior freqüência, enquanto outras são paulatinamente eliminadas. No âmbito da comunidade de aprendizagem de máquina, este modelo de aprendizagem é empregado de modo estritamente técnico, compreendendo o *aprendizado por reforço* e o *aprendizado baseado em valor*. Apesar de similaridades entre estas duas propostas, o presente trabalho foca na abordagem baseada em valor, por ser mais adequada e compatível com os ideais de agentes de *software* situados e corporificados (PFEIFER; SCHEIER, 2001, p. 467–501).

As pesquisas realizadas por neurocientistas revelam que o processo de aprendizagem animal é modulado pela atribuição, de diferentes valores, aos estímulos recebidos (EDELMAN, 1995). Os estímulos que chegam ao cérebro, tanto externos quanto internos, são tratados com importância distinta. Por exemplo, a sensação causada pela destruição da pele de um animal recebe um valor bem maior do que uma simples pressão na mesma área. De forma análoga, estímulos externos que identificam alimentos recebem um valor maior do que um equivalente não alimentar. Acredita-se que este processo é realizado por *neuromoduladores* presentes em regiões cerebrais especializadas, denominadas por *sistemas de valores* (EDELMAN, 1995). Cada espécie tem o seu sistema de valores (também denominado de centro hedonista ou sistema homeostático), que inclui informações do seu corpo e informações do ambiente. Em síntese, o sistema de valores identifica os sinais mais salientes e guia de certa forma o comportamento animal.

As técnicas de aprendizagem baseada em valor buscam inspiração nos sistemas de valores dos seres vivos. Basicamente, estas técnicas se enquadram em duas categorias principais: *explícita* e *implícita*. Em ambas as categorias, a idéia é que *forças motivacionais intrínsecas* influenciam, direta ou indiretamente, o comportamento do agente de *software*. As forças motivacionais podem ser compreendidas como influências, sejam externas ou internas, que codificam o que é "valioso" para o agente. Isto faz com que o agente prefira um determinado comportamento em detrimento de outro.

Na *categoria implícita* da aprendizagem baseada em valor, a modulação é realizada por mecanismos que selecionam determinadas interações com o ambiente, conduzindo a um aumento na capacidade de adaptação do agente de *software* ao seu nicho ecológico. Geralmente, os mecanismos empregados são *reflexos básicos* introduzidos pelo projetista do sistema. Estes reflexos aumentam a probabilidade do agente receber um valor explícito para uma ação particular. Por exemplo, a capacidade de distinguir entre itens alimentares e itens venenosos baseado em dicas exteroceptivas, como a visão. Esta habilidade direciona o agente à procura por comida de forma mais rápida e, por isto, proporciona uma melhor adaptação ao meio ambiente.

Na *categoria explícita* da aprendizagem baseada em valor, sinais de valores são gerados como consequência do comportamento do agente de *software*. Estes sinais podem ser *neurais* ou *hormonais* e são utilizados posteriormente para modular o processo de aprendizagem. Em arquiteturas baseadas em redes neurais artificiais, esta modulação está relacionada com a regra de aprendizagem neural, ou seja, o quão rápido uma mudança nas conexões sinápticas ocorre. Tipicamente, a modulação é codificada através de uma versão modificada da *regra de aprendizado por coincidência hebbiano*⁶.

As pesquisas neurofisiológicas confirmam que a idéia da regra de aprendizado por coincidência hebbiano é aproximadamente correta, muito embora isto ocorra de uma forma muito mais complexa do que o simples “aumento da eficiência”. O efeito do reforço da conexão entre dois neurônios, quando um contribui para o disparo do outro, pode ser simulado matematicamente ajustando-se o peso nesta conexão por uma constante multiplicada pelo produto de seus valores de saída, conforme é apresentado na Equação 2.1.

$$\Delta w_{ij} = \eta \cdot a_i \cdot a_j, \quad (2.1)$$

onde w_{ij} é o peso da conexão sináptica entre o neurônio pré-sináptico j e o neurônio pós-sináptico i , η é o parâmetro da taxa de aprendizagem, e a_i e a_j representam as ativações dos neurônios pós e pré-sináptico, respectivamente. A Equação 2.2 ilustra a idéia básica subjacente à aprendizagem explícita baseada em valor. Matematicamente, adiciona-se um termo multiplicativo à regra de Hebb, tal que:

$$\Delta w_{ij} = V \cdot \eta \cdot \bar{a}_i \cdot \bar{a}_j, \quad (2.2)$$

onde V é o termo relacionado à contribuição do sistema de valores, e \bar{a}_i e \bar{a}_j representam as ativações dos neurônios pós e pré-sináptico, respectivamente, fixadas em uma janela temporal. No caso em que $V > 0$, o peso da conexão sináptica entre os neurônios i e j é fortalecida. Caso contrário, se $V < 0$, ela é enfraquecida. Em especial, quando $V = 0$, não existe atualização sináptica e, conseqüentemente, não há aprendizado. Como pode ser observado, a modulação da aprendizagem acontece *a posteriori*. Por isto, é necessário ter uma espécie de memória da história recente das ativações neurais responsáveis pelo comportamento que disparou o sinal de valor. Tipicamente, a forma mais simples de codificar tal memória é estabelecer uma média temporal das ativações neurais (vide termos \bar{a}_i e \bar{a}_j na Equação 2.2). Como resultado, os neurônios que conduziram a um determinado comportamento ainda estão suficientemente ativos quando o sinal de valor é disparado.

Não existe ainda uma teoria unificada ou uma abordagem comumente aceita na comunidade científica para a aprendizagem robótica. Dessa forma, não há consenso de qual

⁶A teoria de Hebb é baseada na observação que, em sistemas biológicos, quando um neurônio contribui para o disparo de outro neurônio, a conexão entre eles é reforçada (HEBB, 1949).

o melhor paradigma a ser utilizado em agentes de *software*. Contudo, pode-se enumerar uma série de características desejáveis em um algoritmo de aprendizagem: robustez e tolerância a ruídos; convergência rápida; tratabilidade computacional; adaptatividade a eventuais mudanças ambientais; operação em modo seqüencial (*on-line*); e dependência de informações que possam apenas ser extraídas de sensores do agente, e não aquelas fornecidas por um projetista ou por um observador externo.

2.4 Considerações do capítulo

Este capítulo apresentou os princípios, modelos e métodos de investigação de cada uma das vertentes da ciência da cognição. Diante do exposto, pode-se argumentar que a categoria tradicional da ciência cognitiva assume a realidade como objetiva, pré-dada e independente do sujeito e, portanto, o saber e conhecer humanos são vistos como representações ou modelos internos da realidade externa. Já a categoria contemporânea da ciência cognitiva parte do princípio de que a realidade é construída pelo próprio sujeito no seu curso de interação com o meio ambiente e, por esse motivo, o mundo não é pré-dado e não há representações ou modelos internos da realidade externa.

De acordo com os fundamentos teórico-conceituais de cada uma das vertentes da ciência da cognição, a Inteligência Artificial busca projetar e desenvolver mecanismos de *software* e/ou *hardware* que apresentem comportamento inteligente. Em sua grande maioria, os desenvolvimentos têm se embasado em abordagens tradicionais, alcançando relativo sucesso. Por outro lado, as abordagens contemporâneas oferecem uma grande síntese, genuinamente transdisciplinar, que surgiu por meio das contribuições de diversas áreas do conhecimento que têm se ocupado com o fenômeno da cognição em seus múltiplos aspectos, tais como: neurociência, biologia, física, lingüística, sociologia e antropologia.

Os agentes de *software* construídos em conformidade com as abordagens tradicionais, tanto cognitivistas quanto conexionistas, se encontram bem desenvolvidos com relação a suas pesquisas. Os pesquisadores desta área, em sua grande parte, estão preocupados em aprimorar as técnicas e ferramentas já existentes. Cabe salientar que em virtude do representacionismo e da separação em sentir-pensar-agir inerente às abordagens clássicas das ciências da cognição, inúmeros problemas fundamentais são inseridos na modelagem de agentes de *software*, conforme discutidos na Seção 2.2.3.

O desenvolvimento de agentes de *software* que partilham dos conceitos derivados das abordagens contemporâneas da cognição ainda é incipiente. Recentemente, Pfeifer e Scheier (2001) propuseram cinco princípios de projeto para serem empregados no projeto de agentes de *software* situados e corporificados, conforme expostos na Seção 2.3.6. No entanto, ainda existe uma carência muito grande por uma fundamentação teórica, uma arquitetura, uma metodologia de criação e ferramentas que estejam definidas e difundidas no meio acadêmico. Isto implica um amplo campo de pesquisas a ser explorado. Alguns esforços iniciais já foram realizados, como, por exemplo, por Santos (2003), Verschure e Althaus (2003), Krichmar e Edelman () e Harter e Kozma (May 2005).

A maioria dos trabalhos que vêm sendo desenvolvidos na área da Inteligência Artificial se baseia no paradigma conexionista, com a utilização de redes neurais artificiais. Ressalta-se que isso só não vai de encontro com os ideais contemporâneos da cognição desde que a rede neural seja adequadamente incorporada à arquitetura de controle do agente e este, por sua vez, seja situado e corporificado em um ambiente simulado ou real. Assim, o comportamento do agente é dado pela rede neural em conjunto com a física de seu corpo. O próximo capítulo apresenta os fundamentos de redes neurais artificiais.

3 FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS

Este capítulo apresenta alguns dos principais conceitos relacionados a redes neurais artificiais, com o intuito de servir de referência ao leitor para a nomenclatura no restante do texto. Inicialmente, discute-se brevemente o funcionamento do sistema nervoso, em especial do neurônio biológico. Em seguida, faz-se uma rápida análise sobre os componentes básicos de um neurônio artificial, em particular do neurônio de McCulloch e Pitts, e caracteriza-se uma rede neural artificial genérica, enfatizando o seu funcionamento, a sua topologia e o seu processo de aprendizado. Logo após, apresenta-se a *rede neural auto-organizável baseada em grades* proposta por Teuvo Kohonen, discute-se sobre determinadas métricas de convergência e estabilidade da rede e examina-se algumas propriedades do mapa de características. Por fim, discorre-se sobre os principais benefícios e aplicações das redes neurais artificiais. As informações aqui contidas foram extraídas principalmente dos seguintes livros-texto: (HAYKIN, 2001; KOVÁCS, 2002; CONNORS; BEAR; PARADISO, 2002; LENT, 2002; KOHONEN, 2001). Em (KASKI; KANGAS; KOHONEN, 1998) e (OJA; KASKI; KOHONEN, 2002) são apresentadas 5435 referências de artigos sobre o mapa auto-organizável de Kohonen, publicados entre os anos de 1981 e 2002.

3.1 Introdução

O estudo do cérebro é atrativo, sob o ponto de vista da computação, por propiciar a construção de modelos de processamento de informação biologicamente plausíveis, como é o caso das redes neurais artificiais. Essas, representam uma tentativa de superar as limitações que o modelo de um computador digital apresenta, buscando, para tanto, imitar os princípios e mecanismos que regem o funcionamento do cérebro. Dentre as principais características do cérebro desejáveis em um sistema computacional, pode-se enumerar: alto grau de paralelismo; robustez e tolerância a falhas; elevada imunidade a ruído; e capacidade de adaptação e auto-organização através da experiência e do aprendizado.

Uma rede neural artificial pode ser vista como um sistema de computação paralela com parâmetros distribuídos, os quais podem ser ajustados através de algoritmos de treinamento eficientes. Embora o nível de complexidade que se possa representar em uma rede neural artificial seja muito inferior àquele apresentado pelo cérebro humano, que lhe serviu de inspiração, sua capacidade de aprendizado fez com que se tornasse uma tecnologia promissora e uma candidata natural para realizar tarefas para as quais soluções algorítmicas não sejam eficientes ou mesmo conhecidas (ENGEL, 2001). Por esta razão, elas têm sido largamente aplicadas em tarefas relacionadas com o reconhecimento de padrões, processamento de sinais, identificação de sistemas e controle.

Nas últimas décadas, progressos significativos têm sido realizados na área, tanto no surgimento de novas arquiteturas de redes, quanto na diversidade de aplicações. No en-

tanto, alguns dos principais conceitos básicos que permitem a compreensão de sua estrutura e funcionamento geral permanecem válidos até hoje. Por esse motivo, neste capítulo não se discute de forma exaustiva todo o ferramental teórico-conceitual existente mas, pelo contrário, apenas apresenta-se o essencial para a compreensão do presente trabalho.

3.2 O neurônio biológico

O sistema nervoso dos seres vivos é constituído por um emaranhado de células que se interconectam de forma específica e precisa, formando *circuitos neurais* funcionais. Basicamente, há dois tipos de células presentes no sistema nervoso: *célula nervosa* (ou *neurônio*) e *célula glial* (ou *gliócito*) (CONNORS; BEAR; PARADISO, 2002). O neurônio é a unidade sinalizadora do sistema nervoso, cuja morfologia está adaptada para funções de transmissão e processamento de sinais. Já os gliócitos desempenham funções de infraestrutura: nutrem, dão sustentação mecânica, controlam o metabolismo dos neurônios, ajudam a construir o tecido nervoso durante o desenvolvimento e funcionam como células imunitárias.

De acordo com Lent (2002, p. 14), classicamente considera-se o neurônio como a unidade morfofuncional fundamental do sistema nervoso e o gliócito como a unidade de apoio. Por essa razão, nos últimos anos a pesquisa na área da neurociência se concentrou na investigação do papel dos neurônios no processamento de informação. Apesar da localização e da função dos neurônios variarem, em geral todos compartilham a mesma estrutura básica. De maneira extremamente simplificada, um neurônio é formado por:

- um *corpo celular* ou *soma* que fornece suporte metabólico e contém o *núcleo* da célula, onde está armazenado o código genético;
- diversos *dendritos* que consistem em variados prolongamentos próximo ao corpo celular, em forma de ramificações, e funcionam como receptores de sinais de outros neurônios; e
- um *axônio* que compreende um prolongamento fino e longo, através do qual sinais são enviados para outros neurônios. Esse prolongamento é ramificado na sua extremidade e forma uma *arborização terminal*.

A Figura 3.1 mostra, de forma esquemática, a estrutura de um neurônio biológico.

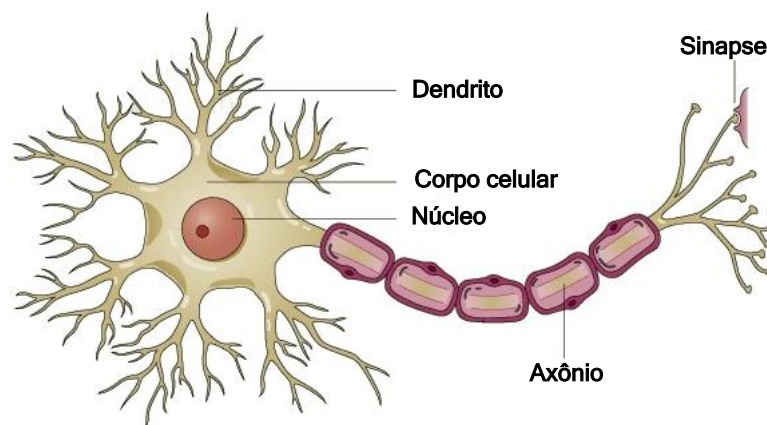


Figura 3.1: Elementos básicos de um neurônio biológico

O *impulso nervoso* (ou *potencial de ação*) é o principal sinal de comunicação utilizado por um neurônio. A origem, propagação e transmissão do impulso nervoso são devidas às propriedades eletroquímicas da membrana que envolve o neurônio. A membrana separa dois meios caracterizados por concentrações iônicas diferentes. Em estado de repouso, o interior da célula nervosa é constituído por grande quantidade de *íons de potássio* K^+ e o exterior por *íons de sódio* Na^+ e de *cloro* Cl^- . Nesse caso, a diferença de potencial entre o interior e o exterior da célula é da ordem de -60 mV. Quando um estímulo chega a um receptor ou terminação nervosa, sua energia causa uma inversão temporária de cargas na membrana. Como consequência, a diferença de potencial passa a ser positiva, com o valor aproximado de $+40$ mV, e a célula fica mais propensa a gerar um sinal.

Um sinal pode ser caracterizado como um pulso elétrico gerado pela membrana, rápido e invariável, que se propaga com enorme velocidade ao longo do axônio. Ao chegar à extremidade do axônio do neurônio, o impulso nervoso provoca a emissão de uma mensagem química, conhecida como *neurotransmissor*, que, por meio de uma zona de contato (a *sinapse*), leva a informação – intacta ou modificada – para a célula seguinte. Dependendo do tipo de neurotransmissor, a conexão sináptica pode ser *excitatória* ou *inibitória*. Uma conexão excitatória contribui para a formação de um impulso nervoso na célula seguinte enquanto que uma conexão inibitória age no sentido oposto.

De forma geral, os neurônios que compõem o sistema nervoso de qualquer ser vivo podem ser classificados em três tipos fundamentais (LENT, 2002): *aférente* (ou sensitivo), *eferente* (ou motor) e *de associação* (ou internuncial). Somente os neurônios sensitivos e motores realizam a interface do sistema nervoso com o meio ambiente ou com outros órgãos que formam o organismo. No entanto, todos estabelecem inúmeros circuitos fechados entre si através de suas sinapses. Por essa razão, o cérebro humano é a estrutura mais complexa conhecida. Atualmente, calcula-se que o cérebro possui 30 bilhões de neurônios, sendo que $1/3$ deles está localizado no córtex cerebral. Como cada um possui cerca de 10.000 sinapses, é possível estimar-se que o número de sinapses seja da ordem de 10^{15} . Gerald Edelman (1999) calculou que o número de circuitos funcionais possíveis ultrapassa, em muitos milhares, a quantidade de partículas sub-atômicas do universo. Desta forma, nenhum cérebro é igual a outro, ainda que todos sejam dependentes, para um funcionamento adequado, de uma mesma organização física.

3.3 O neurônio artificial

O primeiro modelo de neurônio foi proposto por Warren McCulloch e Walter Pitts em 1943, em uma tentativa de formalizar o conhecimento da época sobre o funcionamento do neurônio biológico (KOVÁCS, 2002, p. 27–38). A Figura 3.2 apresenta um diagrama de blocos do neurônio artificial de McCulloch e Pitts, denominado unidade de limiar linear.

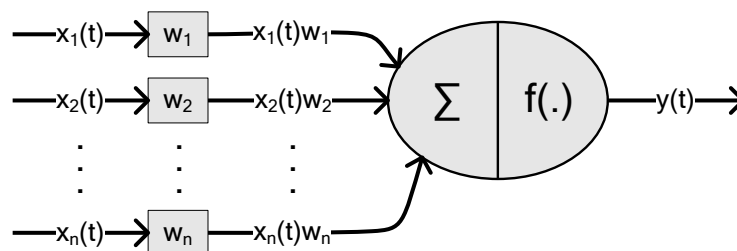


Figura 3.2: Unidade de limiar linear de McCulloch e Pitts

Formalmente, o funcionamento do modelo de neurônio de McCulloch e Pitts pode ser descrito da seguinte maneira. Considere o i -ésimo neurônio de uma rede neural com n neurônios. Este neurônio é caracterizado pelo valor σ_i , chamado *nível de ativação* do neurônio, e pelo valor y_i , chamado *atividade* do neurônio. O nível de ativação corresponde ao potencial da membrana do neurônio biológico. A atividade do neurônio compreende a taxa média de disparos dos potenciais de ação do neurônio biológico. No modelo de McCulloch e Pitts, o nível de ativação é definido da seguinte forma:

$$\sigma_i(t) = \sum_{j=1}^n x_j(t)w_{ij} \quad (3.1)$$

onde $w_{ij} \in \mathbb{R}$ é o peso atribuído àquela entrada do neurônio i cuja origem é a atividade $x_j(t)$ do neurônio j , e $t \in \mathbb{Z}$ indica o tempo, sendo este um valor discreto. O peso simula a sinapse entre dois neurônios e quando estes não estão conectados assume o valor zero. Assim como no caso biológico, as sinapses podem ser tanto excitatórias como inibitórias. Uma sinapse excitatória tem o seu peso positivo, $w_{ij} > 0$, e uma sinapse inibitória tem o seu peso negativo, $w_{ij} < 0$. A atividade do neurônio i no instante t é dada por $f(\sigma_i(t))$. A função f , chamada *função de ativação* ou de *transferência*, adotada neste modelo é a *função degrau* com limiar de disparo dado por α :

$$f(x) = \begin{cases} 0 & \text{se } x \leq \alpha \\ 1 & \text{se } x > \alpha \end{cases} \quad (3.2)$$

onde α é o limite de disparo.

O modelo de neurônio artificial de McCulloch e Pitts tem duas sérias limitações (BITTENCOURT, 2001, p. 301–309): a sua natureza binária e a ausência de uma forma sistemática para determinar os pesos e limiares, sendo necessário que esses valores sejam definidos manualmente, baseado na experiência empírica do projetista. No entanto, mesmo com este modelo rudimentar de neurônio, McCulloch e Pitts foram capazes de provar que uma rede neural é equivalente a uma máquina de Turing e, portanto, capaz de calcular qualquer função computável (MCCULLOCH; PITTS, 1943 apud BITTENCOURT, 2001, p. 304). Essa equivalência deve-se ao fato de que é possível simular os operadores booleanos elementares (E, OU e NÃO) utilizando redes neurais, bastando para isso ajustar os pesos das conexões e o valor do limiar. Desta forma, a partir destes operadores, pode-se construir um computador convencional.

Recentemente, inúmeras melhorias e modificações foram realizadas no modelo de McCulloch e Pitts, no tocante à plausibilidade biológica e à tratabilidade computacional como, por exemplo, em (IZHIKEVICH, 2003). De forma resumida, os seguintes aspectos foram generalizados da proposta original: o nível de ativação passa a ser definido como uma função qualquer g das atividades dos neurônios, $\sigma_i = g(x_1, \dots, x_n)$; a função de ativação f passa a ser uma função limitada qualquer; e é introduzido um valor de polarização $\theta \in \mathbb{R}$, de modo que a atividade de um neurônio passa a ser calculada por $y_i = f(\sigma_i + \theta)$.

A escolha da função f pode influenciar drasticamente a complexidade e o desempenho de uma rede neural. Duch e Jankowski (1999) propõem uma taxonomia para caracterização e classificação de funções de ativação. A Figura 3.3 apresenta três exemplos de funções de ativação. A *função de ativação linear* é definida por $f(x) = \alpha x$, sendo $\alpha \in \mathbb{R}$ uma constante que regula a intensidade da atividade de x . A *função rampa* é uma função linear limitada pelo intervalo $[-\varphi, +\varphi]$. Por fim, a *função sigmóide* é a mais utilizada em redes neurais por ser monotônica, diferenciável em todos os pontos e produzir uma saída

gradual e não-linear. Em geral, a função sigmóide assume as seguintes formas: logística, definida no intervalo $[0, 1]$, e tangente hiperbólica, definida no intervalo $[-1, +1]$.

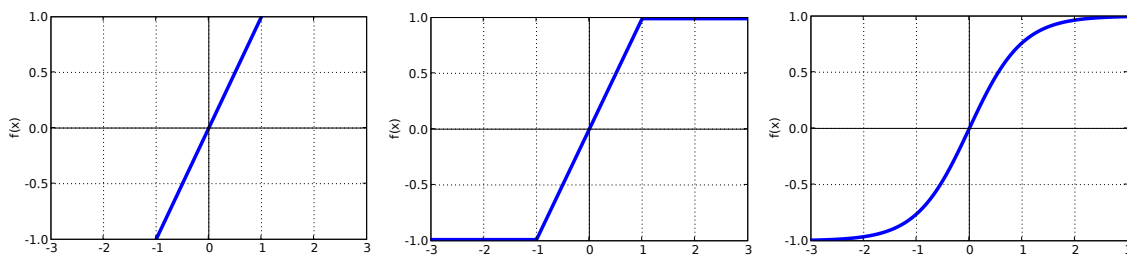


Figura 3.3: Exemplos de funções de ativação. À esquerda, função linear. No centro, função rampa. À direita, função sigmóide (tangente hiperbólica).

3.4 A rede neural de múltiplos neurônios

Uma vez definido um neurônio, é possível estudar as propriedades de redes de neurônios interconectados, as chamadas redes neurais. Resumidamente, as principais características de uma rede neural artificial são comentadas a seguir, conforme exposto por Haykin (2001). Do ponto de vista do fluxo de informação, uma rede pode conter três tipos de neurônios: *de entrada*, análogos aos neurônios sensoriais dos seres vivos; *de saída*, análogos aos neurônios motores; e *internos*, análogos aos neurônios associativos. Do ponto de vista funcional, uma rede pode ser *homogênea*, se todos os neurônios se comportarem da mesma forma, ou, no caso contrário, *heterogênea*. Quanto a conectividade, uma rede pode ser *fracamente* (ou parcialmente) conectada e *completamente* conectada. Quanto a topologia da interligação entre neurônios, uma rede pode ser *de alimentação direta* ou *acíclica* (do inglês *feed-forward*) e *recorrente* ou *cíclica* (do inglês *feedback*). Uma rede de alimentação direta representa uma função de sua entrada atual; assim, ela não tem nenhum estado interno além dos pesos propriamente dito. Por outro lado, uma rede recorrente utiliza suas saídas para alimentar de volta suas próprias entradas. Isso significa que os níveis de ativação da rede formam um sistema dinâmico que pode atingir um estado estável, exibir oscilações ou, até mesmo, apresentar um comportamento caótico.

No final da década de 1950, Frank Rosenblatt deu prosseguimento às idéias de McCulloch e Pitts, criando uma genuína rede de múltiplos neurônios baseada na unidade de limiar linear (HAYKIN, 2001, p. 143–281). Esta rede foi denominada *perceptron* e possui uma topologia em camadas, como mostra a Figura 3.4. Cada círculo denota um neurônio e cada coluna de neurônios representa uma camada. Assim, os neurônios de índice $1, \dots, i, \dots, N$ compõem a camada de entrada; os neurônios de índice $1, \dots, j, \dots, L$ formam a camada oculta; e os neurônios de índice $1, \dots, k, \dots, M$ constituem a camada de saída. A camada de entrada é responsável pela distribuição dos valores de entrada x para os neurônios da camada oculta que, por sua vez, realizam um mapeamento intermediário do problema em questão, gerando vetores linearmente separáveis para a camada de saída, a qual realiza a rotulação das classes ou o mapeamento desejado. A rede neural da Figura 3.4 é formada por uma única camada oculta. Alternativamente, o mapeamento intermediário do problema pode ser realizado por sucessivas camadas ocultas. Todavia, sabe-se hoje em dia que qualquer problema pode ser solucionado por um perceptron de três camadas, com um número suficiente de neurônios na camada oculta (HAYKIN, 2001). O termo $i_j(t)$ é o valor de saída do neurônio genérico j da camada oculta; $y_k(t)$ é o valor

de saída do neurônio genérico k da camada de saída; w_{ji}^o e w_{kj}^s são os pesos sinápticos genéricos da camada oculta e de saída, respectivamente.

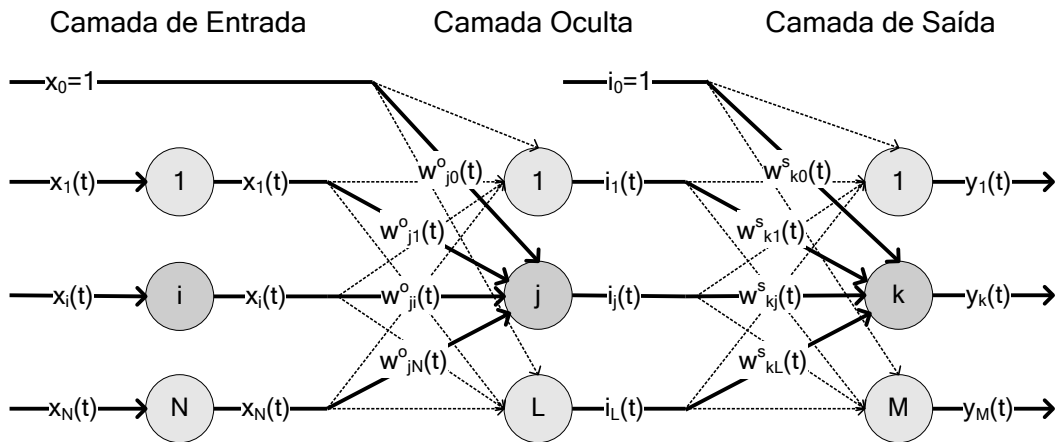


Figura 3.4: Perceptron de múltiplas camadas de Frank Rosenblatt

Nesta mesma época, Bernard Widrow desenvolveu um modelo neural linear intitulado *ADALINE* e, mais tarde, a sua expansão multidimensional *MADALINE* (KOVÁCS, 2002, p. 53–66). Contudo, a maior contribuição de Widrow foi a invenção de um princípio de treinamento extremamente poderoso, conhecido como *regra Delta*, que realiza o ajuste automático dos pesos das conexões entre os neurônios. Posteriormente, esta regra foi generalizada por Paul Werbos que inventou o algoritmo de *retropropagação de erros*. Entretanto, foram os pesquisadores Rummelhart, Hinton e Williams que popularizaram e consagraram o algoritmo de *retropropagação de erros* (KOVÁCS, 2002, p. 67–90).

Haykin (2001, p. 188–202) apresenta o formalismo matemático empregado no algoritmo de retropropagação de erros. O algoritmo parte de uma rede *feedforward* inicializada com pesos aleatórios e então realiza as operações descritas no Algoritmo 3.1, onde d_k é a saída esperada do k -ésimo neurônio da camada de saída, σ_k e σ_j são os níveis de ativação do k -ésimo neurônio da camada de saída e do j -ésimo neurônio da camada oculta, respectivamente, e $f^{s'}(\cdot)$ e $f^{o'}(\cdot)$ são as derivadas da função de ativação da camada de saída e da camada oculta, respectivamente.

Algoritmo 3.1: Algoritmo de Retropropagação de Erros (*Backpropagation*)

```

1 enquanto condição de parada não for satisfeita faça
2   Apresentar um exemplo  $\mathbf{x}$  e computar o valor de saída de cada unidade na rede;
3   para cada unidade de saída  $k$  faça
4     Calcular o termo de erro segundo  $\delta_k^s = f^{s'}(\sigma_k)(d_k - y_k)$ ;
5     Ajustar o vetor de peso sináptico segundo  $\mathbf{w}_k^s = \mathbf{w}_k^s + \eta \delta_k^s \mathbf{i}$ ;
6   fim
7   para cada unidade oculta  $j$  faça
8     Calcular o termo de erro segundo  $\delta_j^o = f^{o'}(\sigma_j) \sum_{k=1}^M \delta_k^s w_{kj}^s$ ;
9     Ajustar o vetor de peso sináptico segundo  $\mathbf{w}_j^o = \mathbf{w}_j^o + \eta \delta_j^o \mathbf{x}$ ;
10  fim
11 fim

```

A condição de parada é satisfeita tipicamente após um número fixo de iterações ou quando o erro torna-se menor que um erro definido como aceitável. Como o algoritmo de retropropagação requer o cálculo do gradiente do vetor de erro, é desejável que a função de ativação dos neurônios seja diferenciável em todos os pontos. Isso explica o sucesso da função sigmóide, uma vez que ela possui esta propriedade.

Desde então inúmeras técnicas de aprendizado foram propostas na literatura (HAYKIN, 2001, p. 75–142), tais como: a rede auto-organizável baseada em grades (*Self-Organizing Map*) proposta por Teuvo Kohonen; as redes auto-organizáveis incrementais da família ART (*Adaptive Resonance Theory*) elaborada por Carpenter e Grossberg; a rede competitiva CPN (*Counterpropagation*) desenvolvida por Robert Nielsen; e a rede recorrente de Hopfield. A Tabela 3.1, a seguir, relaciona alguns modelos de redes neurais com suas aplicações, indicando a principal inspiração biológica/teórica que os influenciaram.

Tabela 3.1: Inspiração biológica das redes neurais. Adaptado de: (BARONE, 2003, p. 120).

Rede neural	Inspiração biológica/teórica	Aplicações
Perceptron	Retina (sistema visual)	Identificação de padrões linearmente separáveis
Rede Backpropagation	Multicamadas	Identificação de padrões não linearmente separáveis, classificação de imagens
Redes auto-organizáveis	Córtex cerebral	Reconhecimento de padrões
Redes da família ART	Teoria da ressonância adaptativa	Reconhecimento de padrões
Rede CPN	Sinapses	Mineração, processamento de imagens
Rede de Hopfield	Redes de atratores	Reconhecimento de padrões, otimização

3.5 A rede neural auto-organizável de Kohonen

Um conjunto de modelos conexionistas vem sendo proposto na literatura com a finalidade de modelar as propriedades peculiares de *mapas corticais* (ou *mapas topográficos*) do cérebro de seres vivos, sendo genericamente conhecidos na literatura por *mapas topológicos auto-organizáveis*. Sabe-se hoje em dia que os mapas corticais têm importância ímpar nos processos de aprendizagem e memória, pois fornecem uma *representação compacta e fiel* dos dados sensoriais. Dependendo da natureza da informação envolvida, um mapa cortical recebe um nome específico – por exemplo, o *mapa retinotópico* faz o mapeamento da retina sobre o córtex visual, o *mapa somatotópico* da pele sobre o córtex somato-sensorio, e o *mapa tonotópico* do ouvido sobre o córtex auditivo.

Em essência, um mapa topológico auto-organizável é definido por um *arranjo* de neurônios representando diferentes elementos processadores ou filtros, que operam em paralelo sobre sinais portadores de informação. Christoph von der Malsburg e David Willshaw foram os pioneiros na construção de mapas topográficos artificiais, propondo em 1976 um modelo computacional para explicar o problema do mapeamento retinotópico (HAYKIN, 2001, p. 484). Inspirado nesta idéia, no início da década de 1980, Teuvo Kohonen também elaborou um modelo que capturava as características essenciais dos variados mapas computacionais do cérebro (KOHONEN, Sept. 1990, 2001). Este modelo, intitulado mapa de Kohonen (ou SOM), é uma rede engenhosa construída em torno de uma grade uni- ou bidimensional que tem o propósito de transformar um sinal de entrada de dimensão arbitrária em um arranjo discreto de unidades de processamento. Essa transformação é realizada de forma adaptativa e mantém relações de similaridade entre os dois

espaços. Por esse motivo, a rede de Kohonen geralmente é utilizada para efetuar análise de agrupamentos e visualização de dados multivariados em dimensões reduzidas.

Considere que \mathcal{X} seja um espaço contínuo de dados de entrada e que possua uma topologia determinada por certas relações métricas entre vetores $\mathbf{x} \in \mathcal{X}$. Do espaço \mathcal{X} conhece-se somente um conjunto finito de vetores (amostras) $\mathbf{x} \in \mathcal{X}$, organizados de acordo com uma densidade de probabilidade $p(\mathbf{x})$. Considere, ainda, que a topologia do espaço de saída \mathcal{A} seja definida por um arranjo geométrico de um conjunto de neurônios $i \in \mathcal{A}$. Desta forma, o mapeamento do espaço de entrada \mathcal{X} para o espaço de saída \mathcal{A} é realizado segundo uma transformação não-linear Φ , tal que:

$$\Phi : \mathcal{X} \rightarrow \mathcal{A} \quad (3.3)$$

Seja m a dimensão do vetor do espaço (contínuo) de entrada \mathcal{X} . Um vetor (padrão) de entrada $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^m$, selecionado de forma aleatória, é denotado por:

$$\mathbf{x} = [x_1, x_2, \dots, x_m]^T \quad (3.4)$$

Cada neurônio da grade possui um vetor de pesos sinápticos de mesma dimensão do vetor de entrada. A representação do vetor de pesos de um neurônio i é dada por:

$$\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{im}]^T, \quad i = 1, 2, \dots, l \quad (3.5)$$

onde l é o número total de neurônios na grade.

A Figura 3.5 representa esquematicamente os elementos básicos da rede de Kohonen.

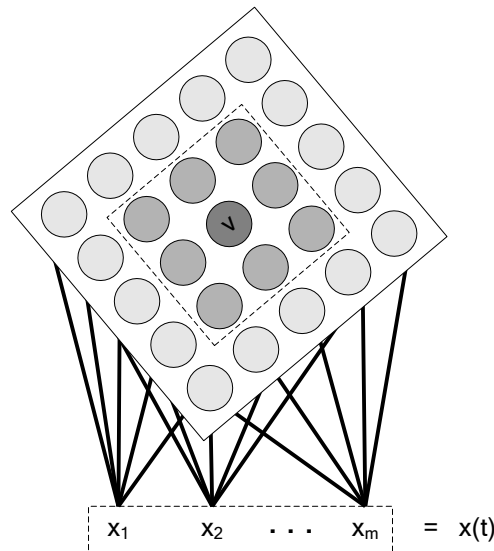


Figura 3.5: Elementos básicos de um mapa auto-organizável de Kohonen. Um padrão de entrada \mathbf{x} de dimensão m é fornecido para a rede no instante t . Dentre todos os l neurônios da rede, um deles é considerado o vencedor v . A vizinhança h_{vi} é dada pelos neurônios mais próximos ao neurônio vencedor. Apenas o vencedor e a sua vizinhança têm os pesos sinápticos modificados, de forma que fiquem ainda mais próximos do padrão de entrada.

De acordo com a Figura 3.5, cada neurônio da grade está totalmente conectado às m unidades de entrada \mathbf{x} , através de *pesos sinápticos* agrupados no vetor \mathbf{w} . Esta rede tem uma estrutura de propagação para frente com uma única camada computacional de

neurônios, normalmente dispostos em uma configuração *linear*, *retangular* ou *hexagonal*. A saída da rede pode ser o índice do neurônio vencedor ou, simplesmente, um vetor binário com todos os bits em zero, exceto o bit que corresponde ao neurônio vencedor. Cabe salientar que a representação gerada pela rede é tipicamente espacial, uma vez que a saída se refere unicamente ao último padrão de entrada apresentado.

O algoritmo responsável pela formação do mapa auto-organizável é de natureza competitiva e não-supervisionada. O primeiro passo realizado pelo algoritmo diz respeito à atribuição de valores pequenos e aleatórios aos pesos sinápticos da rede. Desta forma, nenhuma organização prévia é imposta à rede. Uma vez que os valores iniciais para os parâmetros livres da rede tenham sido apropriadamente escolhidos, há três processos envolvidos na formação do mapa auto-organizável: *competição*, *cooperação* e *adaptação*. O treinamento da rede pode ser resumido através das operações descritas no Algoritmo 3.2. A condição de parada é satisfeita tipicamente após um número fixo de iterações.

Algoritmo 3.2: Algoritmo do Mapa Auto-Organizável de Kohonen

```

1 Inicializar os parâmetros  $\eta$  e  $\sigma$ ;
2 enquanto condição de parada não for satisfeita faça
3   Apresentar um exemplo  $\mathbf{x}$  e computar o valor de saída de cada unidade na rede;
4   Determinar o neurônio vencedor segundo  $v = \operatorname{argmin}_{1 \leq i \leq l} \{\|\mathbf{x} - \mathbf{w}_i\|\}$ ;
5   para cada unidade  $i$  na rede faça
6     Calcular o fator de vizinhança segundo  $h_{vi} = \exp\left(-\frac{\|\mathbf{r}_v - \mathbf{r}_i\|^2}{2\sigma^2}\right)$ ;
7     Ajustar o vetor de peso sináptico segundo  $\mathbf{w}_i = \mathbf{w}_i + \eta h_{vi} [\mathbf{x} - \mathbf{w}_i]$ ;
8   fim
9   Atualizar os parâmetros  $\eta$  e  $\sigma$ ;
10 fim

```

A seguir é discutido em detalhes cada um dos processos envolvidos no algoritmo.

3.5.1 Processo de competição

O processo de competição consiste em identificar o neurônio que melhor represente o vetor de entrada atual. Para tanto, o vetor de pesos sinápticos de todos os neurônios da rede são comparados com o vetor de entrada. Como regra geral, essa comparação é realizada por uma medida de distância. O neurônio cujo vetor de pesos sinápticos mais se aproxima do vetor de entrada é considerado o *vencedor*. Matematicamente, a competição pode ser codificada em termos da distância euclidiana, conforme a Equação 3.6:

$$v(t) = \operatorname{argmin}_{1 \leq i \leq l} \{\|\mathbf{x}(t) - \mathbf{w}_i(t)\|\} \quad (3.6)$$

onde v é o neurônio vencedor para o padrão de entrada \mathbf{x} apresentado à rede no instante de tempo t , dentre todos os l neurônios presentes na rede.

3.5.2 Processo de cooperação

O processo de cooperação consiste em determinar a *vizinhança topológica* centrada no neurônio vencedor. Em geral, a vizinhança é codificada por uma função contínua em forma de “sino”, de acordo com a Equação 3.7:

$$h_{vi}(t) = \exp\left(-\frac{\|\mathbf{r}_v - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right) \quad (3.7)$$

onde \mathbf{r}_v e \mathbf{r}_i são as posições (i.e., coordenada no mapa discreto \mathcal{A}) do neurônio vencedor e dos demais neurônios da rede, respectivamente, e $\sigma(t)$ especifica a “largura” da vizinhança topológica (i.e., determina o número de neurônios que participam no processo de aprendizagem juntamente com o neurônio vencedor).

Através da *função vizinhança* $h_{vi}(t)$, o neurônio vencedor e seus vizinhos no mapa discreto interagem, de tal forma que quanto mais próximo um determinado neurônio estiver do neurônio vencedor mais intensa e positiva será a sua interação. Por esta razão, neurônios próximos tendem a reconhecer características similares e, portanto, pode-se dizer que o mapa de neurônios preserva a topologia do padrão dos vetores de entrada.

O valor do parâmetro σ , via de regra, decresce monotonicamente com o passar do processo de aprendizagem. A diminuição no número de vizinhos é de fundamental importância para o *ordenamento e convergência* da rede. Em geral, a largura deve ser inicialmente alta para promover um rápido ordenamento dos pesos sinápticos e diminuir gradualmente de modo a garantir a convergência dos mesmos. Uma escolha comum para a dependência temporal de $\sigma(t)$ é apresentada na Equação 3.8:

$$\sigma(t) = \sigma_0 \cdot \left(\frac{\sigma_f}{\sigma_0}\right)^{\frac{t}{\tau_1}}, \quad t = 0, 1, \dots, \tau_1 \quad (3.8)$$

em que τ_1 é uma constante de tempo associada ao decaimento do parâmetro σ .

3.5.3 Processo de adaptação

Uma vez determinado o neurônio vencedor e a sua vizinhança topológica, ocorre o processo de adaptação dos pesos sinápticos. Este processo basicamente faz com que o neurônio vencedor e os seus vizinhos se tornem mais especializados no reconhecimento do último padrão de entrada apresentado à rede. Matematicamente, a regra de aprendizagem pode ser calculada a partir da Equação 3.9:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t)h_{vi}(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (3.9)$$

onde $0 < \eta(t) \leq 1$ é o parâmetro da taxa de aprendizagem que controla a velocidade com que os pesos sinápticos são modificados. Dito de outra forma, η especifica a velocidade com que a rede adquire um novo conhecimento.

Assim como no caso da largura da vizinhança topológica, a taxa de aprendizagem deve diminuir com o transcorrer do treinamento de modo a garantir a *convergência* e, principalmente, a *estabilidade* (i.e., manutenção da memória previamente aprendida) do mapa. O parâmetro η , em geral, deve começar com um valor inicial η_0 para $t = t_0$ e, então, decair linearmente no tempo até atingir um valor final η_f , após um número fixo de iterações τ_2 . Este requisito pode ser satisfeito escolhendo-se um decaimento exponencial, de acordo com a Equação 3.10:

$$\eta(t) = \eta_0 \cdot \left(\frac{\eta_f}{\eta_0}\right)^{\frac{t}{\tau_2}}, \quad t = 0, 1, \dots, \tau_2 \quad (3.10)$$

em que τ_2 é uma constante de tempo associada ao decaimento do parâmetro η .

3.5.4 Métricas de precisão e preservação de topologia

Com o intuito de gerar uma configuração *estável e organizada* do mapa discreto de saída \mathcal{A} , deve-se realizar uma seleção criteriosa dos parâmetros σ_0 , σ_f , η_0 e η_f . Devido à ausência de uma fundamentação teórica sólida do algoritmo de Kohonen, na literatura há algumas heurísticas que auxiliam nesta escolha, entretanto, geralmente restritas a um domínio específico de aplicação. De forma geral, duas métricas principais são empregadas na avaliação da *precisão do mapeamento* e da *preservação da topologia*.

A primeira medida tem o propósito de calcular a precisão do mapeamento através de um *erro médio de quantização*, que corresponde à média das distâncias entre cada amostra \mathbf{x} e o correspondente vetor de pesos sinápticos \mathbf{w} do neurônio vencedor v :

$$EMQ = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}^n - \mathbf{w}_v^n\| \quad (3.11)$$

onde N é o número de padrões de treinamento e \mathbf{w}_v^n é o vetor de pesos sinápticos do neurônio vencedor associado ao padrão \mathbf{x}^n .

A segunda medida tem a finalidade de quantificar a capacidade do mapa em representar o arranjo geométrico do espaço de entrada \mathcal{X} . Uma abordagem simples para avaliar o *erro médio topológico* considera, para cada amostra \mathbf{x} , a distância entre o primeiro e o segundo neurônios vencedores:

$$EMT = \frac{1}{N} \sum_{n=1}^N u(\mathbf{x}^n) \quad (3.12)$$

em que $u(\mathbf{x}^n) = 1$ se o primeiro e o segundo vencedores associados ao padrão \mathbf{x}^n não são adjacentes. Caso contrário, $u(\mathbf{x}^n) = 0$. Ou seja, se as unidades não são vizinhas, então a topologia não está sendo preservada.

De forma resumida, a métrica do erro médio de quantização (Eq. 3.11) corresponde à *acuidade* (ou *resolução*) do mapa. Já a métrica do erro topológico (Eq. 3.12) está relacionada com a capacidade da rede em preservar a topologia dos padrões de entrada. Ambos os erros são inversamente proporcionais ao número de neurônios da rede; ou seja, diminuem com o aumento de neurônios. No entanto, quando existe um número elevado de neurônios pode ocorrer um fenômeno chamado sobre-ajuste (*overfitting*), tal que a rede perde sua capacidade de generalização. Por outro lado, quando há um número reduzido de neurônios, pode acontecer o fenômeno inverso denominado sub-ajuste (*underfitting*), tal que a rede se torna rígida demais para lidar com variados padrões de entrada.

As duas métricas supracitadas podem ser combinadas, de modo a formar uma medida única da qualidade do mapeamento aprendido, tal qual foi proposto por Kaski e Lagus em 1996. Apesar de promissora, essa métrica híbrida é computacionalmente onerosa, podendo tornar-se proibitiva em arranjos com dimensão maior que 2.

3.5.5 Propriedades do mapa de características

Uma vez concluído o processo de aprendizagem da rede, o mapa discreto de saída \mathcal{A} apresenta características importantes do espaço de entrada \mathcal{X} . De acordo com (HAYKIN, 2001, p. 494–502), pode-se enumerar as seguintes propriedades do mapa de características Φ (vide Eq. 3.3) resultantes da aplicação do algoritmo de Kohonen:

Propriedade 1. Aproximação do espaço de entrada. O mapeamento Φ fornece uma aproximação discreta do espaço de entrada \mathcal{X} . Cada neurônio da grade é responsável

por uma região do espaço de entrada. A *região de influência* (ou, *campo receptivo*) de um neurônio pode indicar que todos os padrões de treinamento que ativam tal neurônio compõem um *grupo* ou uma *categoria*. Assim, é correto afirmar que uma propriedade intrínseca do algoritmo é a *categorização* dos padrões de entrada.

Propriedade 2. Ordenação topológica. O mapeamento Φ é *ordenado de forma topológica*, no sentido de que a localização espacial de um neurônio na grade corresponde a uma característica particular dos padrões de entrada.

Propriedade 3. Casamento de densidade. O mapeamento Φ reflete variações na estatística da distribuição da entrada. Regiões no espaço de entrada \mathcal{X} de onde as amostras x sejam retiradas com uma alta probabilidade de ocorrência são mapeadas em regiões maiores do espaço de saída \mathcal{A} , e vice-versa.

Propriedade 4. Seleção de características. O mapeamento Φ fornece uma aproximação discreta das *curvas-principais*, e pode, por conseguinte, ser visto como uma generalização não-linear da *análise de componentes principais*¹ (JOLLIFFE, 2002).

Propriedade 5. Representação visual. O mapeamento Φ pode ser visualizado de quatro formas: representação do vetor de pesos no espaço \mathbb{R}^m ; *matriz de distância unificada* (matriz-U) que permite analisar as relações topológicas entre os neurônios por meio de uma imagem; *planos de componentes* que utiliza um método de coloração para analisar como cada vetor de entrada x foi organizado na grade; e atribuição de *rótulos de classe* a cada neurônio, tal que a grade seja particionada em um número de *regiões coerentes* e forme um *mapa contextual* (ou, *mapa semântico*).

3.6 Considerações do capítulo

Este capítulo apresentou um panorama geral da área de redes neurais artificiais, introduzindo os principais conceitos, propriedades e componentes de uma rede neural. As redes neurais representam um *tema multidisciplinar* com raízes na neurociência, matemática, estatística, física, ciência da computação e engenharia. A sua habilidade para *aprender* a partir de exemplos as tornou ferramentas inestimáveis em aplicações tão diversas como modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinais e controle. Em particular, as redes neurais têm muito a oferecer quando a solução de um problema de interesse é dificultada por alguns dos seguintes pontos:

1. Falta de conhecimento físico/estatístico do problema;
2. Variações estatísticas nos dados observáveis; e
3. Mecanismo não-linear responsável pela geração dos dados.

De forma resumida, uma rede neural artificial compreende um sistema não-linear do tipo *caixa-preta*, pois fornece uma estrutura com parâmetros ajustáveis, sendo capaz de implementar uma ampla classe de modelos (ENGEL, 2001). Os modelos que podem ser implementados dependem essencialmente da arquitetura específica da rede considerada.

¹A análise de componentes principais é uma técnica estatística poderosa utilizada para reduzir a dimensionalidade dos dados com um grande número de variáveis correlacionadas e para fornecer uma visão estatisticamente privilegiada do conjunto de dados original.

Por exemplo, a rede neural artificial mais simples, constituída por um único neurônio artificial, também chamada de *perceptron elementar*, equivale a um *discriminador linear*. Por isso, o perceptron elementar é capaz de implementar apenas modelos *linearmente separáveis*, o que restringe a sua aplicação a problemas bastante simples. Por outro lado, uma arquitetura onde existem vários neurônios dispostos paralelamente em camadas, com ligações adiante (*feedforward*) entre as camadas, equivale a um *aproximador universal*.

Em especial, a rede neural de Kohonen constitui uma classe de sistemas auto-organizáveis, cujo algoritmo é inspirado na neurobiologia², incorporando os mecanismos que são básicos para a auto-organização: *competição*, *cooperação* e *auto-amplificação*. A sua principal funcionalidade é construir um mapeamento de um espaço de dimensão elevada para um arranjo discreto de neurônios de forma a conservar a topologia do espaço de entrada. Atualmente, a rede de Kohonen é comumente empregada nos mais diversos domínios de aplicação como, por exemplo, no reconhecimento e classificação de padrões em geral, identificação e controle de processos e na área da robótica.

Por ser uma rede competitiva, a rede de Kohonen serve naturalmente para categorizar os vetores de entrada, sendo capaz de extrair relações complexas e não-lineares existentes entre dados estáticos, ou seja, a rede é capaz de realizar *quantização vetorial*. A quantização vetorial pode ser compreendida como um procedimento que divide o espaço de entrada em um número de subespaços disjuntos, de tal forma que cada vetor de entrada x é representado pelo rótulo do subespaço ao qual ele pertence. Em termos técnicos, a quantização vetorial é uma estimativa pontual da *função densidade de probabilidade* dos dados de entrada, codificada pelo vetor de pesos sinápticos dos neurônios da rede. Portanto, a densidade de neurônios é máxima nas áreas em que entradas aparecem com maior probabilidade e mínima em regiões onde as entradas são escassas. A quantização vetorial oferece três vantagens principais às redes neurais artificiais:

- *Compressão de dados*, uma vez que, após a estabilização da rede com um limiar de erro aceitável, seus pesos podem ser utilizados em vez dos próprios dados;
- *Redução considerável de esforço computacional*, pois a quantidade de vetores de pesos (protótipos) é bastante reduzida em relação à quantidade de dados; e
- *Aumento da robustez*, visto que os protótipos extraem qualidades estatísticas médias, filtrando flutuações aleatórias que porventura estejam presentes nos dados.

Na rede de Kohonen os neurônios estão dispostos conforme um arranjo predeterminado uni- ou bidimensional, que é mantido durante todo o processo de treinamento. Na maioria das vezes, pouca ou nenhuma informação está disponível acerca do tamanho e da distribuição do conjunto de amostras do espaço de entrada \mathcal{X} . Portanto, a definição de uma arquitetura de rede apropriada é um fator que depende, em grande parte, da experiência empírica do projetista. Além disso, as amostras podem estar correlacionadas no tempo, ou seja, a ordem em que são observadas no tempo tem um papel fundamental e deve ser levada em conta pelo modelo neural. No entanto, a rede de Kohonen convencional somente é capaz de processar dados que estão relacionados de forma espacial.

Com o propósito de resolver as duas limitações supracitadas, inúmeras derivações e variações da rede de Kohonen foram propostas na literatura especializada da área. Espera-se que esta introdução sobre os fundamentos de redes neurais artificiais possibilite a compreensão dos formalismos e das técnicas de aprendizagem empregadas em redes auto-organizáveis temporais e construtivas, tema a ser explorado no próximo capítulo.

²A plausibilidade neurobiológica do mapa de Kohonen é discutida em (KOHONEN, 1993).

4 TRABALHOS CORRELATOS

Este capítulo busca oferecer uma visão geral das pesquisas relacionadas ao tema desta dissertação. Em um primeiro momento, trabalhos que tratam da questão temporal em redes neurais auto-organizáveis são discutidos. Logo após, é feito um apanhado sobre redes neurais auto-organizáveis construtivas. Como não existem métricas rígidas para se avaliar em termos quantitativos cada modelo, a maioria foi implementada no simulador robótico desenvolvido e testada em casos simples, considerando tarefas de segmentação temporal. O objetivo principal da análise foi o de obter uma idéia das características e peculiaridades de cada modelo e, ao mesmo tempo, fornecer parâmetros sobre que limites precisam ser superados. O resultado desta análise está sintetizado nas Tabelas 4.1 e 4.2.

4.1 As redes neurais auto-organizáveis temporais

A proposta original da rede de Kohonen é baseada no processamento de dados *estáticos*. Em outras palavras, dados que estão relacionados apenas de forma *espacial*. Contudo, também pode existir uma correlação *temporal* entre eles. Este é o caso de padrões que ocorrem sequencialmente no tempo, tal que um padrão individual não tem significado algum a não ser que seja considerado em conjunto com aqueles que o precedem e/ou seguem. Para tais padrões, doravante denominados indistintamente de *dinâmicos*, *temporais* ou *seqüenciais*, a ordem em que eles são observados no tempo tem um papel fundamental e deve ser levada em conta pelo modelo neural que os processa¹.

Desta forma, é de suma importância investigar como incorporar o tempo na operação de uma rede neural, pois muitas das tarefas desempenhadas por seres humanos e animais envolvem tomadas de decisão e respostas comportamentais a estímulos espaço-temporais, como, por exemplo, a visão, a fala, o processamento de sinais e o controle motor. A capacidade de processar padrões temporais deve ser uma propriedade das mais relevantes de um sistema inteligente, seja ele biológico ou artificial.

A grande maioria das redes neurais para processar informação temporal é baseada no aprendizado supervisionado, entre elas o perceptron de múltiplas camadas, redes de base radial e redes baseadas em memória associativa (MOZER, 1993; HAYKIN, 2001, p. 685–847). Apesar de existirem redes não-supervisionadas projetadas para lidar com seqüências temporais desde o final dos anos 60, apenas recentemente é que elas têm recebido atenção especial por parte da comunidade científica. Em (BARRETO; ARAÚJO, 2001; MOSER, 2004, p. 19–57) é apresentada uma revisão bibliográfica do estado da arte em modelos de representação temporal baseados na arquitetura da rede de Kohonen.

¹Para uma discussão sobre o papel do tempo no processamento neural, consulte (ELMAN, 1990).

4.1.1 Processamento de seqüências temporais: definições e propriedades

De acordo com (BARRETO; ARAÚJO, 2001, p. 4), um padrão dinâmico é uma série ou *seqüência espaço-temporal* de natureza *determinística*, *estocástica*, ou ainda *caótica*, formada por um conjunto finito e ordenado no tempo de N vetores de características:

$$X_N = \{\mathbf{x}(t), \mathbf{x}(t-1), \mathbf{x}(t-2), \dots, \mathbf{x}(t-N)\} \quad (4.1)$$

onde cada *componente* é denotado por $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \subset \mathbb{R}^n$. Caso existam várias seqüências, comumente os itens de uma seqüência particular são identificados por meio de um sobrescrito, por exemplo, μ , $\mu = 1, \dots, P$.

Para fins de simulação em um computador digital é útil adotar-se uma versão discreta da dimensão temporal, tal que o tempo evolua em intervalos regulares Δt . Desta forma, o símbolo t é usado para representar um instante particular no tempo, onde $t \in \{0, \Delta t, 2\Delta t, 3\Delta t, \dots\}$. Em geral, a unidade de medida Δt é expressada como um elemento do conjunto de números inteiros: $t \in \{0, 1, 2, 3, \dots\}$

O processamento de seqüências temporais é uma tarefa desafiadora para a qual ainda não há uma abordagem unificada (BARRETO; ARAÚJO, 2001, p. 5). Isto se deve em parte à diversidade de propriedades dos sinais temporais. O conhecimento dessas propriedades é útil para entender as aplicações envolvidas em redes neurais temporais. As seguintes definições têm caráter geral e se aplicam igualmente a qualquer seqüência (BARRETO; ARAÚJO, 2001, p. 5):

1. *Ordem temporal*: posição relativa de cada item da seqüência em relação a outro;
2. *Métrica*: duração de um padrão no tempo em relação a outros padrões da seqüência;
3. *Densidade*: número de vetores de características que compõem uma seqüência;
4. *Assimetria*: sentido do tempo que normalmente não pode ser invertido;
5. *Contexto temporal*: menor subseqüência formada por itens passados que possibilitam determinar sem ambigüidades o item atual da seqüência; e
6. *Grau*: número de itens que compõem o contexto temporal.

As propriedades supracitadas são usualmente codificadas como parâmetros do modelo. Se, porventura, as propriedades não variarem com o tempo, então diz-se que o sinal é *estacionário* ou *invariante no tempo*. No entanto, muitos sinais do mundo real são gerados a partir de sistemas cujos parâmetros variam significativamente com o tempo e, por isso, são denominados de *não-estacionários* ou *variantes no tempo*.

4.1.2 Memórias de curto prazo

Para estabelecer associações e extrair correlações temporais entre padrões consecutivos de uma dada seqüência, uma rede neural deve ser capaz de reter e manter provisoriamente informação sobre itens passados desta seqüência. O mecanismo de retenção, usualmente denominado *memória de curto prazo* (STM, do inglês *short-term memory*), é amplamente utilizado no processamento de seqüências temporais².

²Para uma revisão das estruturas de memória de curto prazo e seu papel no processamento temporal, consulte (MOZER, 1993), (HAYKIN, 2001, p. 685–690) e (BARRETO; ARAÚJO, 2001).

Basicamente, memórias de curto prazo convertem informação temporal em espacial, possibilitando que redes estáticas convencionais sejam empregadas em tarefas de processamento de informação temporal. Esta abordagem para construir um sistema dinâmico não-linear é direta porque fornece uma clara separação de responsabilidades: a rede estática é responsável pela não-linearidade, e a memória é responsável pelo tempo.

Uma memória de curto prazo não possui nenhum tipo de aprendizado e, por essa razão, o seu comportamento é exclusivamente determinado por seus parâmetros de funcionamento. Para relacionar o comportamento dos diferentes tipos de memórias de curto prazo, Mozer (1993) introduziu um modelo matemático que generaliza a computação realizada por uma memória, tal que:

$$\mathbf{y}(t) = \sum_{\tau=1}^t c(t - \tau)\mathbf{x}(\tau) \quad (4.2)$$

onde t se refere a tempo decorrido e \mathbf{y} representa a convolução dos valores de entrada $\mathbf{x}(t)$ em cada instante de tempo por pesos de participação $c(t)$. Os pesos de participação $c(t)$ caracterizam o comportamento da memória e são conhecidos como função *kernel*³.

Nas subseções seguintes serão apresentadas e examinadas as principais memórias de curto prazo relatadas na literatura especializada da área (MOZER, 1993; HAYKIN, 2001; BARRETO; ARAÚJO, 2001), a saber: *memória de atraso*, *memória de traço exponencial baseada em integradores* (do inglês *leaky integrators*), *memória gama*, *memória diferenciadora-integradora* e *unidade de habituação*.

4.1.2.1 Memória de atraso

A memória de atraso consiste na forma mais simples de representar a informação temporal. Basicamente, a memória armazena o valor do k -ésimo instante de tempo passado do vetor \mathbf{x} . O valor de k especifica o atraso da memória, ou seja, de quanto tempo o sinal de saída da memória está defasado em relação ao sinal de entrada. A representação gráfica de uma memória de atraso está ilustrada na Figura 4.1.

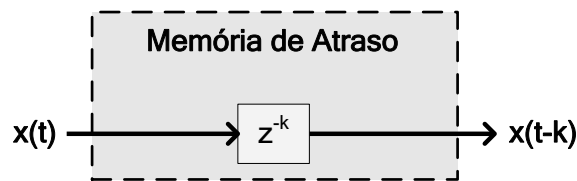


Figura 4.1: Representação gráfica de uma memória de atraso k .

Para representar mais de uma informação passada, a idéia é *encadear* as memórias de atraso, de tal forma que sejam formadas *linhas de atraso* (do inglês *tapped delay line*). O vetor de entrada $\mathbf{x}(t)$ a ser apresentado à rede neural é substituído por uma janela de tempo de tamanho $\tau \geq 0$ que abrange o item atual da seqüência e seus τ predecessores:

$$\bar{\mathbf{x}}(t) = [\mathbf{x}(t), \mathbf{x}(t - 1), \mathbf{x}(t - 2), \dots, \mathbf{x}(t - \tau + 1), \mathbf{x}(t - \tau)]^T \quad (4.3)$$

tal que $\dim(\bar{\mathbf{x}}(t)) = (\tau + 1) \cdot \dim(\mathbf{x}(t))$.

A linha de atraso representa de forma explícita e sem perda de informação os últimos τ instantes de tempo da seqüência temporal. Contudo, todos os instantes anteriores não

³Em técnicas digitais a função *kernel* é definida como a resposta a um impulso unitário do sistema.

são representados. Assim, a informação gerada sofre um truncamento abrupto. Deve-se ter cuidado, portanto, na escolha do tamanho da linha de atraso utilizada. Se for menor que o tamanho mínimo do padrão temporal mais simples, a rede não terá como identificá-lo sem cometer *ambigüidades*. Por outro lado, se for muito maior, a rede terá dificuldades em criar *generalizações*. Em geral, o tamanho τ adequado da janela é estipulado por tentativa-e-erro, após alguma experimentação inicial com os dados e a rede.

A memória de atraso mantém o conteúdo exato do k -ésimo sinal de entrada e, por essa razão, a sua função *kernel* $c(t)$ é igual a 1 no instante de tempo k e para todos os demais valores de t é igual a zero:

$$c(t) = \begin{cases} 1 & \text{se } t = k \\ 0 & \text{caso contrário} \end{cases} \quad (4.4)$$

A Figura 4.2 ilustra o comportamento da memória de atraso unitário, onde $k = 1$.

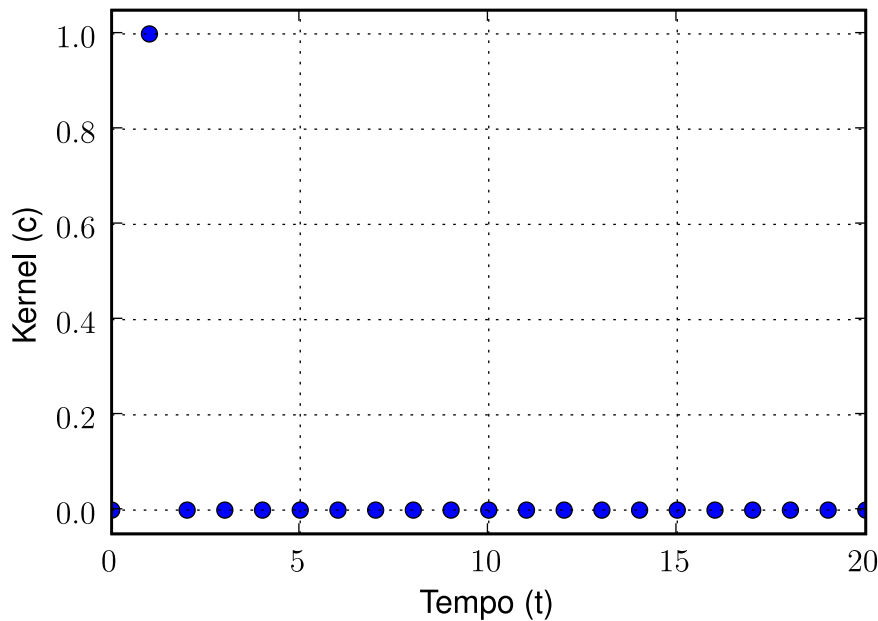


Figura 4.2: Função *kernel* da memória de atraso unitário ($k = 1$).

4.1.2.2 Memória de traço exponencial

A memória de traço exponencial consiste na utilização de integradores. Esta técnica agrega ao vetor de entrada a atividade da rede no instante anterior, conforme:

$$\bar{x}(t) = (1 - \lambda)\bar{x}(t - 1) + \lambda x(t) \quad (4.5)$$

onde $0 < \lambda \leq 1$ é a constante de integração que especifica o grau de influência das entradas anteriores sobre a entrada atual. Em geral, este parâmetro também é definido por tentativa-e-erro. Considera-se a condição inicial $\bar{x}(0) = 0$.

O mecanismo de integração da memória de traço exponencial não perde informação de forma abrupta a um dado ponto no tempo. Ao invés disso, a informação degrada suavemente, de acordo com o parâmetro λ . A representação gerada pela memória de traço exponencial é determinada predominantemente pelas entradas mais recentes. Logo, também há um tamanho máximo de padrão temporal que pode ser distinguido sem ambigüidades.

No entanto, a definição deste tamanho é implícita, realizada através da escolha da taxa de integração λ . A Figura 4.3 representa graficamente a memória de traço exponencial.

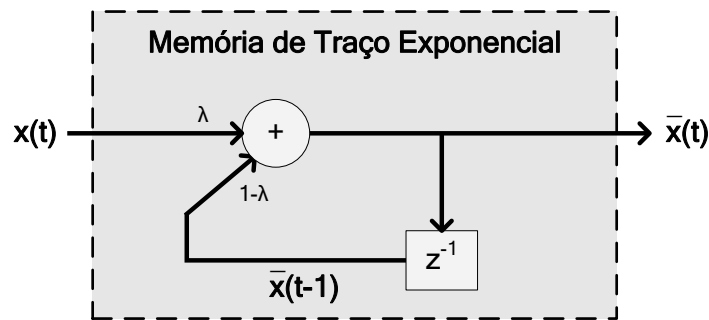


Figura 4.3: Representação gráfica de uma memória de traço exponencial.

A memória de traço exponencial realiza uma integração temporal sobre os dados de entrada, de tal forma que a informação armazenada decai com a passagem do tempo. Esse comportamento é melhor compreendido através da função *kernel* $c(t)$ apresentada na Equação 4.6 e ilustrada na Figura 4.4 para os parâmetros $\lambda = 0.6, 0.7, 0.8, \text{ e } 0.9$.

$$c(t) = (1 - \lambda)\lambda^t \quad (4.6)$$

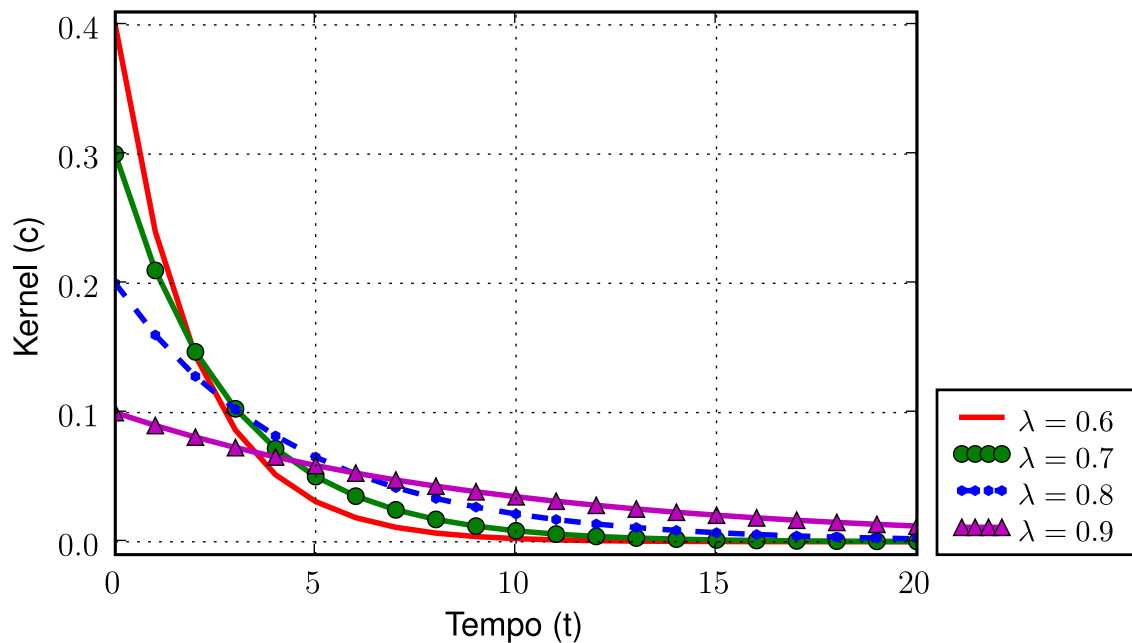


Figura 4.4: Função *kernel* da memória de traço exponencial para $\lambda = 0.6, 0.7, 0.8, \text{ e } 0.9$.

4.1.2.3 Memória Gama

A memória Gama fornece uma solução de memória robusta e parametrizável, que generaliza as memórias de curto prazo de linhas de atraso e de traço exponencial. Na representação gráfica da Figura 4.5 pode-se observar *unidades de memória exponencial encadeadas*, formando *linhas de atraso*. As memórias exponenciais são governadas pelo parâmetro λ , e a quantidade de memórias empregadas é definida pelo parâmetro τ .

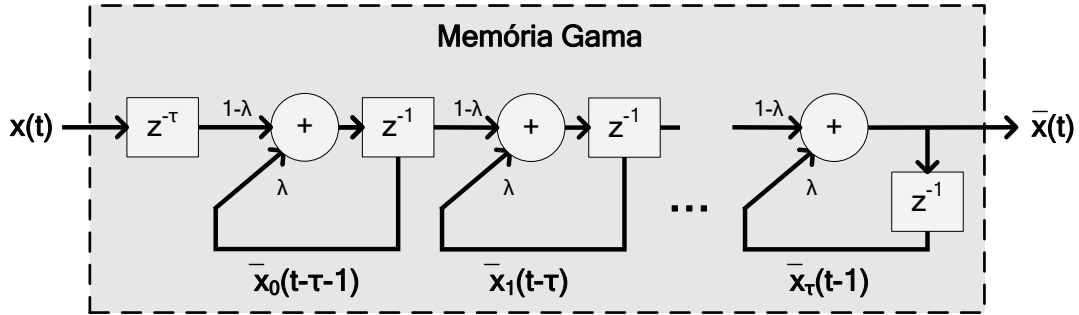


Figura 4.5: Representação gráfica da memória Gama.

A Equação 4.7 define o cálculo de atualização da memória Gama. Para calcular o valor de saída de uma memória $\bar{x}_\tau(t)$ é necessário calcular também $\bar{x}_j(t)$ para $j = 0, 1, 2, 3, \dots, \tau - 1$. Consideram-se as seguintes condições iniciais: $\bar{x}_{-1}(t) = x(t + 1)$ para todo $t \geq 0$, e $\bar{x}_j(0) = 0$ para todo $j \geq 0$.

$$\bar{x}_j(t) = (1 - \lambda)\bar{x}_{j-1}(t - 1) + \lambda\bar{x}_j(t - 1) \quad (4.7)$$

A função *kernel* $c(t)$ da memória Gama é estabelecida na Equação 4.8 e representada na Figura 4.6 para os parâmetros $\tau = 6$ e $\lambda = 0.3, 0.5$ e, 0.7 .

$$c(t) = \begin{cases} \binom{t}{\tau} (1 - \lambda)^{\tau+1} \lambda^{t-\tau} & \text{se } t \geq \tau \\ 0 & \text{caso contrário} \end{cases} \quad (4.8)$$

Como pode-se observar na Figura 4.6, a memória Gama agrega a funcionalidade das memórias de atraso e das memórias de traço exponencial.

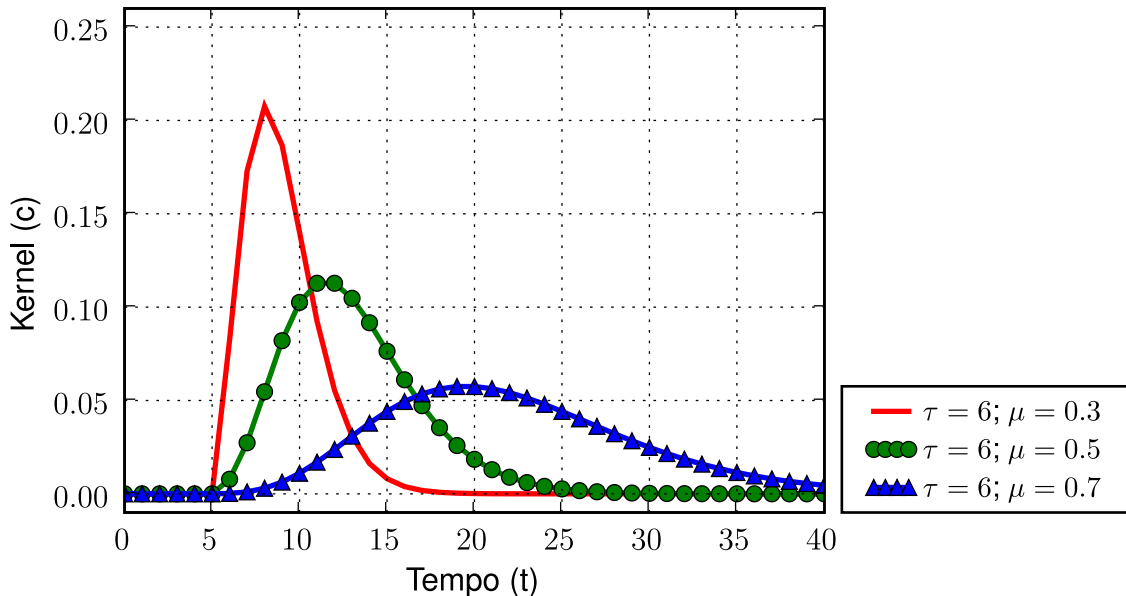


Figura 4.6: Função *kernel* da memória Gama para $\tau = 6$ e $\lambda = 0.3, 0.5$, e, 0.7 .

4.1.2.4 Memória diferenciadora-integradora

A memória diferenciadora-integradora proposta por Moser (2004) tem como finalidade principal extrair informações temporais implícitas do sinal de entrada original $x(t)$,

realizando uma *derivação* e/ou *integração* da informação. Matematicamente, tem-se as seguintes equações que governam o funcionamento da memória:

$$\bar{x}(t) = (1 - \mu)x(t) + \mu\bar{x}(t - 1) \quad (4.9)$$

$$dx(t) = x(t) - \bar{x}(t - 1) \quad (4.10)$$

$$\overline{dx}(t) = (1 - \eta)dx(t) + \eta\overline{dx}(t - 1) \quad (4.11)$$

onde $x(t)$ é o vetor de entrada, $\bar{x}(t)$ representa a média móvel do sinal de entrada com constante de decaimento μ , $dx(t)$ representa a diferença do sinal de entrada por sua média móvel, e $\overline{dx}(t)$ é a saída da memória no instante t e representa a média móvel do sinal $dx(t)$ com constante de decaimento η . As constantes de integração μ e η devem ter o valor entre 0 e 1. Consideram-se as condições iniciais $\bar{x}(0) = 0$ e $\overline{dx}(0) = 0$. A Figura 4.7 representa graficamente a memória diferenciadora-integradora.

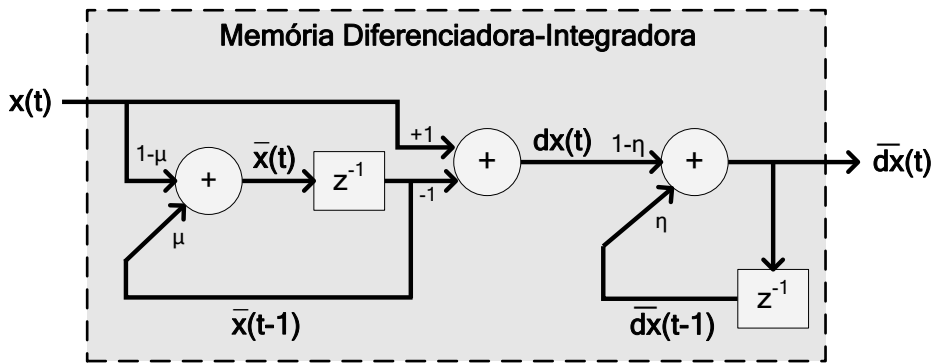


Figura 4.7: Representação gráfica da memória diferenciadora-integradora.

A função *kernel* $c(t)$ da memória diferenciadora-integradora é definida na Equação 4.12 e representada graficamente na Figura 4.8:

$$c(t) = (1 - \eta)\eta^t - (1 - \eta)(1 - \mu) \sum_{j=0}^{t-1} (\eta^{t-1-j} \mu^j) \quad (4.12)$$

Pela Figura 4.8 pode-se perceber que a função *kernel* da memória diferenciadora-integradora tem um *comportamento padrão* independente dos parâmetros μ e η (MOSER, 2004, p. 63–65). Para os casos em que $\mu < 1$, a função *kernel* computa a diferença dos sinais mais recentes pelos sinais que os antecedem, reproduzindo o cálculo da *diferenciação*. Para os casos em que $\mu = 1$, a função *kernel* é sempre positiva e tende a zero em uma taxa exponencial, refletindo o comportamento da *integração*. Outra observação importante a respeito da função *kernel* é relativa à sua área total. Com exceção de $\mu = 1$, a integral da função *kernel* é sempre zero (i.e., a região positiva tem a mesma área que a região negativa). Esta propriedade faz com que se um sinal constante for aplicado à memória, a sua saída tenderá a zero.

A seguir faz-se uma análise sobre a influência dos parâmetros μ e η no funcionamento da memória diferenciadora-integradora:

- Para $\mu < 1$ e $\eta = 0$, a Equação 4.11 é reduzida ao termo $x(t) - \bar{x}(t - 1)$, que corresponde a uma subtração do sinal de entrada $x(t)$ por sua integração $\bar{x}(t - 1)$.
- Para $\mu = 1$ e $\eta < 1$, a Equação 4.11 é reduzida ao termo $(1 - \eta)x(t) + \eta\overline{dx}(t - 1)$, que consiste em uma integração do sinal de entrada.

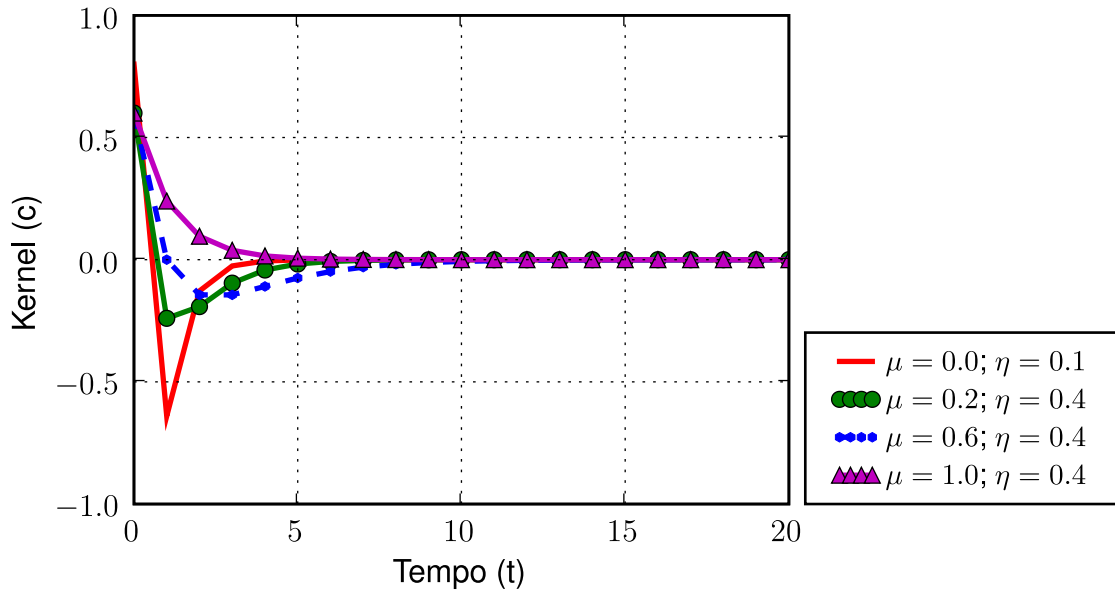


Figura 4.8: Função *kernel* da memória diferenciadora-integradora para $\mu = 0.0$ e $\eta = 0.1$; $\mu = 0.2$ e $\eta = 0.4$; $\mu = 0.6$ e $\eta = 0.4$; $\mu = 1.0$ e $\eta = 0.4$.

- Para μ e η entre 0 e 1, a memória funciona como um diferenciador e um integrador.
- Para $\mu = 0$ e $\eta = 0$, a Equação 4.11 é reduzida ao termo $\mathbf{x}(t) - \mathbf{x}(t - 1)$, que é a *diferenciação de dois pontos adjacentes*.
- Para $\eta = 1$, as integrações $\bar{\mathbf{x}}(t)$ e $\overline{d\mathbf{x}}(t)$ são anuladas, resultando nas equações $\overline{d\mathbf{x}}(t) = 0$ e $\bar{\mathbf{x}}(t) = 0$ para todo t .

4.1.2.5 Unidade de habituação

A habituação é uma forma simples de aprendizado tipicamente observada em organismos biológicos. Do ponto de vista da neurociência, a habituação ocorre através de mudanças estruturais nas conexões sinápticas entre os neurônios, de tal forma que a resposta elétrica a estímulos repetitivos e irrelevantes seja paulatinamente reduzida até ser nula. Ou seja, a habituação funciona como um *filtro de novidade* para o organismo.

O fenômeno da habituação já foi utilizado como fonte de inspiração em inúmeros modelos conexionistas como uma forma de representação temporal. DeLiang Wang e Michael Arbib (BRISTOL; PURCELL; CAREW, 2003, p. 504–507) propuseram um modelo computacional da habituação que opera da seguinte forma:

$$y(t) = y(t - 1) + \tau(\alpha z(t)(y(0) - y(t - 1)) - \beta y(t - 1)x(t)) \quad (4.13)$$

$$z(t) = z(t - 1) + \gamma z(t - 1)(z(t - 1) - 1)x(t) \quad (4.14)$$

onde $x(t)$ é o sinal de entrada, α determina a taxa de recuperação da habituação, β controla a velocidade de habituação, τ é uma constante de tempo que governa a variação da taxa de habituação e $z(t)$ regula a recuperação da habituação. Inicialmente, quando $z(t)$ é relativamente grande, a recuperação é rápida. Em contraste, quando $z(t)$ é pequeno, a recuperação é lenta. Ou seja, à medida que o processo de habituação se desenvolve, a velocidade de recuperação da habituação se torna cada vez mais lenta, produzindo um efeito de habituação mais duradouro (i.e., a longo prazo).

A representação gráfica da unidade de habituação é ilustrada na Figura 4.9.

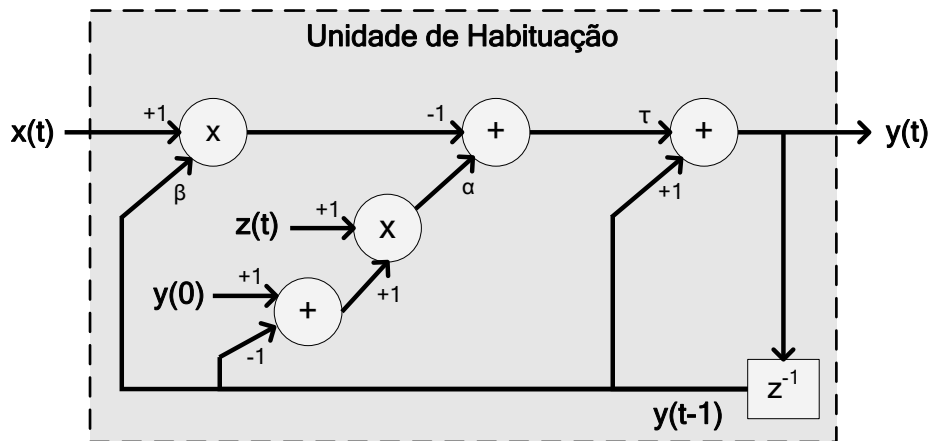


Figura 4.9: Representação gráfica da unidade de habituação

O comportamento da unidade de habituação é parametrizável, através de α , β e γ . A seguir faz-se uma análise sobre a influência dos parâmetros no funcionamento da unidade:

- Para $\alpha = 0$ e $\beta \leq 1$, a Equação 4.13 é reduzida ao termo $y(t - 1) - \gamma(\beta y(t - 1)x(t))$, fator de habituação, o que faz com que a resposta da unidade diminua gradativamente do estado de não habituação até o seu ponto máximo de saturação.
- Para $\alpha \leq 1$ e $\beta = 0$, a Equação 4.13 é reduzida ao termo $y(t - 1) + \gamma(\alpha z(t)(y(0) - y(t - 1)))$, fator de recuperação, o que faz com que a resposta da unidade aumente sucessivamente de um estado qualquer até o ponto inicial de não habituação.
- Para α e β entre 0 e 1, a unidade é controlada por dois processos independentes e conflitantes de habituação e sensibilização⁴. Se a estimulação $x(t)$ for igual a 1, a saída da unidade rapidamente chega ao seu ponto máximo de saturação (valor zero). No momento que o sinal de entrada voltar a zero, a unidade lentamente volta ao seu estado inicial de não habituação ($y(0)$).
- Para $\alpha = 0$ e $\beta = 0$, os fatores de recuperação e habituação são anulados, resultando na equação $y(t) = y(t - 1)$ que corresponde ao estado inicial de não habituação.

A unidade de habituação não pode ser expressa com a convolução por alguma função *kernel*, haja vista que o sinal de entrada $x(t)$ não é multiplicado por pesos constantes $c(t)$, mas pelo valor da própria memória do instante anterior $y(t - 1)$ (vide Equação 4.13).

4.1.2.6 Discussão

As memórias de curto prazo podem ser analisadas segundo os critérios de *profundidade* e *resolução* (MOZER, 1993, p. 5). A profundidade indica o quão distante no tempo a memória pode representar uma informação. A resolução determina a qualidade da informação armazenada para cada instante de tempo. A memória de atraso unitário tem pequena profundidade e alta resolução, uma vez que armazena um único valor do passado sem perda de informação. A memória de traço exponencial tem grande profundidade e pequena resolução, pois representam teoricamente todos os dados do passado mas com degradação exponencial na qualidade da informação.

⁴Na unidade de habituação, a sensibilização tem como propósito sobrepor-se aos efeitos da habituação, caracterizando um processo denominado *desabituação* (BRISTOL; PURCELL; CAREW, 2003, p. 504–507).

A memória Gama generaliza o comportamento das memórias de atraso e das memórias exponenciais, no entanto, com um elevado custo computacional atrelado e uma quantidade bem maior de parâmetros que devem ser ajustados. A memória diferenciadora-integradora realiza uma integração e/ou diferenciação da informação temporal, eliminando toda a redundância e a correlação espaço-temporal do sinal de entrada. A unidade de habituação tem um comportamento similar ao da memória de traço exponencial, entretanto, a integração da informação temporal realizada tem um viés para acumulação do sinal de entrada até seu ponto de saturação (valor zero).

A Figura 4.10 apresenta uma comparação entre as memórias de curto prazo discutidas. Como pode ser observado, as seis memórias geram ambigüidades na representação, entretanto, em instantes de tempo distintos (e.g., instantes 40 e 60 na Figura 4.10d). Provavelmente, a solução mais robusta seja obtida pela utilização adequada de diferentes tipos de memórias, cada uma ressaltando uma propriedade temporal distinta.

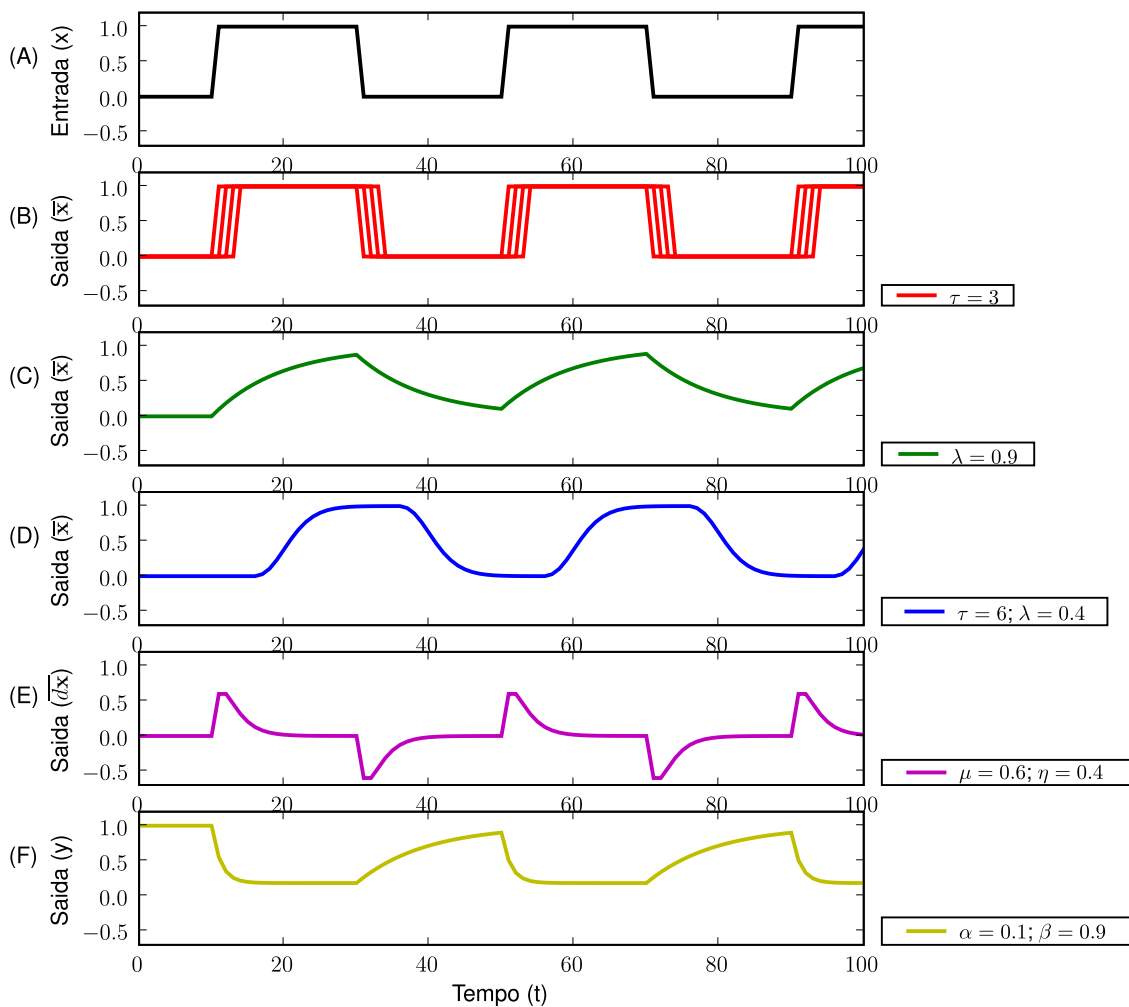


Figura 4.10: (A) Sinal temporal de entrada $x(t)$ aplicado às memórias de curto prazo. (B) Resposta de uma memória de linhas de atraso de parâmetro $\tau = 3$. (C) Resposta de uma memória de traço exponencial de parâmetro $\lambda = 0.9$. (D) Resposta de uma memória gama de parâmetros $\tau = 6$ e $\lambda = 0.4$. (E) Resposta de uma memória diferenciadora-integradora de parâmetros $\mu = 0.6$ e $\eta = 0.4$. (F) Resposta de uma unidade de habituação de parâmetros $\alpha = 0.1$ e $\beta = 0.9$.

Nessa discussão sobre memórias de curto prazo torna-se relevante enfatizar a dife-

rença entre treinar uma rede de Kohonen com entrada estática e com uma entrada dinâmica. No primeiro caso há um particionamento do conjunto de treinamento em categorias conforme as similaridades espaciais, enquanto que no segundo existe uma separação por uma combinação da semelhança espacial e da proximidade temporal. As redes de Kohonen temporais aplicam as técnicas de memória de curto prazo de variadas formas, conforme será descrito nas subseções seguintes.

4.1.3 Time-Delayed Self-Organizing Map (TDSOM)

O *Mapa Auto-Organizável de Linhas de Atraso* (KANGAS, 1990) é uma rede de Kohonen convencional que utiliza memórias de curto prazo *externamente* à rede. Ou seja, as memórias são aplicadas aos itens da seqüência e não à rede propriamente dita, que permanece estática. O vetor de entrada original $\mathbf{x}(t)$ simplesmente é substituído por uma de suas versões temporais mostradas nas Equações 4.3 e 4.5. Além disso, Kangas (1990) propõe usar duas redes de Kohonen em cascata com o mecanismo de memória de curto prazo interposto entre as duas redes. Esta noção de *hierarquia* é atraente pois supõem-se que o desempenho global do sistema seja melhorado por meio de sucessivos refinamentos da representação obtida por cada rede em uma camada inferior da hierarquia.

O autor reportou melhorias no reconhecimento de fonemas isolados e em testes com sinais de voz não-segmentados, quando comparado com uma rede de Kohonen convencional. Ademais, o autor comparou os três modelos temporais na classificação de uma seqüência temporal de duas dimensões em forma de “8”. Nesta seqüência, há um único ponto ambíguo, localizado na coordenada (0.5; 0.5), que a rede deve diferenciar para os instantes $t = 50$ e $t = 150$. A rede de Kohonen classifica os dois instantes com o mesmo neurônio, pois carece de uma memória. No entanto, os três modelos temporais elaborados por Kangas (1990) conseguem distinguir casos de ambigüidade espacial como este.

4.1.4 Differentiator-Integrator Neuron (DIN)

O *Mapa Auto-Organizável de Neurônios Diferenciadores-Integradores* (MOSER, 2004) é uma rede de Kohonen convencional que aplica memórias de curto prazo à entrada da rede. Cada memória funciona de acordo com a Equação 4.11, realizando uma derivação e/ou integração de uma dimensão do vetor $\mathbf{x}(t)$. Dessa forma, o espaço de entrada dos neurônios computacionais da rede de Kohonen é significativamente ampliado e melhorado. O autor reportou melhorias na segmentação de seqüências temporais com perturbações de ruído, translação, deformação nas dimensões x e y e rotação, frente a outros métodos conexionistas para processamento de informação temporal.

4.1.5 Temporal Kohonen Map (TKM)

O *Mapa de Kohonen Temporal* (CHAPPELL; TAYLOR, 1993) aplica integradores na saída de cada neurônio da rede. A memória de eventos passados é agregada à rede propriamente dita, tornando-a dinâmica. A integração é calculada a partir do histórico de ativação $\alpha_i(t)$ de cada um dos neurônios da rede. De forma similar à operação realizada pela memória de traço exponencial relatada na Seção 4.1.2, tem-se:

$$\alpha_i(t) = \lambda \alpha_i(t-1) - \frac{1}{2} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2 \quad (4.15)$$

onde λ é a profundidade da memória e deve ser um valor entre 0 e 1, $\mathbf{x}(t)$ é o item atual da seqüência, e $\mathbf{w}_i(t)$ é o vetor de pesos sinápticos do i -ésimo neurônio da rede. Cada neurônio calcula a sua ativação no instante t , a qual decai a uma taxa dada pela constante

de tempo $1/(1 - \lambda)$. O neurônio que apresentar o maior nível de atividade é considerado o vencedor ($v(t) = \max\{\alpha_i(t)\}$). Os pesos sinápticos são ajustados segundo Equação 3.9.

A rede TKM foi empregada com relativo sucesso no reconhecimento de palavras que ocorrem na mesma posição de uma seqüência porém em contextos diferentes. Por exemplo, as seqüências $\{\mathbf{a-b-c-d}\}$ e $\{\mathbf{x-y-z-d}\}$ são classificadas como distintas porque a letra \mathbf{d} tem evidentemente diferentes contextos temporais. Contudo, a rede falha na identificação de seqüências que exigem uma informação contextual maior e acaba por gerar ambigüidades para todas as seqüências que possuam a mesma média móvel. Estes problemas acontecem porque a seleção do neurônio vencedor é baseada principalmente em entradas mais recentes e a profundidade da memória é limitada pelo parâmetro λ .

4.1.6 Recurrent Self-Organizing Map (RSOM)

O *Mapa Auto-Organizável Recorrente* (VARSTA; HEIKKONEN; MILLÁN, 1997) é uma variação da rede TKM, na qual os integradores são aplicados à entrada de cada neurônio da rede. Os dois algoritmos fornecem essencialmente o mesmo resultado, entretanto, é importante observar que a grandeza sujeita à recursividade temporal na rede RSOM é um vetor $\mathbf{x}(t) - \mathbf{w}_i(t)$, e não uma norma. Logo, a integração realizada também dá um indicativo da direção do erro para cada neurônio em relação aos últimos itens da seqüência. Essa informação é explorada no ajuste dos pesos durante o processo de treinamento da rede. Matematicamente tem-se a seguinte expressão:

$$\mathbf{y}_i(t) = (1 - \lambda)\mathbf{y}_i(t - 1) + \lambda[\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (4.16)$$

O neurônio vencedor no instante t é aquele que apresentar menor valor de integração ($v(t) = \min\{\|\mathbf{y}_i(t)\|\}$). Os pesos sinápticos são alterados de acordo com a seguinte regra:

$$\mathbf{w}_i(t + 1) = \mathbf{w}_i(t) + \eta(t)h_{vi}(t)\mathbf{y}_i(t) \quad (4.17)$$

Em particular, quando $\lambda = 1$, a Equação 4.17 reduz-se a regra de aprendizagem proposta originalmente por Kohonen (vide Eq. 3.9).

4.1.7 Contextual Self-Organizing Map (CSOM)

O *Mapa Auto-Organizável Contextual* (VOEGTLIN, 2000, 2002) inclui uma espécie de *memória recorrente* que codifica aspectos temporais através de *laços de realimentação* da atividade dos neurônios. Cada neurônio da rede tem um campo receptivo definido por dois vetores de pesos sinápticos, \mathbf{w}_i^x e \mathbf{w}_i^y , relativos ao sinal de entrada atual $\mathbf{x}(t)$ e à atividade da rede no instante anterior $\mathbf{y}(t - 1)$, respectivamente. A ativação do i -ésimo neurônio da rede no instante t é definido como:

$$E_i(t) = \alpha\|\mathbf{x}(t) - \mathbf{w}_i^x(t)\|^2 + \beta\|\mathbf{y}(t - 1) - \mathbf{w}_i^y(t)\|^2 \quad (4.18)$$

onde α e β são duas constantes de esquecimento que devem ter o valor entre 0 e 1. Em especial, quando $\alpha = 1$ e $\beta = 0$, o comportamento da rede passa a ser análogo à rede de Kohonen original. O neurônio que melhor minimizar o erro de quantização $E_i(t)$ é escolhido como o vencedor ($v(t) = \min\{E_i\}$). A *informação de contexto*, representada pelo vetor $\mathbf{y}(t)$, é derivada do erro de quantização do instante anterior, conforme:

$$y_i(t) = \exp(-E_i(t - 1)), \quad \forall i \quad (4.19)$$

A forma não-linear da Equação 4.19 é uma sugestão do autor para garantir a estabilidade das representações temporais geradas. A informação de contexto faz com que os neurônios da rede se agrupem de acordo com a similaridade espacial do padrão atual e também conforme a classificação do padrão anterior. Os pesos sinápticos são ajustados a cada passo de tempo, a partir das seguintes regras:

$$\mathbf{w}_i^x(t+1) = \mathbf{w}_i^x(t) + \eta(t)h_{vi}(t)[\mathbf{x}(t) - \mathbf{w}_i^x(t)] \quad (4.20)$$

$$\mathbf{w}_i^y(t+1) = \mathbf{w}_i^y(t) + \eta(t)h_{vi}(t)[\mathbf{y}(t-1) - \mathbf{w}_i^y(t)] \quad (4.21)$$

Os valores iniciais para os pesos \mathbf{w}_i^x são selecionados de forma aleatória entre 0 e 1, enquanto que os valores iniciais para os pesos \mathbf{w}_i^y são feitos iguais a zero.

Outras referências para redes que usam a recorrência como forma de agregar informação temporal podem ser obtidas em: (HAGENBUCHNER; SPERDUTI; TSOI, 2003), onde é proposto o *Mapa Auto-Organizável para Dados Estruturados* (ou, SOMSD); e (STRICKERT; HAMMER, 2004), onde é sugerido o *Merging-SOM* (ou, MSOM).

4.1.8 Sequential Activation Retention and Decay NETWORK (SARDNET)

A *Rede de Retenção e Decaimento de Atividade Sequencial* (JAMES; MIIKKULAINEN, 1995), como o próprio nome diz, inclui um mecanismo muito simples de retenção e decaimento de atividade dos neurônios para gerar uma *representação esparsa e contínua* de seqüências temporais. Por essa razão, a capacidade de representação temporal se torna muito maior em relação aos modelos anteriores, pois a representação de uma seqüência é dada pela configuração de todos os neurônios da rede. O funcionamento do algoritmo inclui dois novos passos: atribuição do valor 1.0 à atividade do neurônio considerado vencedor e decaimento constante na atividade de todos os neurônios da rede:

$$y_i(t) = \tau y_i(t-1), \quad \forall i \quad (4.22)$$

onde $0 \leq \tau \leq 1$ é a taxa de decaimento de atividade. Para que seqüências temporais repetitivas não gerem ambigüidade na representação, a rede impede que o protótipo vencedor se repita de um instante para o seguinte. Esta restrição faz com que a rede possa ter dois protótipos representando o mesmo padrão de entrada.

4.1.9 Adaptive Resource Allocating Vector Quantization (ARAVQ)

A *Quantização Vetorial por Alocação Adaptativa de Recursos* (LINÅKER; NIKLASSON, 2000) consiste em uma rede neural convencional que aplica *quantização vetorial incremental* sobre uma *média móvel finita* do sinal temporal. Apesar de não ter a propriedade auto-organizável das redes de Kohonen até então discutidas, optou-se por apresentá-la por ser uma rede de aprendizado competitivo extremamente interessante. O vetor de entrada \mathbf{x} meramente é substituído pela média de uma memória de linhas de atraso:

$$\bar{\mathbf{x}}(t) = \frac{1}{\tau} \sum_{j=0}^{\tau-1} \mathbf{x}(t-j) \quad (4.23)$$

onde τ especifica o tamanho da janela temporal.

Não há uma organização espacial dos neurônios da rede, pois somente o neurônio vencedor que codifica um determinado protótipo tem seus pesos sinápticos ajustados em direção à média móvel. Logo, o aprendizado não leva em consideração a vizinhança do neurônio vencedor. O neurônio cujos pesos mais se aproximam da média móvel $\bar{\mathbf{x}}$ é

selecionado como o vencedor ($v = \min\{\|\bar{\mathbf{x}}(t) - \mathbf{w}_i(t)\|\}$). O aprendizado somente é realizado se o erro de quantização for relativamente pequeno ($\|\bar{\mathbf{x}}(t) - \mathbf{w}_v(t)\| < \frac{\epsilon}{2}$):

$$\mathbf{w}_v(t+1) = \mathbf{w}_v(t) + \eta[\bar{\mathbf{x}}(t) - \mathbf{w}_v(t)] \quad (4.24)$$

A partir da média móvel, são computadas duas medidas: a dispersão do sinal dentro da janela temporal (Equação 4.25) e o erro de quantização do melhor neurônio existente em relação à janela temporal (Equação 4.26):

$$d_{\mathbf{x}(t)} = \frac{1}{\tau} \sum_{j=0}^{\tau-1} \|\mathbf{x}(t-j) - \bar{\mathbf{x}}(t)\| \quad (4.25)$$

$$d_{\mathbf{w}(t)} = \frac{1}{\tau} \sum_{j=0}^{\tau-1} \min_{1 \leq l \leq k} \{\|\mathbf{x}(t-j) - \mathbf{w}_l(t)\|\} \quad (4.26)$$

A inclusão de novos neurônios é realizada se a condição $d_{\mathbf{x}(t)} \leq \min(\epsilon, d_{\mathbf{w}(t)} - \delta)$ for satisfeita. O critério de estabilidade, ϵ , determina o quão próximo a média móvel deve estar dos valores presentes na janela temporal para justificar a criação de um novo neurônio. O critério de novidade, δ , especifica o quão diferente a média móvel deve estar de todo e qualquer protótipo existente para justificar a criação de um novo neurônio. Inicialmente, quando não há neurônios alocados, $d_{\mathbf{w}(t)}$ vale $\epsilon + \delta$ para garantir a criação do primeiro neurônio se a variância for menor que o critério de estabilidade. O novo neurônio recebe o valor da média móvel $\bar{\mathbf{x}}$ calculada no instante de tempo t .

De forma resumida, esta rede usa a informação temporal somente para fins de segmentação e não para representação. Os protótipos gerados correspondem às médias dos padrões de entrada em janelas finitas de tempo e, portanto, não contêm informação de mudança neste período. A segmentação não leva em conta a relevância dos eventos para previsão, pois é orientada para detecção de mudanças nas características do sinal de entrada. Portanto, o método não descarta eventos raros taxando-os como ruído.

Os primeiros experimentos realizados com esta arquitetura utilizaram o robô Khepera em um labirinto com comportamento de “seguir paredes” (LINÅKER; NIKLASSON, 2000). Os oito sinais sensoriais de distância do robô, tipicamente ruidosos, foram submetidos à rede neural enquanto o robô explorava o ambiente. A segmentação obtida foi gerada em tempo real. Surgiram protótipos que representaram as configurações básicas encontradas em labirintos, como “corredor”, “bifurcação”, “curva para direita”, etc.

4.1.10 Resumo

A Tabela 4.1 resume as características principais de cada rede temporal apresentada:

- Topologia determina se os neurônios são organizados de forma hierárquica ou não. Como o modelo ARAVQ é um tipo de rede competitiva que não utiliza vizinhança (i.e., não é baseada em mapas auto-organizáveis), não há uma organização espacial dos protótipos e, por esse motivo, a sua respectiva coluna não foi preenchida;
- Posição indica se a memória de curto prazo foi aplicada externa ou internamente à rede. No primeiro caso, a memória é aplicada sobre os itens da seqüência temporal e não à rede propriamente dita, que permanece estática. No segundo caso, a memória é incluída na rede, tornando-a dinâmica;

- Representação define se o tipo de representação temporal gerada é de origem simbólica (local) ou de origem numérica (esparsa); e
- Natureza da memória especifica o mecanismo de memória de curto prazo empregado, a saber: (i) linha de atrasos de tempo; (ii) integradores; (iii) diferenciadores; ou (iv) laços de realimentação (recorrência).

Tabela 4.1: Resumo das redes neurais auto-organizáveis temporais

Modelo	Topologia	Posição	Representação	Natureza da memória
TDSOM ₁	Não-hierárquica	Externa	Local	Integradores aplicados na seqüência temporal $\mathbf{x}(t)$
TDSOM ₂	Não-hierárquica	Externa	Local	Linha de atrasos de tempo aplicada na seqüência temporal $\mathbf{x}(t)$
TDSOM ₃	Hierárquica	Interna	Local	Integradores aplicados na saída da primeira rede (SOM ₁)
DIN	Não-hierárquica	Externa	Local	Diferenciadores e integradores aplicados na seqüência temporal $\mathbf{x}(t)$
TKM	Não-hierárquica	Interna	Local	Integradores aplicados na saída de cada neurônio
RSOM	Não-hierárquica	Interna	Local	Integradores aplicados na entrada de cada neurônio
CSOM	Não-hierárquica	Interna	Local	Recorrência na atividade de todos os neurônios da rede
SOMSD	Não-hierárquica	Interna	Local	Recorrência somente na atividade do neurônio vencedor
MSOM	Não-hierárquica	Interna	Local	Recorrência na atividade de todos os neurônios da rede
SARDNET	Não-hierárquica	Interna	Esparsa	Decaimento exponencial na atividade de todos os neurônios.
ARAVQ	–	Interna	Local	Média móvel finita baseada em uma janela temporal fixa

4.2 As redes neurais auto-organizáveis construtivas

A proposta original da rede de Kohonen é baseada em uma arquitetura neural previamente definida e extremamente rígida, contendo o mesmo arranjo geométrico e número de neurônios durante todo o processo de treinamento. Em muitos casos, pouca ou nenhuma informação está disponível acerca do tamanho e da distribuição do conjunto de amostras do espaço de entrada \mathcal{X} . Logo, a definição de uma arquitetura de rede apropriada é um fator que depende, em grande parte, da experiência empírica do projetista.

Uma alternativa que resolve o problema suscitado é o *método de aprendizado construtivo* (FRITZKE, 1996, 1997b, 1997c), em que não necessita-se definir *a priori* a arquitetura da rede. Os métodos construtivos se enquadram em duas categorias principais, conforme o mecanismo utilizado para incluir e remover neurônios e conexões (ALPAYDIN, 1994):

1. Iniciar com uma arquitetura de rede de dimensão elevada e paulatinamente eliminar neurônios e conexões desnecessárias no intuito de aumentar a capacidade de

generalização da rede e reduzir os requisitos computacionais; ou

2. Iniciar com uma arquitetura de rede bastante simples e ir incrementando-a conforme necessário, geralmente, quando o treinamento com as unidades existentes não reduz o erro abaixo do valor desejado.

Em sua grande maioria, a pesquisa se concentra em técnicas híbridas, partindo de uma rede de dimensão reduzida e, ao longo do processo de treinamento e de acordo com o problema em questão, realizando a inclusão e remoção de neurônios e conexões na estrutura da rede, até que uma solução adequada seja apresentada.

O termo *aprendizado construtivo* muitas vezes é confundido com o termo *aprendizado incremental*. O aprendizado incremental caracteriza-se pela adaptação dos pesos sinápticos de uma rede neural após a apresentação de cada padrão de treinamento. Esse tipo de aprendizado é o oposto do *aprendizado em lote*, cuja alteração da rede se dá somente após a apresentação de todo o conjunto de treinamento. Portanto, o aprendizado incremental e o aprendizado em lote são apenas técnicas de treinamento e podem ser utilizados juntamente com o aprendizado construtivo (HAYKIN, 2001).

As principais redes neurais artificiais construtivas são analisadas a seguir.

4.2.1 Growing Cell Structures (GCS)

Uma das primeiras redes construtivas foi elaborada por Fritzke (1992) e consiste em um arranjo bidimensional de neurônios conectados entre si na forma de *triângulos*. A rede é limitada a um número máximo de *simplexos k -dimensionais*, onde k é um número inteiro escolhido antes de iniciar o treinamento. Por exemplo, se $k = 1$ o simplexo é uma *linha*, se $k = 2$ é um *triângulo*, se $k = 3$ é um *tetraedro*, e assim por diante. Posteriormente, Fritzke (1994) sugeriu uma generalização de sua proposta original com arranjos baseados em *hipertetraedros*, o que promove uma melhor recuperação da estrutura do espaço de entrada \mathcal{X} , no entanto, acaba por dificultar a sua visualização.

Ao longo do processo de treinamento, padrões são apresentados à rede, modificações nos pesos das conexões sinápticas entre os neurônios são realizadas e uma informação de erro de quantização vetorial é acumulada em cada neurônio. Essa *informação de erro acumulado* é útil para decidir onde devem ser incluídas novas unidades, ou seja, em que posição do arranjo deve-se adicionar novas unidades para diminuir o erro de quantização.

O procedimento de inserção de novos neurônios é realizado se o número de padrões apresentados à rede for múltiplo de um parâmetro λ . A nova unidade é inserida entre o neurônio com o maior erro acumulado e o seu vizinho mais distante. Depois dessa primeira unidade, o resto do simplexo é completado de tal forma que a rede resultante estará formada exclusivamente por *simplexos k -dimensionais*. A rede continua a se adaptar e a crescer até um determinado critério ser satisfeito, que pode ser o tamanho da rede, em número de neurônios ou conexões, ou alguma medida de erro mínimo aceitável.

Bruske e Sommer (1995) sugeriram uma extensão da rede GCS denominada *Growing Dynamical Cell Structure* (GDGS). A principal diferença é que a inserção de novos neurônios somente é realizada para assegurar a preservação da estrutura topológica da rede, e não para igualar o valor previsto do erro de cada neurônio. Além disso, a rede simplifica as estruturas internas que eram necessárias para manter os *simplexos*. Em (AHRNS; BRUSKE; SOMMER, 1995) uma rede GDGS modificada é aplicada com sucesso em um problema de controle clássico da Inteligência Computacional, chamado *Pole Balancing*.

Burzevski e Mohan (1996) propuseram uma rede hierárquica denominada *Hierarchical Growing Cell Structures* (HiGS), na qual submapas hierárquicos são criados de forma

gradual e conforme a complexidade do problema. A capacidade de detecção de agrupamentos e adaptação à distribuição dos dados é melhorada significativamente em relação a proposta original da rede de Fritzke. Anos mais tarde, Hodge e Austin (2001) elaboraram uma nova rede hierárquica chamada *Tree Growing Cell Structures* (TreeGCS), na qual a remoção de conexões eventualmente causa a criação de uma hierarquia de submapas.

4.2.2 Incremental Grid Growing (IGG)

O modelo *Incremental Grid Growing* (BLACKMORE, 1995; BLACKMORE; MIIKKULAINEN, 1995) foi proposto com o intuito de suplantiar a característica indesejável do *Growing Cell Structures* de, eventualmente, gerar arranjos em um espaço de elevada dimensão. O processo de treinamento é similar à rede de Kohonen tradicional, porém, iniciado com uma rede de apenas quatro neurônios conectados entre si formando um quadrado.

Novas unidades são adicionadas apenas nos limites externos do mapa (i.e., na *zona perimetral*), perto daqueles neurônios com maior erro acumulado, através de uma *heurística de crescimento*. A nova unidade é conectada diretamente ao *neurônio de fronteira* do qual derivou. Ao longo do treinamento, conexões entre neurônios adjacentes da rede são inseridas e removidas conforme a distância euclidiana entre o vetor de pesos sinápticos.

Os procedimentos de inserção e remoção de unidades e conexões permite que a rede mantenha uma estrutura retangular. A estrutura gerada assegura uma fácil visualização das amostras do vetor de entrada x , entretanto, nem sempre a melhor para recuperar a estrutura intrínseca do espaço de entrada \mathcal{X} .

Alahakoon, Halgamuge e Srinivasan (2000) propuseram um método de inicialização dos pesos sinápticos dos neurônios que são inseridos no arranjo. Este novo procedimento simplifica e reduz a possibilidade de gerar mapas inapropriados em relação à distribuição dos dados. O resultado é uma rede denominada *Growing Self-Organizing Map* (GSOM) que tem maior flexibilidade para adaptar-se à distribuição dos dados.

Merkel e colaboradores (2003) elaboraram um modelo hierárquico chamado *Adaptive Hierarchical Incremental Grid Growing* (AHIGG), onde cada camada individual da arquitetura é uma variação da rede IGG. Por essa razão, cada mapa individual pode crescer de forma irregular e, eventualmente, remover conexões entre unidades vizinhas. Dessa forma, as camadas dos níveis inferiores da hierarquia representam com maior fidelidade a distribuição dos dados, o que facilita a exploração visual.

4.2.3 Growing Grid (GG)

O modelo *Growing Grid* (FRITZKE, 1995a) é bastante similar às duas propostas anteriores. No entanto, a partir de um arranjo com quatro neurônios iniciais (i.e., uma grade 2×2), são inseridas *linhas* e *colunas* de novos neurônios no arranjo, promovendo uma *busca automática* da dimensão ideal para a rede. Assim, a topologia da rede é estritamente retangular, o que torna a sua visualização extremamente simples.

O algoritmo opera em duas fases distintas. Durante a *fase de crescimento* uma rede retangular vai sendo formada pela inclusão de linhas e colunas no arranjo até um determinado critério ser satisfeito. Valores estatísticos acumulados são usados para determinar onde novas unidades devem ser inseridas. Diferentemente da rede de Kohonen tradicional, o parâmetro da largura da vizinhança, σ , é um valor relativamente baixo e *constante* no tempo. Apesar disso, à medida que a rede cresce, a *fração* de todas as unidades que são adaptadas junto com o neurônio considerado vencedor diminui. Na *fase de convergência* o tamanho da rede não é mais alterado e a taxa de aprendizagem, η , sofre um decaimento exponencial para realizar um ajuste fino da rede, permitindo quantificar regularidades

estatísticas do espaço de entrada \mathcal{X} de forma mais acurada.

Dittenbach, Merkl e Rauber (2000) propuseram uma extensão ao modelo GG, conhecido por *Growing Hierarchical Self-Organizing Map* (GHSOM). Este modelo usa uma estrutura hierárquica de múltiplas camadas, onde cada uma delas é composta por um número independente de redes neurais com algoritmo similar ao GG. O primeiro nível é composto por apenas uma rede. Para cada unidade da primeira rede, pode ser criada uma sub-rede no próximo nível da hierarquia. Esse princípio é aplicado de forma recursiva aos outros níveis da rede. Dessa maneira, os níveis superiores representam informação mais genérica, enquanto que os inferiores representam a informação com maior detalhe.

4.2.4 Growing Neural Gas (GNG)

O consagrado modelo *Growing Neural Gas* (FRITZKE, 1995b) é uma versão construtiva do método *Neural Gas* (NG) proposto por Martinetz (1991), tendo surgido principalmente como resultado da combinação do mecanismo de crescimento do modelo *Growing Cell Structures* (FRITZKE, 1992) e do mecanismo de geração de topologia da técnica *Competitive Hebbian Learning* (CHL) (MARTINETZ, 1993; MARTINETZ; SCHULTEN, 1994).

O treinamento da rede é iniciado com apenas dois neurônios conectados entre si. Para cada padrão apresentado à rede, as duas unidades mais próximas são selecionadas conforme uma *função de distância euclidiana*. Se, porventura, não existir uma conexão entre elas, o algoritmo se encarrega de criá-la. A unidade vencedora juntamente com a sua *região de vizinhança* sofrem um processo de adaptação de pesos sinápticos.

Cada conexão da rede tem associada a si um contador que controla a sua *idade*. Esse valor somente é zerado caso os neurônios que ela conecte sejam escolhidos como vencedores. Se o contador de idade atingir certo valor a conexão é eliminada. Portanto, a idéia é priorizar as conexões novas em detrimento de antigas. Esta técnica visa atualizar a cada instante a topologia da rede, mantendo apenas conexões que sejam úteis.

Novas unidades vão sendo inseridas e removidas sucessivamente ao longo do processo de treinamento, conforme a taxa de crescimento dada pelo parâmetro λ . Para determinar em que posição do arranjo deverá ser adicionada uma nova unidade, informações relacionadas ao *erro de quantização* são coletadas durante o processo de adaptação. A nova unidade é inserida entre o neurônio com o maior erro acumulado e um de seus vizinhos, e as variáveis de erro são localmente redistribuídas. A rede continua a se adaptar e a crescer até um determinado *critério de parada* ser satisfeito (e.g., tamanho máximo da rede).

O arranjo da rede gerado ao final do processo de treinamento é chamado de *triangulação induzida de Delaunay* (FRITZKE, 1995b), pois preserva de maneira ótima a topologia da massa de dados de entrada. A rede é capaz de recuperar a estrutura de dados bastante complexos, inclusive com diferentes dimensões e regiões desconexas. No entanto, a convergência é demasiadamente lenta pois novos neurônios somente são inseridos na rede em interações múltiplas de uma constante de tempo predefinida (λ). Marsland, Shapiro e Nehmzow (2002) propuseram uma nova abordagem chamada *Grow When Required* (GWR), na qual neurônios são inseridos à rede de forma *automática*. Assim, a rede cresce rapidamente quando novos dados são apresentados a ela mas interrompe o processo de crescimento uma vez que tenha convergido para uma solução estável.

Uma das maiores deficiências do método GNG é a incapacidade de se adaptar rapidamente a dados com distribuições que mudam rapidamente (i.e., *dados não-estacionários*). Fritzke (1997a) sugere associar um *fator de utilidade* a cada neurônio da rede. A idéia é remover as unidades que contribuem pouco para a diminuição do erro de quantização e tentar adicionar novas unidades em posições mais significativas para o sucesso do

algoritmo. Trabalhos similares foram realizados por Lang e Warwick (2002), que desenvolveram a rede *Plastic Self-Organizing Map* (PSOM), e por Hung e Wermter (2005), que elaboraram a rede hierárquica *Dynamic Adaptive Self-Organising Hybrid Model* (DASH).

4.2.5 Adaptive Resonance Theory (ART)

Em meados dos anos 80, Grossberg estabeleceu um novo princípio de auto-organização conhecido como *teoria da ressonância adaptativa*. Basicamente, a teoria propõe auto-organizar padrões de entrada em categorias de reconhecimento, mantendo equilíbrio entre as propriedades de *plasticidade* e de *estabilidade* do sistema. A plasticidade é a característica que a rede possui de aprender um novo padrão, em qualquer tempo de sua operação, sem perder o conhecido adquirido anteriormente. A estabilidade é responsável pela habilidade da rede em agrupar padrões similares na mesma categoria de reconhecimento.

De acordo com Carpenter e Grossberg (2003), a teoria da ressonância adaptativa envolve uma camada de reconhecimento de baixo para cima (*bottom-up*) e uma camada generativa de cima para baixo (*top-down*). No momento em que o padrão de entrada e o padrão realimentado aprendido coincidirem, ocorre um estado dinâmico denominado “ressonância adaptativa” (i.e., amplificação e prolongamento da atividade neural).

A partir desta teoria, Grossberg juntamente com Carpenter propuseram um conjunto de redes neurais artificiais para o reconhecimento de padrões. Atualmente, existe uma família de arquiteturas ART, que cobrem desde o processamento de padrões binários (ART₁) e contínuos (ART₂ e ART₃), passando por uma abordagem difusa da informação (Fuzzy ART), até a integração de diversos módulos para possibilitar um esquema de aprendizado não-supervisionado e supervisionado simultaneamente (ARTMAP e Fuzzy ARTMAP) (GUAZZELLI, 1992). Em todas estas arquiteturas existem algumas características básicas comuns, como o *aprendizado construtivo* controlado por um *parâmetro de vigilância* que determina a *resolução* do processo de clusterização (*granularidade*).

Nas redes da família ART, dois subsistemas complementares interagem entre si para o processamento dos padrões de entrada: *subsistema atencional* e *subsistema de orientação*. O subsistema atencional é responsável pelo processamento de padrões de entrada familiares, estabelecendo respostas e representações internas mais precisas de tais padrões. O subsistema de orientação trata dos padrões estranhos, “resetando” o subsistema atencional quando tais padrões são apresentados à rede.

4.2.6 Resumo

A Tabela 4.2 resume as características principais de cada rede construtiva discutida:

- Arranjo define a topologia da rede neural (e.g., linear, simplexos, retangular);
- Dimensão específica como a rede realiza a projeção do espaço de entrada \mathcal{X} para o espaço de saída \mathcal{A} . Redes com *dimensionalidade fixa* são geralmente usadas para visualização de dados. Redes com *dimensionalidade variável* são comumente empregadas em tarefas de agrupamento e classificação de padrões;
- Modo de inserção define o mecanismo de crescimento da rede adotado. Em geral, a inserção é baseada em uma informação de erro de quantização acumulada em cada neurônio da rede, sendo realizada segundo uma taxa de crescimento predefinida; e
- Modo de poda determina o mecanismo de redução da rede adotado. As redes neurais construtivas que não possuem um método de redução e/ou eliminação de neurônios têm a respectiva coluna marcada com um hífen (sinal gráfico -).

Tabela 4.2: Resumo das redes neurais auto-organizáveis construtivas

Modelo	Arranjo	Dimensão	Modo de Inserção	Modo de Poda
GCS	Simplexos	Fixa	Constante λ ; um único neurônio próximo à zona de maior erro médio acumulado	-
IGG	Retangular	Fixa	Constante λ ; N -neurônios próximos à zona de fronteira com maior erro acumulado	Automático, através de um limiar de desconexão
GG	Retangular	Fixa	Constante λ ; uma linha ou coluna próxima à zona de maior erro médio acumulado	-
GNG	Grafo	Variável	Constante λ ; um único neurônio próximo à zona de maior erro médio acumulado	Automático, através de um contador de idade
GWR	Grafo	Variável	Automático, através de um limiar de atividade e de um contador de disparo	Automático, através de um contador de idade
ART	Linear	Variável	Automático, através de um parâmetro de vigilância	-

4.3 Considerações do capítulo

O presente capítulo não teve a pretensão de ser completo ou exaustivo, pelo contrário, teve como meta descrever de forma breve e de fácil entendimento os conceitos, as propriedades, limitações e variações das principais abordagens *temporais* e *construtivas* das redes neurais artificiais auto-organizáveis. O resultado do apanhado realizado neste capítulo foi sintetizado através das Tabelas 4.1 e 4.2. A análise qualitativa realizada se baseou nos comentários dos respectivos autores, em observações deduzidas a partir das características e peculiaridades de cada modelo neural e, principalmente, em simulações computacionais desenvolvidas a partir de uma implementação própria.

Até a data corrente, não há na literatura especializada da área algoritmos de aprendizado de natureza auto-organizável que integrem a representação temporal e o aprendizado construtivo. Neste trabalho, admite-se a hipótese de trabalho de que estas três características, quando combinadas adequadamente em uma rede neural artificial, viabilizam a resolução de *tarefas robóticas de horizonte infinito e de dimensão espaço-temporal*. Desse modo, o desenvolvimento de um novo modelo de rede neural artificial que contemple a auto-organização, a representação temporal e o aprendizado construtivo é extremamente desejável e constitui a principal meta a ser atingida no presente trabalho.

5 PROPOSTA DE REDE NEURAL ARTIFICIAL

Neste capítulo propõe-se um modelo de rede neural auto-organizável, temporal e construtivo. A rede é denominada GTSOM (do acrônimo em inglês para *Growing Temporal Self-Organizing Map*). Em um primeiro momento, dá-se uma visão geral da estrutura da rede que está sendo proposta. Logo após, detalha-se o seu funcionamento, com especial ênfase nos processos de inclusão, remoção e adaptação de neurônios e conexões. Em seguida, realiza-se uma análise sobre os seus parâmetros e variáveis e argumenta-se sobre sua plausibilidade biológica. Por fim, conclui-se com um resumo e uma discussão final.

5.1 Introdução

O presente trabalho propõe o desenvolvimento de um novo modelo de rede neural artificial voltado a aplicações da área da robótica, em especial a tarefas de *natureza espaço-temporal* e de *horizonte infinito*. Este modelo apresenta três características que o tornam único e que foram tomadas como guia para a sua concepção: *auto-organização*, *representação temporal* e *aprendizado construtivo*. A seguir faz-se uma breve análise de como estas três características estão relacionadas com o funcionamento da rede neural proposta:

- O processo de *aprendizagem auto-organizada* consiste em modificar repetidamente e de forma não-supervisionada os pesos sinápticos de todas as conexões da rede em resposta a *padrões de entrada espaço-temporais*. O algoritmo de aprendizagem ocorre, basicamente, através de mecanismos de *competição global*, *cooperação local* e *auto-amplificação seletiva*. Após o treinamento, os neurônios da rede se tornam *seletivamente sintonizados* a vários padrões de entrada espaço-temporais;
- O tempo é incluído na operação da rede neural por meio de uma *memória de curto prazo*. A memória de curto prazo, doravante denominada indistintamente de *memória* ou *unidade de memorização*, é aplicada externamente sobre os itens da seqüência temporal e não à rede propriamente dita, que permanece estática¹. A memória utilizada é uma sugestão de Moser (2004) e tem como função realizar uma *derivacão e/ou integração* da informação de entrada; e
- A construção automática da arquitetura da rede neural é realizada de acordo com uma *unidade de habituação*. A unidade de habituação regula tanto o *crescimento*

¹Kangas (1990) originariamente propôs três modelos neurais para representar aspectos seqüenciais de dados. Os modelos propostos por Kangas (1990) usam memórias de linhas de atraso de tempo e de traço exponencial. De forma inovadora, as memórias de curto prazo são aplicadas *externamente* à rede.

quanto à *poda* do número de neurônios da rede neural². A unidade de habituação utilizada é uma versão simplificada e discreta do *modelo de Habituação de DeLiang Wang e Michael Arbib* (BRISTOL; PURCELL; CAREW, 2003, p. 504–507).

5.2 Estrutura geral da rede neural

A rede neural proposta faz parte de uma família de modelos conexionistas auto-organizáveis, cuja essência é construir um mapeamento não-linear Φ de um espaço de entrada contínuo \mathcal{Z} para um espaço de saída discreto \mathcal{A} . O espaço \mathcal{Z} é formado por um subespaço de entrada sensorial \mathcal{X} e um subespaço de entrada contextual \mathcal{Y} , tal que:

$$\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), (\mathbf{x}_3, \mathbf{y}_3), \dots\} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots\} \quad (5.1)$$

O mapeamento não-linear Φ é realizado de forma adaptativa e mantém relações de similaridade entre os dois espaços:

$$\Phi : \mathcal{Z} \rightarrow \mathcal{A}, (\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}) \rightarrow (\Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{A}) \quad (5.2)$$

O subespaço de entrada sensorial \mathcal{X} tem uma topologia definida por certas relações métricas entre vetores $\mathbf{x} \in \mathcal{X}$. Do subespaço \mathcal{X} conhece-se apenas um conjunto finito de amostras organizadas conforme uma densidade de probabilidade $p(\mathbf{x})$:

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots\} \quad (5.3)$$

A dimensão do subespaço \mathcal{X} é representada por m . Um padrão de entrada $\mathbf{x} \in \mathbb{R}^m \subset \mathcal{X}$ define um ponto em um espaço euclidiano. De forma mais específica, \mathbf{x} representa um vetor de dimensão m -por-1, cujos elementos são todos números reais:

$$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T \quad (5.4)$$

onde t especifica o instante de tempo discreto em que a amostra foi observada e o índice superior T indica a transposição matricial.

O subespaço de entrada contextual \mathcal{Y} é gerado a partir da *memória de curto prazo* Ψ :

$$\Psi : \mathbf{x} \rightarrow \mathbf{y}, (\mathbf{x} \in \mathcal{X}) \rightarrow (\Psi(\mathbf{x}) \in \mathcal{Y}) \quad (5.5)$$

A principal finalidade de uma memória de curto prazo é extrair informações temporais implícitas de um sinal de entrada original. Dessa forma, a saída da rede neural se torna uma função do tempo e tarefas que envolvam o processamento de padrões espaço-temporais se tornam passíveis de serem realizadas pela rede neural. Basicamente, há dois aspectos a serem considerados quanto à incorporação de uma memória de curto prazo no projeto de uma rede neural: a *posição* e a *natureza* da memória. A posição indica se a memória é aplicada *externa* ou *internamente* à rede neural. A natureza diz respeito ao mecanismo de funcionamento da memória, que pode ser por *linha de atrasos de tempo*, *diferenciadores*, *integradores* e/ou *laços de realimentação (recorrência)*.

²Marsland, Nehmzow e Shapiro (1999) originariamente propuseram usar uma unidade de habituação simplificada para controlar somente o crescimento de uma rede neural auto-organizável. Mais especificamente, o *modelo de Habituação de Stanley* foi acoplado a uma rede de Kohonen convencional. A rede HSOM (*Habituating Self-Organizing Map*), como é denominada em (MARSLAND; NEHMZOW; SHAPIRO, 2005), mostrou-se adequada em várias aplicações robóticas de detecção de novidades.

Na rede neural em questão, a memória é aplicada sobre os itens da seqüência temporal e não à rede propriamente dita, que permanece estática. A memória utilizada é uma sugestão de Moser (2004) e tem como função realizar uma *derivação e/ou integração* da informação contida em cada dimensão do vetor $\mathbf{x}(t)$, dependendo dos parâmetros escolhidos³. Matematicamente, a memória tem o seu funcionamento regido por:

$$\bar{\mathbf{x}}(t) = (1 - \mu)\mathbf{x}(t) + \mu\bar{\mathbf{x}}(t - 1) \quad (5.6)$$

$$\mathbf{y}(t) = (1 - \eta)(\mathbf{x}(t) - \bar{\mathbf{x}}(t - 1)) + \eta\mathbf{y}(t - 1) \quad (5.7)$$

onde μ e η são duas constantes de integração que devem ter o valor entre 0 e 1, e \mathbf{y} é a saída da memória no instante de tempo t . Consideram-se as condições iniciais $\bar{\mathbf{x}}(0) = 0$ e $\mathbf{y}(0) = 0$. A representação gráfica das Equações 5.6 e 5.7 é ilustrada na Figura 5.1.

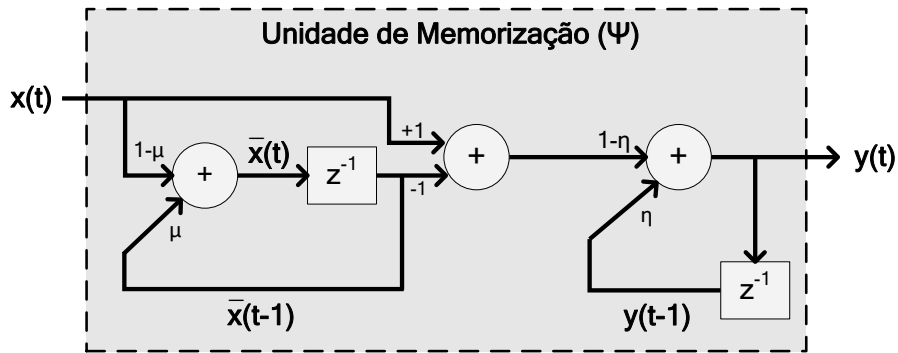


Figura 5.1: Representação gráfica da unidade de memorização.

O subespaço de entrada contextual \mathcal{Y} tem uma topologia definida por certas relações métricas entre vetores $\mathbf{y} \in \mathcal{Y}$, tal que $\mathbf{y}_i = \Psi(\mathbf{x}_i), \forall_i \in \mathcal{X}$. A dimensão de \mathcal{Y} é denotada por m . A representação de um vetor $\mathbf{y} \in \mathbb{R}^m \subset \mathcal{Y}$ de dimensão m -por-1 é dada por:

$$\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots\} \quad (5.8)$$

$$\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_m(t)]^T \quad (5.9)$$

Para fins de simplificação, considera-se o vetor $\mathbf{z} \in \mathbb{R}^{2m} \subset \mathcal{Z}$ de dimensão $2m$ -por-1 como o *padrão espaço-temporal* aplicado à camada de entrada da rede neural, tal que:

$$\mathbf{z}(t) = [x_1(t), x_2(t), \dots, x_m(t), y_1(t), y_2(t), \dots, y_m(t)]^T \quad (5.10)$$

O espaço de saída discreto \mathcal{A} é definido pelo arranjo geométrico de um conjunto de neurônios. O número de neurônios que compõe o arranjo não é predeterminado pelo projetista da rede e nem constante durante o processo de aprendizagem. Pelo contrário, a rede é capaz de inserir e remover neurônios de forma automática, de modo que continuamente esteja adaptada à distribuição dos dados de entrada. O conjunto de neurônios da rede neural é representado por \mathcal{A} e é limitado a um número máximo de neurônios n_{max} :

$$\mathcal{A} = \{n_1, n_2, n_3, \dots\} \quad (5.11)$$

Cada neurônio da rede está totalmente conectado às m unidades de entrada \mathbf{x} através do vetor de pesos sinápticos $\mathbf{w}^x \in \mathbb{R}^m$ e às m unidades de contexto \mathbf{y} por meio de termos

³Para maiores detalhes sobre a memória de curto prazo de Lúcio Moser (2004), consulte a Seção 4.1.2.4.

ponderados $\mathbf{w}^y \in \mathbb{R}^m$. Para fins de simplificação, considera-se o vetor $\mathbf{w}_i^z \in \mathbb{R}^{2m}$ de dimensão $2m$ -por-1 como o *vetor de pesos sinápticos espaço-temporal* de um neurônio i :

$$\mathbf{w}_i^z(t) = \underbrace{[w_{i1}^x(t), w_{i2}^x(t), \dots, w_{im}^x(t)]}_{\mathbf{w}_i^x(t)}, \underbrace{[w_{i1}^y(t), w_{i2}^y(t), \dots, w_{im}^y(t)]}_{\mathbf{w}_i^y(t)} \quad \forall_i = 1, 2, \dots, n \quad (5.12)$$

onde \mathbf{w}_i^x e \mathbf{w}_i^y têm a mesma dimensão do vetor de entrada \mathbf{x} (i.e., igual a m) e n é o número de neurônios da rede no instante de tempo t , i.e., $n = |\mathcal{A}(t)|$.

Além de pesos de entrada \mathbf{w}^x e de contexto \mathbf{w}^y , cada neurônio está conectado a uma unidade de habituação Θ através de um escalar $w^h \in \mathbb{R}^1$, tal que $w_i^h = \Theta(S_i), \forall_i \in \mathcal{A}$.

$$\Theta : \mathbf{S} \rightarrow \mathbf{w}^h, (\mathbf{S} \in \mathbb{R}^n) \rightarrow (\Theta(\mathbf{S}) \in \mathbb{R}^n) \quad (5.13)$$

A unidade de habituação é uma versão simplificada e discreta do modelo de Habituação de Wang e Arbib (BRISTOL; PURCELL; CAREW, 2003, p. 504–507)⁴. No modelo neural proposto, a unidade de habituação serve de referência para tomada de decisão estratégica nos procedimentos de crescimento e de poda de neurônios. Matematicamente, a unidade de habituação de um neurônio de índice i tem o funcionamento regido pela Equação 5.14:

$$w_i^h(t) = w_i^h(t-1) + \gamma(\alpha(1 - w_i^h(t-1)) - \beta w_i^h(t-1)S_i(t)) \quad \forall_i = 1, 2, \dots, n \quad (5.14)$$

onde α define a taxa de recuperação da habituação, β controla a velocidade de habituação, γ governa a variação da taxa de habituação e $S_i \in \mathbb{R}^1$ é um escalar que representa a estimulação externa recebida pelo i -ésimo neurônio. A estimulação externa $S_i(t)$ no instante t é a própria atividade do i -ésimo neurônio do instante anterior $t-1$ (vide Eq. 5.31). A Equação 5.14 é derivada a partir da Equação 4.13 fixando $z(t) = 1$, para eliminar os efeitos de longo prazo da unidade de habituação, e considerando $w_i^h(0) = 1.0$, para especificar o estado inicial da unidade de habituação no valor de 1.0. A representação gráfica da Equação 5.14 é ilustrada na Figura 5.2.

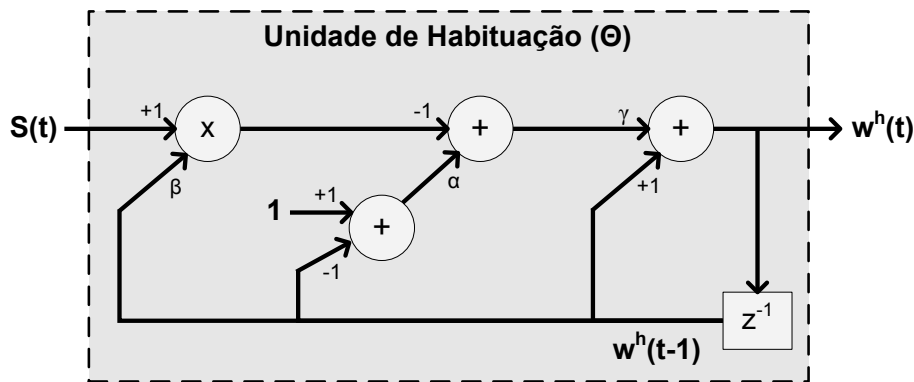


Figura 5.2: Representação gráfica da unidade de habituação.

Cada neurônio da rede pode estar conectado a quaisquer outros neurônios, com exceção de si mesmo. Assim, as conexões entre os neurônios podem criar tanto malhas regulares como diversas figuras geométricas e, por essa razão, uma maior capacidade de

⁴Para maiores informações a respeito da unidade de habituação de DeLiang Wang e Michael Arbib (BRISTOL; PURCELL; CAREW, 2003, p. 504–507), consulte a Seção 4.1.2.5.

aprendizado é conferida à rede neural. O conjunto de conexões sinápticas da rede neural é representado por $\mathcal{C} \subset \mathcal{A} \times \mathcal{A}$ e é limitado a um número máximo de conexões c_{max} :

$$\mathcal{C} = \{(n_1, n_2), (n_1, n_3), (n_2, n_3), \dots\} \quad (5.15)$$

As conexões entre neurônios são simétricas e valoradas, tal que:

$$(n_i, n_j) \in \mathcal{C} \iff (n_j, n_i) \in \mathcal{C} \quad (5.16)$$

$$\text{idade} : (n_i, n_j) \rightarrow \mathbb{N}^1 \cup \{0\} \quad (5.17)$$

onde $\mathbb{N}^1 \cup \{0\}$ representa a idade da conexão. Esse valor tem como propósito determinar o tempo de vida útil da conexão. Caso o valor exceda o limiar i_{max} a conexão é removida.

A vizinhança topológica de um neurônio i é representada por $N_i \subset \mathcal{A}$. No tempo t , a vizinhança topológica N_i vale:

$$N_i = \{n_j \in \mathcal{A} | (n_i, n_j) \in \mathcal{C}\} \quad (5.18)$$

A Figura 5.3 representa de forma esquemática os elementos básicos da rede neural. A rede tem uma estrutura de propagação para frente com apenas uma camada computacional de neurônios. Cada neurônio está totalmente conectado ao vetor de entrada x , ao vetor de contexto y e a uma fonte externa S , por meio de termos ponderados w . A unidade de memorização Ψ extrai informações temporais implícitas do vetor de entrada x e tem papel fundamental no processamento de padrões espaço-temporais. A unidade de habituação Θ funciona como um filtro de novidade para a rede e tem papel fundamental nos processos de crescimento e de poda de neurônios. Cada neurônio pode estar conectado a quaisquer outros neurônios no arranjo, com exceção de si mesmo. Cada conexão sináptica tem certa idade que controla o seu tempo de vida restante. A representação gerada pela rede é espacial e temporal, uma vez que a saída se refere tanto ao padrão de entrada x quanto à informação de contexto y gerada pela memória Ψ . A saída da rede é um vetor binário com todos os bits em zero, exceto o bit que corresponde ao neurônio vencedor.

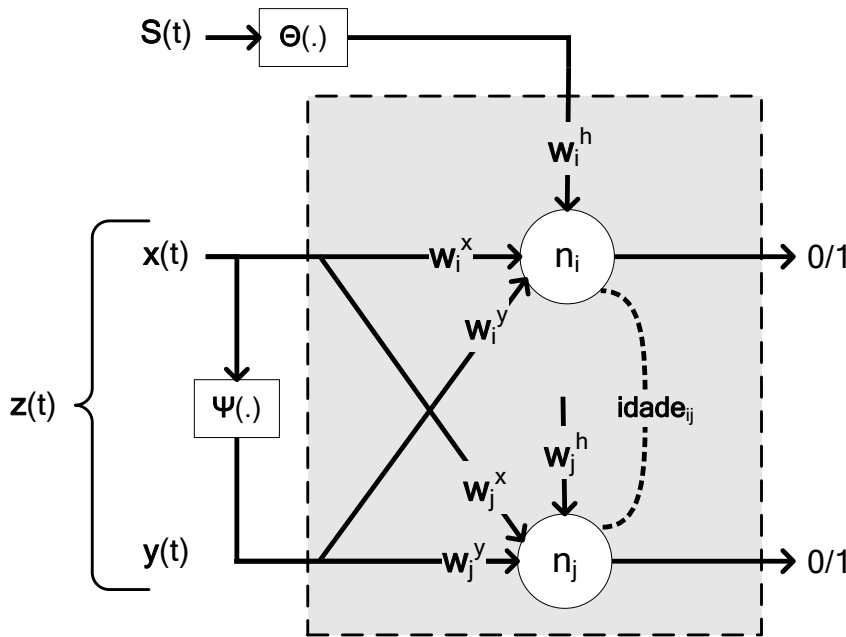


Figura 5.3: Elementos básicos da rede neural proposta.

5.3 Funcionamento da rede neural

O algoritmo de aprendizagem responsável pela formação do mapa auto-organizável é de natureza *auto-organizada*. O objetivo de um algoritmo auto-organizado é *descobrir* padrões significativos ou características nos dados de entrada *sem* o auxílio de um professor (i.e., de forma não-supervisionada) (HAYKIN, 2001, p. 429). Para tanto, o algoritmo deve dispor de um conjunto de regras de natureza *local*⁵, que o capacitam a aprender a calcular um mapeamento de entrada-saída com propriedades desejáveis específicas.

A Figura 5.4 ilustra como a rede neural proposta implementa o mapeamento Φ (vide Eq. 5.2). Os círculos cheios indicam o vetor de pesos dos neurônios da rede. As linhas representam conexões sinápticas entre os neurônios. O subespaço de entrada sensorial \mathcal{X} tem uma distribuição uniforme de dados, cuja forma geométrica se assemelha ao número 8. O subespaço de entrada contextual \mathcal{Y} é formado a partir da memória de curto prazo Ψ . O espaço de saída discreto \mathcal{A} é gerado a partir da aplicação do algoritmo proposto.

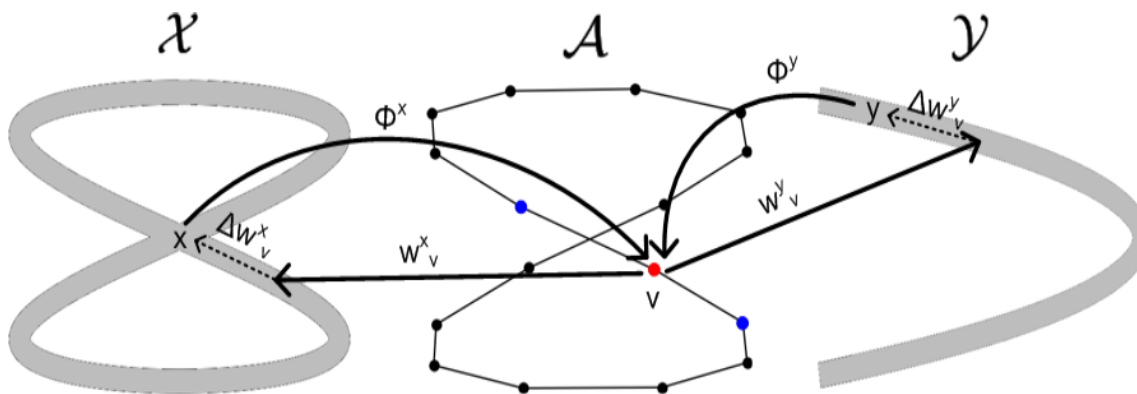


Figura 5.4: Mapeamento Φ implementado pela rede neural proposta.

No exemplo da Figura 5.4, as amostras do espaço de entrada são submetidos à rede neural em uma determinada ordem, de tal maneira que a seqüência possa ser interpretada como a trajetória de um robô realizada dentro de um ambiente⁶. Desta forma, existe um único ponto de ambigüidade na seqüência temporal, posicionado no centro do 8 e indicado pela letra x. Em uma distribuição de 200 pontos igualmente espaçados, os dois instantes em que a seqüência passa por x são $t = 50$ e $t = 150$. A rede neural resultante da aplicação do algoritmo proposto consegue identificar casos de ambigüidade espacial como este, pois classifica os dois instantes de tempo com dois neurônios diferentes.

A organização básica da arquitetura da rede neural proposta envolve dois componentes principais: o *subsistema de atuação* e o *subsistema de controle*. O subsistema de atuação tem como finalidades compor o padrão espaço-temporal segundo o vetor de entrada x e o vetor de contexto y ; e agrupar os padrões espaço-temporais em categorias de reconhecimento. O subsistema de controle é formado por um conjunto de mecanismos que governam o crescimento e a poda de neurônios e conexões sinápticas da rede neural. Basicamente, cinco processos estão envolvidos na formação do mapa auto-organizável:

- *Competição global*. Para cada padrão espaço-temporal $z(t)$, os neurônios computacionais da rede neural competem entre si para serem ativados. A competição

⁵O termo local refere-se à *restrição de localidade*, no sentido de que a computação realizada por um neurônio é influenciada apenas por aqueles neurônios que estão em contato físico com ele.

⁶Para maiores detalhes sobre o experimento computacional do oito, consulte a Seção 6.3.1.

consiste em identificar o neurônio cujo vetor de pesos sinápticos melhor represente o padrão de entrada, em termos de uma medida de distância euclidiana. O reconhecimento não depende apenas do valor presente da entrada $\mathbf{x}(t)$, mas também de valores passados, que estão armazenados na memória de contexto $\mathbf{y}(t)$.

- *Cooperação local.* O neurônio vencedor e seus vizinhos no mapa discreto interagem por meio de uma função denominada *função vizinhança*. Como a rede não tem uma estrutura fixa (e.g., retangular, triangular, hexagonal ou outras), a função vizinhança de um determinado neurônio meramente define que neurônios são vizinhos imediatos a ele no instante de tempo t .
- *Adaptação.* O ajuste do vetor de pesos sinápticos é confinado à vizinhança do neurônio considerado vencedor da competição, de tal forma que o neurônio vencedor juntamente com sua vizinhança se tornem ainda mais especializados no reconhecimento do último padrão espaço-temporal apresentado à rede. Como a rede é não-supervisionada, o vetor de pesos de um dado neurônio deve ser modificado apenas como função do estímulo de entrada, tal que seja possível extrair algum tipo de regularidade espaço-temporal presente no espaço de entrada contínuo \mathcal{Z} .
- *Crescimento.* O procedimento de inserção de novos neurônios e/ou conexões é realizado de forma automática e em qualquer instante de tempo, sem que isto obrigue a reinicialização do processo de aprendizagem. Esse procedimento retira o ônus do projetista em ter que especificar *a priori* as dimensões da rede neural e capacita a rede para lidar com uma distribuição de dados não-estacionários.
- *Poda.* O procedimento de remoção de neurônios e/ou conexões é realizado de forma automática e visa manter a estrutura da rede em conformidade com a distribuição de dados de entrada. Esse procedimento remove neurônios e conexões que não sejam úteis sob o ponto de vista da precisão e preservação da topologia da rede.

Nas subseções seguintes é descrito em maiores detalhes cada um dos processos envolvidos na formação do mapa auto-organizável. Os processos de competição global e cooperação local estão em concordância com os princípios de auto-organização apresentados em (HAYKIN, 2001, p. 430–432). A presença de redundância na informação contida nos padrões de entrada, não mencionada explicitamente na descrição do algoritmo, é necessária para a aprendizagem visto que ela provê o conhecimento a ser extraído pela rede. É importante salientar que o algoritmo opera em apenas uma única fase, denominada *fase de aprendizagem*. Ou seja, não existe uma fase distinta entre treinamento e atuação.

5.3.1 Processo de inicialização

O processo de inicialização consiste em criar apenas dois neurônios no espaço de saída discreto \mathcal{A} . Com o intuito de não impor nenhuma organização prévia à rede, são atribuídos valores pequenos e aleatórios aos pesos sinápticos dos dois neurônios, n_1 e n_2 . Normalmente, esses valores são restritos ao intervalo fechado $[0, 1]$. O conjunto de conexões sinápticas \mathcal{C} da rede é definido como vazio.

$$\mathcal{A}(0) = \{n_1, n_2\} \quad (5.19)$$

$$\mathcal{C}(0) = \emptyset \quad (5.20)$$

5.3.2 Processo de competição

O processo de competição pode ser modelado de forma *discreta* ou *contínua*. Na versão discreta, o processo é implementado com a ajuda de um *juiz externo*, que decide qual neurônio é considerado o vencedor, adotando algum critério baseado numa métrica de distância entre o vetor de entrada e o vetor de pesos dos neurônios. Na versão contínua, o processo é *autônomo*, envolvendo conexões realimentadoras para produzir excitação central e inibição lateral, cujo comportamento é dado por um sistema de equações diferenciais. Por questões de simplicidade, utiliza-se a versão discreta de operação.

Para cada padrão apresentado à rede, as duas unidades mais próximas são selecionadas conforme uma *função de distância euclidiana*. Como se deseja preservar a topologia do espaço de entrada, o padrão apresentado à rede não deve ser normalizado. Com isso, o processo de escolha dos vencedores somente deve ser baseado na distância euclidiana entre o padrão \mathbf{z} e o vetor de pesos sinápticos \mathbf{w}^z de cada neurônio da rede.

$$v(t) = \underset{i}{\operatorname{argmin}} \{ \|\mathbf{z}(t) - \mathbf{w}_i^z(t)\| \} \quad \forall_i = 1, 2, \dots, n \quad (5.21)$$

$$s(t) = \underset{i}{\operatorname{argmin}} \{ \|\mathbf{z}(t) - \mathbf{w}_i^z(t)\| \} \quad \forall_i \neq v, \quad i = 1, 2, \dots, n \quad (5.22)$$

onde v e s são os índices que representam o primeiro e o segundo neurônios vencedores, respectivamente, para o padrão \mathbf{z} apresentado à rede no instante de tempo t , dentre todos os n neurônios presentes na rede. A norma euclidiana $\|\cdot\|$ entre um par de vetores m -por-1 é definida como segue:

$$\|\mathbf{z}(t) - \mathbf{w}_i^z(t)\| = \sqrt{[\mathbf{z}(t) - \mathbf{w}_i^z(t)]^T [\mathbf{z}(t) - \mathbf{w}_i^z(t)]} = \sqrt{\sum_{j=1}^m [z_j(t) - w_{ij}^z(t)]^2} \quad (5.23)$$

onde T denota o vetor transposto. Assume-se que os componentes dos vetores $\mathbf{z}(t)$ e $\mathbf{w}_i^z(t)$ são independentes (i.e., não-correlacionados).

A similaridade entre os vetores \mathbf{z} e \mathbf{w}_i^z é definida como o *recíproco* da distância euclidiana. Quanto mais próximo entre si estiverem os seus elementos individuais, menor será a distância euclidiana e, portanto, maior será a similaridade entre eles. A distância euclidiana está relacionada com a “projeção” do vetor \mathbf{z} sobre o vetor \mathbf{w}_i^z . Para vetores normalizados, quanto menor a distância euclidiana $\|\mathbf{z} - \mathbf{w}_i^z\|$ maior será o *produto interno* entre os vetores \mathbf{z} e \mathbf{w}_i^z . Matematicamente, o produto interno entre os vetores \mathbf{z} e \mathbf{w}_i^z é definido como segue:

$$(\mathbf{z}, \mathbf{w}_i^z) = \mathbf{z}^T \mathbf{w}_i^z = \sum_{j=1}^m z_j w_{ij}^z \quad (5.24)$$

5.3.3 Processo de cooperação

O processo de cooperação consiste em determinar a *vizinhança topológica* centrada no neurônio vencedor v . Como a rede não tem uma estrutura fixa, a vizinhança de um neurônio de índice i simplesmente é o conjunto de neurônios que tem conexão sináptica direta com ele no instante de tempo t , i.e., N_i (vide Eq. 5.18). A *função vizinhança* do neurônio vencedor v é definida por h_{vi} , tal que:

$$h_{vi}(t) = \begin{cases} 1 & \text{se } n_i \in N_v \\ 0 & \text{caso contrário} \end{cases} \quad (5.25)$$

Contrária a inúmeras considerações biológicas, a implicação da *função vizinhança* descrita na Equação 5.25 é que todos os neurônios localizados dentro da vizinhança topológica N_v disparam a uma mesma taxa, e que as interações entre estes neurônios são independentes de suas distâncias laterais ao neurônio vencedor v . Não obstante, essa função é utilizada devido ao seu baixo custo computacional e à facilidade de codificação.

O neurônio vencedor v e seus vizinhos no mapa discreto interagem de tal forma que neurônios próximos tendem a reconhecer características similares. À medida que dois padrões se aproximam no espaço de entrada contínuo \mathcal{Z} , a localização dos respectivos neurônios vencedores fica cada vez mais próxima no arranjo discreto \mathcal{A} . A essa característica dá-se o nome de *propriedade de preservação de topologia*.

5.3.4 Processo de adaptação

Uma vez determinado o neurônio vencedor v e a sua vizinhança topológica N_v , ocorre o processo de adaptação dos pesos sinápticos. Este processo basicamente faz com que o neurônio vencedor v e os seus vizinhos imediatos se tornem mais especializados no reconhecimento do último padrão de entrada \mathbf{z} apresentado à rede. Matematicamente, a regra de aprendizagem pode ser calculada a partir das Equações 5.26 e 5.27:

$$\mathbf{w}_v^z(t+1) = \mathbf{w}_v^z(t) + \epsilon_v(t)[\mathbf{z}(t) - \mathbf{w}_v^z] \quad (5.26)$$

$$\mathbf{w}_i^z(t+1) = \mathbf{w}_i^z(t) + \epsilon_n(t)h_{vi}(t)[\mathbf{z}(t) - \mathbf{w}_i^z] \quad \forall_i = 1, 2, \dots, n \quad (5.27)$$

onde $0 < \epsilon_v < 1$ é a taxa de aprendizagem do neurônio vencedor v , $0 < \epsilon_n < 1$ é a taxa de aprendizagem da vizinhança topológica N_v centrada no neurônio vencedor v , h_{vi} é a função vizinhança (vide Eq. 5.25), e n é o número de neurônios da rede no instante t .

As variáveis ϵ_v e ϵ_n são dinâmicas e não fazem parte dos parâmetros livres da rede neural. Da mesma forma que Berglund e Sitte (2006) calcularam o valor da taxa de aprendizagem para uma rede de Kohonen convencional, o valor de ϵ_v é computado a cada instante de tempo conforme a Equação 5.28:

$$\epsilon_v(t) = \frac{\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\|}{d(t)} \quad (5.28)$$

$$d(t) = \max(\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\|, d(t-1)) \quad (5.29)$$

onde $d(t)$ especifica uma *distância euclidiana normalizada* entre o vetor de entrada $\mathbf{z}(t)$ e o vetor de pesos sinápticos $\mathbf{w}_v^z(t)$ do neurônio vencedor v . Considera-se a condição inicial $d(0) = \|\mathbf{z}(0) - \mathbf{w}_v^z(0)\|$. À proporção que a distância $\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\|$ se torna demasiadamente grande, o valor calculado para a taxa de aprendizagem $\epsilon_v(t)$ tende a 1, de modo que seja realizado um rápido ajuste na rede. No momento que a distância $\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\|$ se torna pequena, o valor computado para $\epsilon_v(t)$ sofre um decaimento exponencial, tal que seja realizado um ajuste fino da rede. Ou seja, $\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\| \approx d(t) \iff \epsilon_v(t) \approx 1$. Em contrapartida, $\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\| \ll d(t) \iff \epsilon_v(t) \approx 0$.

As duas variáveis ϵ_v e ϵ_n controlam a velocidade com que os pesos sinápticos são modificados e, portanto, especificam a velocidade com que a rede adquire um novo conhecimento. Considera-se para efeitos de ordenamento e convergência que ϵ_n seja pelo menos uma ordem de magnitude menor que ϵ_v , tal que $\epsilon_n \ll 1$:

$$\epsilon_n = \frac{\epsilon_v}{10} \quad (5.30)$$

À medida que o processo de aprendizagem se desenvolve, cada neurônio tem sua sinapse de habituação, w^h , ajustada. A modificação da força da conexão w^h é realizada

conforme a unidade de habituação Θ (vide Eq. 5.14). A magnitude do ajuste da sinapse $w_i^h(t)$ de um neurônio de índice i está associada ao valor recebido por sua estimulação externa $S_i(t)$. A estimulação externa $S_i(t)$ aplicada à sinapse $w_i^h(t)$ é dada por:

$$S_i(t) = \begin{cases} \exp(-\|\mathbf{z}(t) - \mathbf{w}_i^z(t)\|) & \text{se } i = v \text{ ou } n_i \in N_v \\ 0 & \text{caso contrário} \end{cases} \quad (5.31)$$

A sinapse de habituação do neurônio vencedor v e da sua vizinhança topológica N_v são atualizadas em direção ao ponto de máxima habituação (valor zero) e, por essa razão, o valor da estimulação externa S deve ser diferente de zero. Por questões de simplicidade, optou-se por utilizar o valor da própria atividade dos neurônios, tal que S vale $\exp(-\|\mathbf{z}(t) - \mathbf{w}_i^z(t)\|)$. Já as sinapses de habituação dos demais neurônios da rede neural sofrem um processo de sensibilização e tendem ao estado inicial de não habituação (valor 1), haja vista que o valor da estimulação externa S é feito igual a 0. Para maiores detalhes sobre o comportamento da unidade de habituação frente à estimulação externa, consulte as propriedades da unidade de habituação na Seção 4.1.2.5.

A Figura 5.5 esclarece como a rede neural realiza o ajuste de pesos sinápticos.

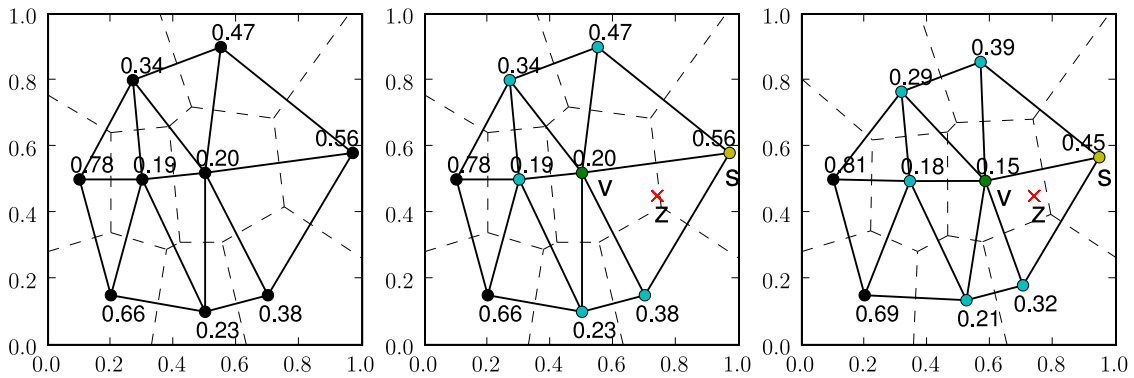


Figura 5.5: Ajuste de pesos sinápticos na rede neural proposta. Os números reais indicam os valores das sinapses de habituação de cada neurônio da rede. À esquerda, configuração da rede antes da atualização sináptica. No centro, um padrão \mathbf{z} é apresentado à rede e o primeiro e segundo neurônios vencedores, v e s , respectivamente, são selecionados. À direita, os vetores de pesos \mathbf{w}^z do neurônio vencedor v e da vizinhança topológica N_v são ajustados em direção ao padrão \mathbf{z} , as sinapses de habituação w^h do neurônio vencedor v e da vizinhança topológica N_v são atualizadas em direção ao ponto de máximo habituação (valor zero) e as sinapses de habituação dos demais neurônios da rede são modificadas rumo ao estado inicial de não habituação (valor 1).

5.3.5 Processo de inclusão e remoção de neurônios

O processo de inclusão de neurônios ocorre quando o erro de quantização da rede neural for demasiadamente alto e o padrão de entrada for relativamente novo para a rede. Sendo assim, a rede cresce rapidamente quando novos dados são apresentados a ela mas interrompe automaticamente o processo de crescimento uma vez que tenha convergido para uma solução estável. Matematicamente, a inclusão de um neurônio no mapa de características ocorre se a seguinte condição for satisfeita:

$$\mathcal{A}(t+1) = \begin{cases} \mathcal{A}(t) \cup \{n_r\} & \text{se } \exp(-\|\mathbf{z}(t) - \mathbf{w}_v^z(t)\|) < \rho \text{ e } w_v^h(t) < \delta \\ \mathcal{A}(t) & \text{caso contrário} \end{cases} \quad (5.32)$$

onde r é o índice da nova unidade a ser adicionada na rede, $0 < \rho < 1$ é o critério de plasticidade e $0 < \delta < 1$ é o critério de novidade.

O critério de plasticidade, ρ , define a máxima similaridade admissível para criar um novo neurônio. A medida de máxima similaridade determina a granularidade da rede (especificidade \times generalidade) e é codificada por uma função exponencial fixada numa escala de 0 a 1 (i.e., $\exp(-0) = 1$ e $\exp(-\infty) = 0$). Se o vetor de pesos sinápticos $\mathbf{w}_v^z(t)$ do neurônio vencedor casar suficientemente bem com o padrão $\mathbf{z}(t)$, a atividade instantânea do neurônio vencedor será próxima de 1; caso contrário, a atividade instantânea do neurônio vencedor será próxima de zero. Dessa forma, quanto maior o valor estabelecido para o parâmetro de plasticidade, mais específico será o critério de plasticidade e uma quantidade maior de neurônios será adicionada na rede. Ou seja, $\rho \rightarrow 1 \therefore \|\mathcal{A}\| \rightarrow n_{max}$.

O critério de novidade, δ , determina uma medida de novidade baseada no valor da sinapse de habituação do neurônio vencedor. A medida de novidade é fixada numa escala de 0 a 1 e, comumente, tem um valor bem pequeno e próximo de zero, i.e., $\delta \ll 1$. À proporção que um neurônio vence a competição ou participa da vizinhança topológica do neurônio vencedor, o peso de sua conexão sináptica w^h tende a zero, que é o ponto máximo de saturação; caso contrário, tende a 1, que é o estado de não habituação. Portanto, o grau de novidade de um padrão é proporcional ao peso da conexão w_v^h do neurônio vencedor. Dessa forma, quanto maior o valor fixado para o parâmetro de novidade, mais rapidamente os neurônios serão inseridos na rede.

O teste da Equação 5.32 pode ser interpretado da seguinte maneira: a criação de um novo neurônio somente é justificada se a similaridade entre o neurônio vencedor e o padrão de entrada for menor que o critério de plasticidade e, ao mesmo tempo, o padrão de entrada não for tão novo como indicado pelo critério de novidade. Como regra geral, a rede cresce até um determinado *critério de parada* ser satisfeito, que pode ser o tamanho da rede em número de neurônios (n_{max}) ou de conexões sinápticas (c_{max}).

Se o critério de plasticidade, ρ , e o critério de novidade, δ , forem satisfeitos uma nova unidade de índice r é inserida entre o neurônio vencedor v e o padrão espaço-temporal \mathbf{z} . A Figura 5.6 ilustra o procedimento de inserção de um novo neurônio na rede neural.

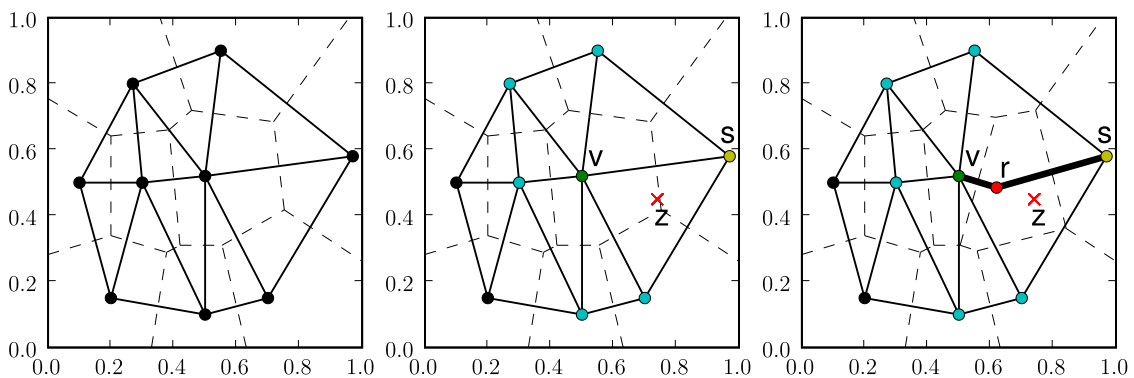


Figura 5.6: Inserção de um novo neurônio na rede neural proposta. À esquerda, configuração da rede antes da inserção. No centro, um padrão espaço-temporal \mathbf{z} é apresentado à rede e o primeiro e segundo neurônios vencedores, v e s , respectivamente, são selecionados. À direita, um novo neurônio de índice r é adicionado na rede entre o neurônio vencedor v e o padrão espaço-temporal \mathbf{z} e a conexão existente entre v e s é realocada.

O vetor de pesos sinápticos \mathbf{w}_r^z da nova unidade r simplesmente é interpolado a partir do vetor de pesos sinápticos \mathbf{w}_v^z da unidade vencedora v e do padrão espaço-temporal \mathbf{z} .

Inserindo o neurônio r entre o neurônio vencedor e o padrão de entrada faz com a rede tenha uma convergência mais estável, em relação à inclusão do neurônio r exatamente sobre o padrão de entrada que ocasionou o erro. O peso da conexão sináptica w_r^h da nova unidade r é feito igual a 1, ou seja, a unidade r é inicializada no estado de não habituação:

$$\mathbf{w}_r^z(t+1) = \frac{\mathbf{w}_v^z(t) + \mathbf{z}(t)}{2} \quad (5.33)$$

$$w_r^h(t+1) = 1.0 \quad (5.34)$$

A nova unidade r é conectada às duas unidades vencedoras v e s e a conexão sináptica existente entre as unidades vencedoras v e s é removida:

$$\mathcal{C}(t+1) = \mathcal{C}(t) \cup \{(n_r, n_v), (n_r, n_s)\} \quad (5.35)$$

$$\mathcal{C}(t+1) = \mathcal{C}(t) / \{(n_v, n_s)\} \quad (5.36)$$

A remoção de um neurônio pode ocorrer em duas situações⁷: (i) quando o neurônio não possuir nenhuma conexão sináptica ($N_i = \emptyset$); ou (ii) em virtude de o neurônio estar posicionado em uma região de baixíssima densidade probabilística, isto é, quando for considerado não representativo da distribuição de pontos no problema ($w_i^h(t) \approx 1$). Matematicamente, um neurônio i é eliminado da rede se a seguinte expressão for satisfeita:

$$\mathcal{A}(t+1) = \begin{cases} \mathcal{A}(t) / \{n_i\} & \text{se } N_i = \emptyset \text{ ou } w_i^h(t) \approx 1 \\ \mathcal{A}(t) & \text{caso contrário} \end{cases} \quad (5.37)$$

No segundo caso descrito, um neurônio é considerado supérfluo caso não vença a competição por um longo período de tempo. Esse período é determinado de forma implícita através das taxas de sensibilização e de habituação (Eq. 5.14). Considera-se um neurônio supérfluo se o valor de sua sinapse de habituação estiver próximo de 1. A idéia por trás do mecanismo de remoção por habituação é punir os neurônios que nunca vencem a competição e capacitar a rede para lidar com uma distribuição de dados não-estacionários.

A Figura 5.7 exemplifica como a rede neural proposta realiza a remoção de neurônios.

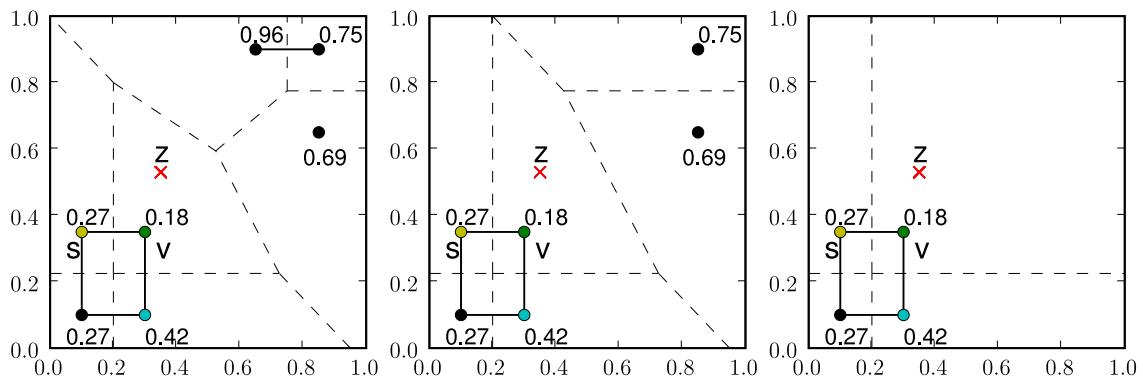


Figura 5.7: Remoção de neurônios na rede neural proposta. Os números reais indicam os valores das sinapses de habituação de cada neurônio. À esquerda, configuração da rede antes da remoção de neurônios. No centro, os neurônios que têm o valor da sinapse de habituação próximo de 1 são removidos da rede (e.g., $w_i^h \geq 0.95$). À direita, os neurônios que não possuem nenhuma conexão sináptica são removidos da rede (i.e., $N_i = \emptyset$).

⁷O neurônio r somente é elegível para eliminação no instante de tempo subsequente ao qual foi inserido.

5.3.6 Processo de inclusão, adaptação e remoção de arestas

O processo de inclusão, adaptação e remoção de arestas é similar à técnica utilizada no método de Aprendizado Hebbiano Competitivo (MARTINETZ, 1993; MARTINETZ; SCHULTEN, 1994). Depois de encontrar as duas unidades mais próximas ao padrão de entrada apresentado à rede, v e s , respectivamente, o algoritmo verifica a existência de uma conexão entre elas e se, por ventura, a conexão não existir o algoritmo se encarrega de criá-la:

$$\mathcal{C}(t+1) = \begin{cases} \mathcal{C}(t) \cup \{(n_v, n_s)\} & \text{se } n_s \in N_v \\ \mathcal{C}(t) & \text{caso contrário} \end{cases} \quad (5.38)$$

Cada conexão sináptica da rede tem associada a si um contador que controla a sua *idade*. Em outras palavras, o contador de idade de uma conexão regula a sua *utilidade* para a rede neural. A cada passo de tempo, o valor do contador dos neurônios da vizinhança do vencedor é incrementado em uma unidade e esse valor somente é zerado caso os neurônios que a conexão conecte sejam considerados os vencedores. A idade da conexão entre o neurônio vencedor v e um neurônio de índice i é atualizada da seguinte forma:

$$\text{idade}_{vi}(t+1) = \begin{cases} 0 & \text{se } i = s \\ \text{idade}_{vi}(t) + 1 & \text{se } i \neq s \text{ e } n_i \in N_v \\ \text{idade}_{vi}(t) & \text{caso contrário} \end{cases} \quad (5.39)$$

Uma conexão da rede neural é eliminada se o seu contador de idade atingir certo limiar (i_{max}). Portanto, a idéia é priorizar as conexões novas em detrimento de antigas. Esta técnica visa atualizar a cada instante a estrutura da rede, mantendo apenas conexões que sejam úteis sob o ponto de vista da precisão e preservação da topologia da rede:

$$\mathcal{C}(t+1) = \begin{cases} \mathcal{C}(t) / \{(n_i, n_j)\} & \text{se } \text{idade}_{ij}(t) > i_{max} \\ \mathcal{C}(t) & \text{caso contrário} \end{cases} \quad (5.40)$$

A Figura 5.8 elucida como a rede neural proposta faz o ajuste de conexões sinápticas.

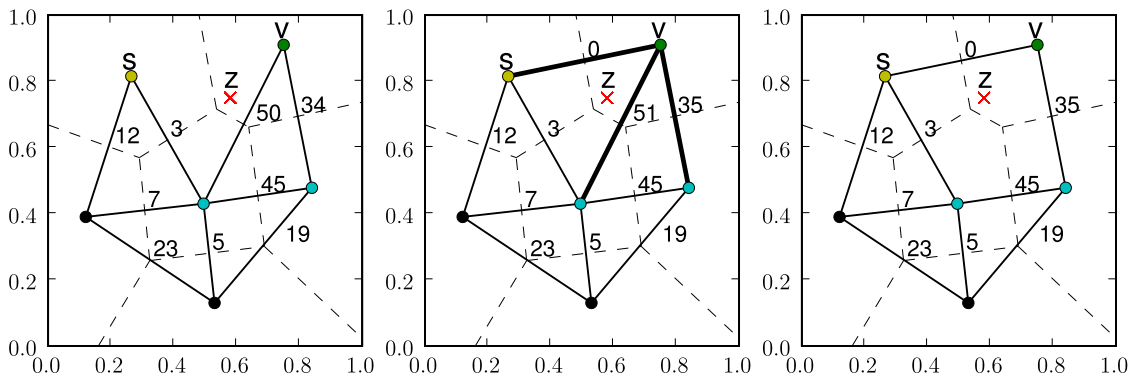


Figura 5.8: Inserção, atualização e remoção de conexões sinápticas na rede neural proposta. Os números inteiros representam os contadores de idade de cada conexão sináptica da rede. À esquerda, configuração da rede em um instante de tempo arbitrário. No centro, uma nova conexão é adicionada na rede entre os neurônios vencedores v e s e o contador de idade dessa conexão é feito igual a zero. Ademais, as idades de todas as conexões existentes entre o neurônio vencedor v e a sua vizinhança topológica N_v , exceto neurônio s , são incrementadas em uma unidade. À direita, as conexões sinápticas que possuem uma idade maior que o limiar i_{max} são removidas da rede (e.g, $i_{max} = 50$).

5.4 Resumo do algoritmo

Para entender melhor o método proposto, no Algoritmo 5.1 pode-se observar, de uma forma geral, todos os passos envolvidos na aprendizagem da rede neural.

Algoritmo 5.1: Algoritmo da rede neural proposta.

```

1 Determinar os parâmetros  $n_{max}$ ,  $c_{max}$ ,  $i_{max}$ ,  $\rho$ ,  $\delta$ ,  $\mu$ ,  $\eta$ ,  $\alpha$ ,  $\beta$  e  $\gamma$ ;
2 Inicializar o arranjo  $\mathcal{A}$  com apenas dois neurônios,  $n_1$  e  $n_2$  (Eq. 5.19);
3 Inicializar o conjunto de conexões sinápticas  $\mathcal{C}$  (Eq. 5.20);
4 enquanto critério de parada não for satisfeito faça
5     Escolher um padrão  $\mathbf{x}$  conforme a distribuição uniforme  $p(\mathbf{x})$ ;
6     Compor um padrão espaço-temporal  $\mathbf{z}$  com o uso de uma memória (Eq. 5.10);
7     Localizar dois neurônios,  $v$  e  $s$ , mais próximos ao padrão  $\mathbf{z}$  (Eqs. 5.21 e 5.22);
8     Criar e inicializar uma conexão entre os neurônios  $v$  e  $s$  (Eqs. 5.38 e 5.39);
9     para cada neurônio  $i$  no arranjo  $\mathcal{A}$  faça
10        Ajustar a sinapse de habituação  $w_i^h$  de acordo com  $S_i$  (Eqs. 5.14 e 5.31);
11    fim
12    se critério de plasticidade  $\rho$  e critério de novidade  $\delta$  forem satisfeitos então
13        Adicionar uma nova unidade de índice  $r$  à rede (Eq. 5.32);
14        Interpolar o vetor de pesos sinápticos  $\mathbf{w}_r^z$  de  $r$  a partir de  $\mathbf{w}_v^z$  e  $\mathbf{z}$  (Eq. 5.33);
15        Atribuir ao peso da conexão sináptica  $w_r^h$  de  $r$  o valor 1.0 (Eq. 5.34);
16        Inserir conexões de  $r$  até  $v$  e de  $r$  até  $s$  (Eq. 5.35);
17        Remover conexão original entre  $v$  e  $s$  (Eq. 5.36);
18    senão
19        Ajustar o vetor de pesos sinápticos de  $v$  (Eqs. 5.26, 5.28 e 5.29);
20        para cada neurônio  $i$  vizinho de  $v$  faça
21            Ajustar o vetor de pesos sinápticos de  $i$  (Eqs. 5.27, 5.30 e 5.25);
22            Incrementar em uma unidade a idade da conexão entre  $v$  e  $i$  (Eq. 5.39);
23        fim
24    fim
25    Remover conexões sinápticas com idade maior que  $i_{max}$  (Eq. 5.40);
26    Remover neurônios isolados e supérfluos (Eq. 5.37);
27 fim

```

5.5 Notação matemática

A seguir, apresenta-se uma listagem completa da notação matemática adotada no presente trabalho. Por questões de conveniência, a notação matemática está sintetizada em três tabelas. A Tabela 5.1 apresenta um resumo das funções da rede neural proposta.

Tabela 5.1: Resumo das funções da rede neural proposta.

Função	Argumento	Imagem	Descrição	Equação	Figura
$\Phi(\cdot)$	(\mathbf{x}, \mathbf{y})	$(\mathbf{w}^x, \mathbf{w}^y)$	Função do mapa de características	5.2	5.4
$\Psi(\cdot)$	\mathbf{x}	\mathbf{y}	Função da unidade de memorização	5.5	5.1
$\Theta(\cdot)$	\mathbf{S}	\mathbf{w}^h	Função da unidade de habituação	5.13	5.2

A Tabela 5.2 apresenta um resumo das variáveis da rede neural proposta.

Tabela 5.2: Resumo das variáveis da rede neural proposta.

Variável	Descrição	Equação
\mathcal{Z}	Espaço de entrada contínuo (padrões espaço-temporais)	5.1
$\mathcal{X} \subset \mathcal{Z}$	Subespaço de entrada sensorial (padrões espaciais)	5.3
$\mathcal{Y} \subset \mathcal{Z}$	Subespaço de entrada contextual (padrões temporais)	5.8
\mathcal{A}	Espaço de saída discreto (neurônios)	5.11
$\mathcal{C} \subset \mathcal{A} \times \mathcal{A}$	Conjunto de conexões sinápticas da rede (arestas)	5.15
$\mathbf{x}(t) \in \mathbb{R}^m \subset \mathcal{X}$	Vetor de entrada	5.4
$\mathbf{y}(t) \in \mathbb{R}^m \subset \mathcal{Y}$	Vetor de contexto	5.9
$\mathbf{z}(t) \in \mathbb{R}^{2m} \subset \mathcal{Z}$	Vetor espaço-temporal	5.10
$\mathbf{w}_i^x(t) \in \mathbb{R}^m$	Vetor de pesos de entrada do i -ésimo neurônio	5.12
$\mathbf{w}_i^y(t) \in \mathbb{R}^m$	Vetor de pesos de contexto do i -ésimo neurônio	5.12
$\mathbf{w}_i^z(t) \in \mathbb{R}^{2m}$	Vetor de pesos espaço-temporal do i -ésimo neurônio	5.12
$w_i^h(t) \in [0, 1]$	Peso sináptico de habituação do i -ésimo neurônio	5.14
$S_i(t) \in \mathbb{R}^1$	Estimulação externa recebida pelo i -ésimo neurônio	5.31
$N_i \subset \mathcal{A}$	Vizinhança topológica centrada no i -ésimo neurônio	5.18
$h_{ij}(t) \in \{0, 1\}$	Função vizinhança centrada no i -ésimo neurônio	5.25
$\text{idade}_{ij}(t) \in \mathbb{N}^1 \cup \{0\}$	Idade da conexão sináptica entre os neurônios i e j	5.17
$n_v(t) \in \mathcal{A}$	Neurônio de índice v considerado primeiro vencedor	5.21
$n_s(t) \in \mathcal{A}$	Neurônio de índice s considerado segundo vencedor	5.22
$n_r(t) \in \mathcal{A}$	Neurônio de índice r adicionado à rede neural	5.32
$\epsilon_v(t) \in [0, 1]$	Taxa de aprendizagem do neurônio vencedor v	5.28
$\epsilon_n(t) \in [0, 1]$	Taxa de aprendizagem da vizinhança topológica N_v	5.30
$d(t)$	Distância euclidiana normalizada entre $\mathbf{z}(t)$ e $\mathbf{w}_v^z(t)$	5.29

A Tabela 5.3 apresenta um resumo dos parâmetros livres da rede neural proposta.

Tabela 5.3: Resumo dos parâmetros da rede neural proposta.

Parâmetro	Descrição	Equação	Valor padrão
n_{max}	Número máximo de neurônios (nodos) permitido	5.11, 5.32	300
c_{max}	Número máximo de conexões (arestas) permitido	5.15, 5.35, 5.38	900
i_{max}	Limiar de idade aceitável para as conexões (arestas)	5.40	50
ρ	Critério de plasticidade	5.32	0.7
δ	Critério de novidade	5.32	0.1
μ	Constante de integração da unidade de memória	5.6	0.7
η	Constante de integração da unidade de memória	5.7	0.7
α	Taxa de sensibilização da unidade de habituação	5.14	0.01
β	Taxa de habituação da unidade de habituação	5.14	0.9
γ	Razão entre as taxas de habituação e sensibilização	5.14	0.7

5.6 Análise paramétrica

O comportamento da unidade de memorização Ψ é parametrizável, através de μ e η (vide Eqs. 5.6 e 5.7). De acordo com os valores definidos para μ e η , a unidade realiza

uma *derivação* e/ou *integração* da informação de entrada. Por um lado, μ determina a taxa de subtração do sinal de entrada por sua integração, isto é, μ funciona como um diferenciador. Por outro, η especifica o grau de influência das entradas anteriores sobre a entrada atual, ou seja, η funciona como um integrador. Na Seção 4.1.2.4 é feita uma análise sobre a influência dos parâmetros μ e η no funcionamento da memória.

O comportamento da unidade de habituação Θ é parametrizável, através de α , β e γ (vide Eq. 5.14). Conforme os valores estipulados para os parâmetros α , β e γ , a unidade realiza uma *habituação* e/ou *sensibilização* da informação de entrada. Por um lado, o parâmetro α determina a taxa de recuperação da habituação, isto é, a velocidade com que a unidade se aproxima do estado de não habituação (valor 1). Por outro lado, o parâmetro β define a taxa de habituação, ou seja, a velocidade com que a unidade se aproxima do ponto máximo de saturação (valor zero). Na Seção 4.1.2.5 é feita uma análise sobre a influência dos parâmetros α , β e γ no funcionamento da unidade.

O critério de plasticidade, ρ , e o critério de novidade, δ , estão relacionados com a inserção de novos neurônios na rede neural (vide Eq. 5.32). Na prática, o valor do parâmetro δ determina o momento em que novos neurônios são adicionados à rede e o valor do parâmetro ρ afeta diretamente o número de neurônios inseridos na rede. Utilizando $\rho = 0$, nenhum neurônio será criado. Com ρ próximo do valor 1, os neurônios criados não serão representativos e a rede neural terá dificuldades em criar *generalizações*. Os valores ideais para os parâmetros ρ e δ estão relacionados com a distância *intra-cluster*, que define a distância entre os padrões contidos dentro de um mesmo agrupamento, e a distância *inter-cluster*, que define a distância entre os centróides de cada agrupamento. A distância *intra-cluster* deve ser pequena o suficiente para obter padrões similares no mesmo agrupamento. A distância *inter-cluster* deve ser suficientemente grande para separar os agrupamentos criados pela rede neural. De forma geral, a busca por valores adequados é guiada por tentativa-e-erro, após alguma experimentação inicial com os dados e a rede.

A relação entre as distâncias *intra-cluster* e *inter-cluster* pode ser observada através de técnicas de *partição*. A idéia de uma técnica de partição é dividir o espaço de dados em regiões por meio de algum critério geométrico. Cada região representa então uma categoria ou classe da distribuição de dados de entrada. A classificação de um determinado vetor de entrada simplesmente é dada pela região ocupada. A forma mais tradicional de partição é o *diagrama de Voronoi*, que consiste em um método de extrema utilidade em reconhecimento de padrões. O diagrama de Voronoi é comumente gerado a partir do método da *triangulação de Delaunay* e tem como propósito delimitar as áreas de categorias ou classes numa representação geométrica de suas distribuições. A Figura 5.9 representa graficamente uma triangulação de Delaunay inicial e o diagrama de Voronoi resultante.

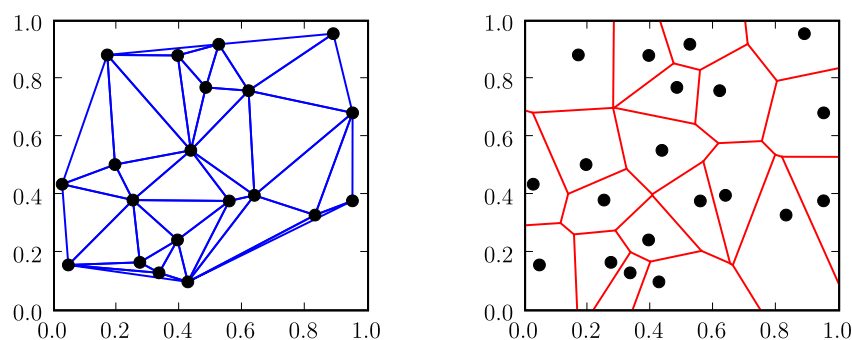


Figura 5.9: À esquerda a triangulação de Delaunay e à direita o diagrama de Voronoi resultante para uma distribuição arbitrária de 20 pontos.

Durante a aprendizagem da rede neural proposta, cada vez que um vetor de pesos é modificado, a região de Voronoi daquele neurônio e dos seus vizinhos imediatos também varia. Como consequência da correção de pesos de um neurônio, vários padrões podem passar de uma região de Voronoi para outra adjacente. A estabilidade na representação da rede ocorre quando as classes de Voronoi convergem para uma solução estável.

5.7 Inspiração biológica

Uma discussão pormenorizada sobre as potenciais implicações biológicas da rede neural proposta não é, obviamente, o objetivo do presente trabalho. Contudo, considerando-se que parte dos avanços na área da inteligência artificial é fruto da tentativa de se desvendar os segredos do sistema nervoso, não custa mencionar, ainda que de forma superficial, algumas intrigantes relações do modelo neural com evidências biológicas modernas. É importante observar aqui que, ainda que exista uma inspiração biológica, não há qualquer pretensão na modelagem exata de propriedades neurais. Com efeito, pode-se ver que a proposta do algoritmo viola suposições usualmente feitas para redes neurais biológicas.

A rede neural proposta faz parte de uma família de modelos conexionistas auto-organizáveis, incorporando todos os mecanismos que são básicos para a auto-organização: *competição*, *cooperação* e *auto-amplificação*. O conceito de auto-organização de sistemas tem suas primeiras formulações matemáticas propostas nos primeiros anos do advento da Cibernética⁸ (ASHBY, 1970). Ross Ashby é citado como sendo provavelmente a primeira pessoa a descrever o sistema nervoso como auto-organizado. Em seu livro intitulado *Design for a Brain: The Origin of Adaptive Behavior*, Ashby (1960) preocupou-se com a noção básica de que o comportamento adaptativo não é inato mas, ao invés disso, aprendido, e que por meio do aprendizado o comportamento de um animal ou sistema muda em geral para melhor. O livro destaca que aspectos dinâmicos de um organismo têm forte analogia com o conceito de estabilidade de uma máquina. Hoje em dia, há muitas teorias que apóiam a hipótese de auto-organização do sistema nervoso (HAYKIN, 2001, p. 430–432). A questão básica consiste em compreender como a ação de vários elementos simples (neurônios), em uma configuração inicial desordenada, dá margem ao surgimento espontâneo (sem regras pré-programadas) de padrões coerentes de comportamento.

A rede neural proposta tem como fonte de inspiração idéias derivadas dos mapas corticais do cérebro. Com o atual conhecimento da neurociência é possível observar *neurônios localmente ajustados* em inúmeras regiões do sistema nervoso, isto é, neurônios que somente são ativados por uma faixa específica de valores de estímulo. Como exemplo, pode-se citar as células de orientação do córtex visual, que respondem seletivamente a estímulos que são locais tanto na posição da retina quanto no ângulo da orientação do objeto (LENT, 2002, p. 293). Em geral, populações de células seletivas são arranjadas em *mapas corticais* e a faixa de valores que ativa cada célula costuma ser restrita à sua *posição* no mapa. Erwin e colaboradores (1995) comprovaram que um mapa de características auto-organizável é capaz de explicar a formação de mapas computacionais no córtex visual primário do macaco. Admitindo-se certas suposições, os pesquisadores demonstraram que a *aprendizagem hebbiana* resulta em padrões espaciais de orientação e dominância ocular que são notavelmente similares àqueles encontrados no macaco.

⁸A palavra *Cibernética* (do grego, *kybernetiké*, piloto, no sentido utilizado por Platão para qualificar a ação da alma) foi cunhada por Norbert Wiener em 1948 como o nome de uma nova ciência que visava à compreensão dos fenômenos naturais e artificiais através do estudo dos processos de comunicação e controle nos seres vivos, nas máquinas e nos processos sociais (BITTENCOURT, 2001, p. 40).

A unidade de memória de curto prazo pode ser considerada biologicamente plausível se for levada em consideração a possível influência dos diferentes neurotransmissores sobre a atividade de um neurônio (MOSER, 2004, p. 67). Sabe-se que cada neurotransmissor presente no sistema nervoso pode desencadear reações de curta ou longa duração, ou seja, a informação computada em um neurônio não se resume aos estímulos elétricos de cada instante, mas também depende de processos químicos que ocorrem no interior do neurônio e que carregam informação do passado. Traçando um paralelo com as Equações 5.6 e 5.7, seriam suficientes apenas dois tipos de neurotransmissores para implementar a unidade de memória sugerida por Moser (2004). O efeito de dois neurotransmissores sobre um neurônio biológico é similar às duas integrações temporais sobre a memória.

A unidade de habituação busca inspiração no fenômeno natural da habituação. A habituação é uma forma simples de aprendizado tipicamente observada em organismos biológicos. Do ponto de vista da neurociência, a habituação ocorre através de mudanças estruturais nas conexões sinápticas entre os neurônios, de tal forma que a resposta elétrica a estímulos repetitivos e irrelevantes seja paulatinamente reduzida até ser nula. Ou seja, a habituação funciona como um *filtro de novidade* para o organismo. Como exemplo, pode-se citar os movimentos reflexos defensivos da aplísia⁹ (“lebre-do-mar”) (LENT, 2002, p. 149). À medida que o sifão da aplísia é estimulado, verifica-se um movimento de retirada de sua brânquia. No entanto, com a repetição, o animal se habitua a esse estímulo inofensivo, e acaba por não retrair mais a sua brânquia.

5.8 Discussão

Neste capítulo, a estrutura dos elementos e a dinâmica dos processos da rede neural proposta foram apresentados e discutidos. Nesse momento, talvez seja apropriado refletir sobre estes elementos e questionar sobre as razões que levaram à sua adoção na rede neural. A discussão é conduzida considerando os seguintes itens: unidade de memorização, unidade de habituação e mecanismo de crescimento e poda de neurônios e conexões.

A formulação da rede neural proposta é baseada no processamento de dados *espaço-temporais*, isto é, dados que estão relacionados de forma espacial e que possuem uma correlação temporal entre si. Em geral, há duas formas de incluir o tempo na operação de uma rede neural: através de laços de realimentação (recorrência) ou por meio de memórias de curto prazo. Hoje em dia, já foi comprovado que redes neurais artificiais que utilizam laços de realimentação não relacionam dados mais distantes que dez instantes de tempo, o que limita consideravelmente a gama de problemas que pode ser solucionada por tais redes (HOCHREITER; SCHMIDHUBER, 1997). Por essa razão, a memória de curto prazo é a peça-chave para incorporar o tempo no funcionamento de uma rede neural. No presente trabalho, a memória de curto prazo é aplicada sobre o sinal de entrada e não à rede propriamente dita, que permanece estática. Basicamente, a memória captura a informação temporal contida no sinal de entrada e o neurônio insere esta informação em seus próprios pesos sinápticos. Desta forma, existe uma evidente separação de papéis: a rede neural é responsável pela não-linearidade e a memória de curto prazo é responsável pelo tempo.

⁹A aplísia é um animal invertebrado, similar a um caramujo marinho sem concha e muito abundante nas costas do Pacífico Norte, que já foi amplamente utilizada para estudos de plasticidade sináptica em pesquisas científicas. A aplísia possui um sistema nervoso simples e um repertório comportamental reduzido, como movimentos de encolhimento da cauda e da cabeça, e a retirada da brânquia e do sifão. O sifão é uma protrusão carnosa, situada acima da brânquia, que expõe água do mar e restos alimentares.

A memória de curto prazo utilizada é uma sugestão de Moser (2004) e tem como função realizar uma derivação e/ou integração da informação de entrada. A diferenciação é útil para eliminação da redundância e da correlação espaço-temporal, pois indica as mudanças do sinal ocorridas no passado, e a integração é útil para eliminação de ruído, pois representa o passado sob a forma de uma média. O funcionamento da memória exige poucos recursos computacionais e pode ser de grande valia em problemas onde a solução ideal depende de uma representação baseada em mudanças e não em valores absolutos da entrada. Segundo Moser (2004, p. 93), problemas temporais que possuem alta correlação temporal (e.g., sinais sensoriais de robôs), com ou sem ruído, provavelmente podem se aproveitar das informações geradas pela memória. Em contrapartida, problemas com baixa correlação temporal ou simbólicos (e.g., cadeias de DNA) provavelmente se agravariam com o uso da memória. Em geral, a memória apresenta grande potencialidade na área da robótica, onde é necessário tratar diretamente o fluxo sensorio-motor de robôs de forma rápida e econômica. A representação gerada pela memória pode ser valiosa para previsão e identificação de eventos importantes, tais como: aproximação de objetos, mudanças entre corredor e curvas, bifurcações e oclusão de objetos.

Até recentemente, a definição apropriada das dimensões de uma rede neural para uma determinada tarefa geralmente envolvia a experiência do projetista em implementar procedimentos de tentativa-e-erro. A construção automática da arquitetura da rede está se tornando parte do processo de treinamento, através da exploração mais eficiente de informações disponíveis durante a aprendizagem. As redes neurais construtivas representam uma das alternativas existentes para esta problemática. No entanto, um dos pontos críticos é determinar quando e em que posição uma nova unidade deve ser inserida. Na maioria das redes construtivas, a velocidade de inserção é controlada por um parâmetro λ . O processo de encontrar um valor apropriado de λ não é uma tarefa fácil e nem trivial. Isso depende de múltiplos fatores, tais como a complexidade da distribuição e o volume de dados de entrada. Por um lado, se o valor de λ for relativamente pequeno, é possível criar uma maior quantidade de unidades, porém o tempo para adaptar as novas unidades à estrutura da rede existente é significativamente reduzido, gerando assim redes visivelmente deformadas. Por outro lado, se o valor de λ for grande, a quantidade de unidades criadas em um período de tempo é menor, entretanto, o tempo de treinamento da rede é consideravelmente maior. De maneira geral, o parâmetro λ deve ser definido de acordo com a distribuição do conjunto de amostras do espaço de entrada. Contudo, em muitos casos, pouca ou nenhuma informação está disponível acerca do tamanho e da distribuição desse conjunto. Logo, a definição de um valor apropriado para λ é um fator que depende, em grande parte, da experiência empírica do projetista da rede.

Diferentemente da maioria das redes construtivas que necessitam especificar previamente uma taxa de crescimento λ , na rede neural proposta o procedimento de inserção de novos neurônios é realizado de forma automática e em qualquer instante de tempo, sem que isto obrigue a reinicialização do processo de aprendizagem. Esse procedimento retira o ônus do projetista em ter que especificar *a priori* as dimensões da rede neural, além de tornar possível resolver tarefas que exijam aprendizado em tempo real, como, por exemplo, *tarefas de horizonte infinito*. Sob o ponto de vista de utilidade em problemas dinâmicos, a rede neural proposta oferece uma boa solução para o *dilema da estabilidade-plasticidade*, apontado por Stephen Grossberg em 1976 (HAYKIN, 2001, p. 30), e faz uso eficiente de memória se comparada com outras redes equivalentes (i.e., economia de neurônios e conexões). O mecanismo que governa o crescimento da rede adiciona um novo neurônio se e somente se dois critérios de vigilância forem satisfeitos: o critério de

plasticidade, ρ , e o critério de novidade, δ . Não obstante, é tarefa do projetista da rede neural definir valores adequados para esses dois parâmetros livres.

O procedimento de remoção automática de neurônios visa aumentar a capacidade de generalização da rede neural, reduzir os requisitos computacionais e permitir a manipulação de dados não-estacionários. A remoção de um neurônio pode ocorrer em duas ocasiões: (i) quando o neurônio não possuir nenhuma conexão sináptica; ou (ii) quando o neurônio estiver posicionado em uma região de baixíssima densidade probabilística, ou seja, quando o neurônio for considerado não representativo da distribuição de pontos no problema. No segundo caso, considera-se que um neurônio é não-representativo (ou supérfluo) quando o valor de sua sinapse de habituação estiver na iminência de atingir o estado de não habituação (valor 1). Ou seja, a idéia por trás do mecanismo de remoção por habituação é punir os neurônios que nunca vencem a competição.

Considerações similares para remoção de neurônios em redes auto-organizáveis já foram realizadas por DeSieno (1988) e Fritzke (1997a). DeSieno propôs usar uma forma de memória para rastrear as atividades cumulativas dos neurônios individuais. Em particular, é incluído um mecanismo de “consciência” para regular o *bias* do processo de aprendizagem de uma rede de Kohonen convencional. Isto é feito de tal maneira que cada neurônio, independentemente de sua localização na grade, tem a chance de vencer a competição com uma probabilidade próxima à ideal de $1/n$, onde n é o número total de neurônios. Apesar deste método funcionar perfeitamente bem em uma rede de Kohonen convencional, o mesmo não se verificou em redes construtivas pois, em certas circunstâncias, o mecanismo de consciência acaba por atrapalhar o processo de aprendizagem. Fritzke sugeriu associar um fator de utilidade a cada neurônio da rede. O objetivo é remover os neurônios que possuem um fator de utilidade abaixo de um coeficiente mínimo aceitável. No entanto, esse método é computacionalmente oneroso, visto que cada neurônio deve armazenar uma informação local de erro e de utilidade.

O algoritmo de aprendizagem proposto opera em apenas uma única fase, denominada *fase de aprendizagem*. Em outras palavras, não existe uma fase distinta entre treinamento e atuação. Isso só é possível pois todos os parâmetros livres da rede neural não são dependentes do tempo. Ou seja, os parâmetros são constantes no tempo, o que elimina a necessidade de definir uma taxa de decaimento, normalmente problemática para aplicações que necessitam de um treinamento continuado. Por essa razão, a aprendizagem realizada pode ser *contínua* e em *tempo real*, e a rede neural pode ser empregada em situações desejáveis de *aprendizado perpétuo* (THRUN; MITCHELL, 1995).

5.9 Considerações do capítulo

Este capítulo propôs um novo modelo de rede neural artificial, denominado GTSOM (do acrônimo em inglês para *Growing Temporal Self-Organizing Map*). O modelo de rede neural artificial é baseado em três pilares fundamentais, a saber:

- *Auto-organização*, por meio da *competição global*, *cooperação local* e *auto-amplificação seletiva*;
- *Representação temporal*, através de uma memória de curto prazo diferenciadora e integradora aplicada externamente à rede; e
- *Aprendizado construtivo*, via inserção e poda de neurônios e conexões sinápticas.

A integração desses três princípios em uma rede neural é inovadora e também muito vantajosa pois potencializa e facilita o desenvolvimento de inúmeras aplicações, como *tarefas de horizonte infinito e de dimensão espaço-temporal*. A Tabela 5.4 apresenta um resumo das principais características do modelo neural proposto.

Tabela 5.4: Resumo das principais características da rede neural proposta.

Aspecto	Característica	Descrição
Auto-organização	Competição	Processo global realizado por um <i>juiz externo</i> . O juiz identifica os neurônios que melhor representam o vetor de entrada conforme uma <i>função de distância euclidiana</i> .
	Cooperação	Processo local baseado numa <i>função de vizinhança</i> . A função de vizinhança determina a cada instante a vizinhança topológica formada em torno do neurônio vencedor.
	Adaptação	Processo de adaptação seletiva confinado à vizinhança do neurônio vencedor. O ajuste é realizado a nível sináptico e computado através da técnica de Berglund e Sitte (2006).
Representação temporal	Topologia	Não-hierárquica. Os neurônios são organizados de forma direta em uma única camada computacional.
	Posição	Externa. A memória é aplicada sobre os itens da seqüência e não à rede propriamente dita, que permanece estática.
	Representação	Representação temporal de origem simbólica (local), ao invés de numérica (esparsa).
	Natureza	Diferenciadores e integradores aplicados na seqüência temporal $\mathbf{x}(t)$. Unidade de memorização de Moser (2004).
Aprendizado construtivo	Arranjo	Rede com topologia dinâmica baseada na estrutura intrínseca do espaço de entrada \mathcal{X} .
	Dimensão	Rede com dimensionalidade variável inicializada com dois neurônios conectados entre si.
	Modo de Inserção	Automático. Neurônios são inseridos através dos critérios de plasticidade e de novidade. Conexões sinápticas são inseridas através da técnica CHL (MARTINETZ, 1993).
	Modo de Poda	Automático. Neurônios são removidos quando se tornam isolados ou supérfluos. Conexões sinápticas são removidas através de um contador de idade.
	Aplicação Alvo	Estacionário (e.g., tarefas de quantização vetorial) e não-estacionário (e.g., tarefas de segmentação temporal).

O próximo capítulo relata os resultados de diferentes experimentos de avaliação aplicados à rede neural proposta, considerando tarefas simples de classificação de padrões e segmentação de seqüências temporais. Para fins de validação da rede, também são apresentadas comparações qualitativas e quantitativas com outros trabalhos similares.

6 RESULTADOS EXPERIMENTAIS

Neste capítulo é relatado um conjunto de experimentos preliminares de avaliação do modelo neural proposto nesta dissertação. Como o objetivo do trabalho é validar o modelo, foram escolhidos experimentos teóricos relativamente simples que pudessem ser implementados e simulados na ferramenta de simulação robótica desenvolvida. No entanto, acredita-se que problemas práticos de maior complexidade, principalmente da robótica, poderiam ser resolvidos pelas mesmas técnicas empregadas neste capítulo.

6.1 Metodologia

A metodologia adotada para realizar as experimentações consiste em inicialmente utilizar simulações em computador para gerar dados e, em um segundo momento, proceder a uma análise qualitativa e quantitativa dos dados gerados, comparando-os a resultados previamente evidenciados na literatura. O interesse principal é determinar as características e peculiaridades do modelo neural proposto, identificar suas principais propriedades e limitações e verificar a influência de seus parâmetros no processo de aprendizagem.

A implementação dos experimentos foi feita dentro de um ambiente próprio de simulação robótica¹ na linguagem de programação Python versão 2.4, compilada no sistema operacional Debian GNU/Linux versão 3.1 com kernel 2.6.15 e executada em um processador Intel Pentium M 750 de 1.86Ghz. Para todos os experimentos descritos a seguir, foram fixados os seguintes parâmetros para o modelo:

- $n_{max} = 300$. O parâmetro n_{max} especifica o número máximo de neurônios (nodos) permitido na rede. Valores próximos a esse são comumente adotados na literatura.
- $c_{max} = 900$. O parâmetro c_{max} estabelece a quantidade máxima de conexões sinápticas (arestas) permitida na rede. Valores próximos a esse são normalmente utilizados na literatura (i.e., $c_{max} = 3 * n_{max}$).
- $i_{max} = 50$. O parâmetro i_{max} define o limiar de idade aceitável para as conexões sinápticas (arestas). Esse é o valor mais amplamente utilizado na literatura e fornece um bom desempenho nas simulações.

Cada um dos testes relatados foi repetido pelo menos dez vezes com diferentes inicializações, exceto onde explicitamente indicado, e selecionado o resultado mais comum. A pequena quantidade de épocas de treinamento escolhida foi suficiente para constatar os resultados, pois o erro de quantização da rede cai rapidamente e estabiliza já

¹O Apêndice A apresenta o ambiente de simulação robótica desenvolvido no âmbito deste trabalho.

nas primeiras épocas de treinamento. A rede é treinada com um vetor de entrada n -dimensional \mathbf{x} , cujos elementos x_1, \dots, x_n estão uniformemente distribuídos na região $\{(0 < x_1 < 1); \dots; (0 < x_n < 1)\}$. Para inicializar a rede, os pesos sinápticos de todos os neurônios são escolhidos de um conjunto aleatório restrito ao intervalo fechado $[0, 1]$.

6.2 Quantização vetorial

Nesta seção são mostradas simulações bastante elementares para ilustrar o comportamento da rede neural proposta e algumas de suas propriedades básicas (e.g., generalização, preservação de topologia) em problemas de classificação de padrões espaciais. Em particular, a rede neural é submetida à tarefa de quantização vetorial. Nesta tarefa, o propósito é discretizar otimamente um espaço contínuo de entrada, de modo que cada vetor de entrada seja atribuído a um vetor protótipo que melhor represente as suas características. Um protótipo é armazenado como pesos sinápticos de um neurônio.

Para a realização desta tarefa, os seguintes valores foram fixados para os parâmetros da rede: $\mu = 1.0$, $\eta = 1.0$, $\alpha = 0.01$, $\beta = 0.9$ e $\gamma = 0.7$. Com $\eta = 1.0$, as duas integrações da unidade de memorização Ψ são anuladas resultando em $\mathbf{y}(t) = 0$, o que faz com que o vetor de contexto calculado pela memória não seja levado em consideração pela rede (vide Figura 6.1). Com $\alpha = 0.01$, $\beta = 0.9$ e $\gamma = 0.7$, a unidade de habituação Θ atinge rapidamente o seu ponto máximo de saturação e se recupera lentamente de um estado qualquer até o ponto inicial de não habituação. Como será visto adiante, a rede tem competência para identificar os mais variados agrupamentos espaciais e tem a capacidade de preservar a topologia subjacente do espaço de entrada. Ao longo da discussão, será analisada a relação entre as distâncias *intra-* e *inter-cluster* através da técnica de Voronoi.

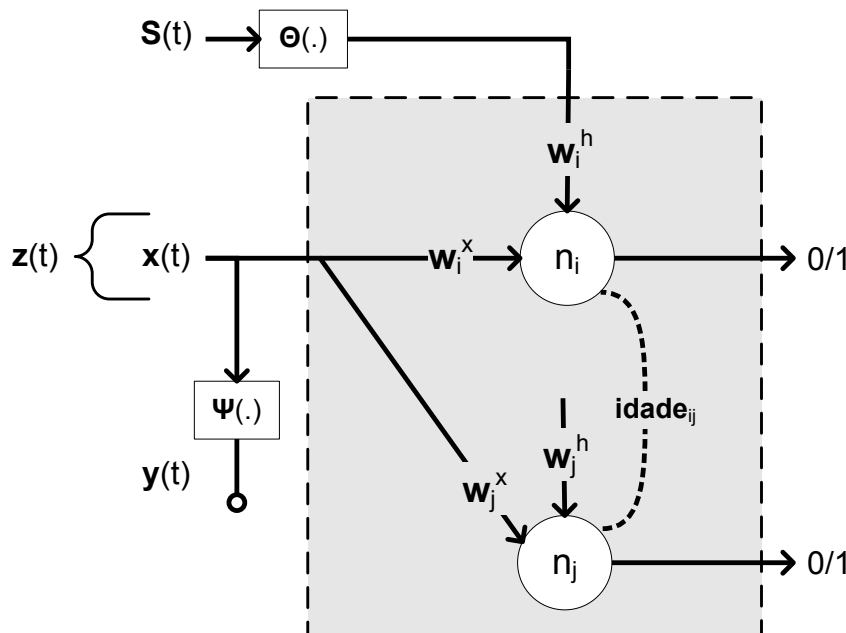


Figura 6.1: Arquitetura simplificada da rede aplicada à tarefa de quantização vetorial.

6.2.1 Distribuição bidimensional de dados

A Figura 6.2a mostra a distribuição uniforme de dados usada para treinar a rede neural. A Figura 6.2b apresenta os valores dos vetores de pesos sinápticos, traçados como pontos

no espaço de entrada. As linhas desenhadas na Figura 6.2b conectam neurônios vizinhos da rede. Cada interseção entre uma linha horizontal e vertical indica a presença de um neurônio. A Figura 6.2c representa graficamente o diagrama de Voronoi resultante. As linhas desenhadas na Figura 6.2c delimitam o espaço de entrada em regiões específicas. Cada região representa uma categoria da distribuição de dados de entrada. A classificação de um determinado vetor de entrada simplesmente é dada pela região de Voronoi ocupada.

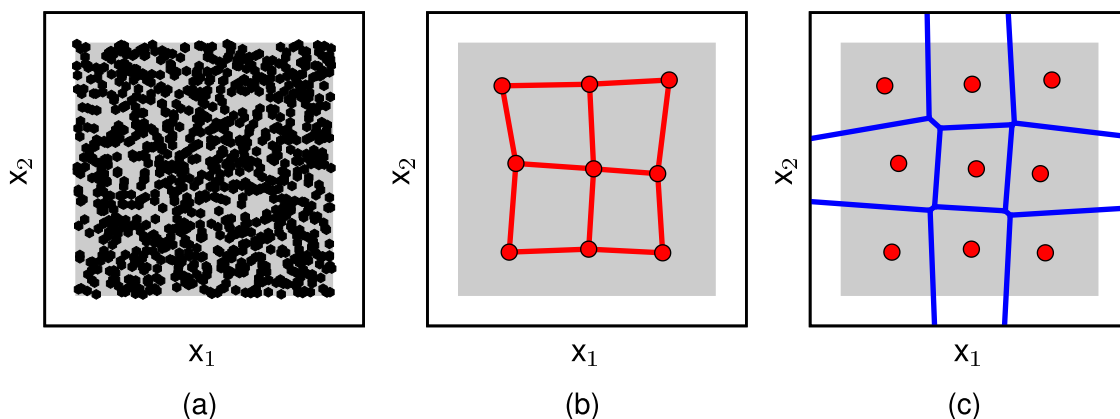


Figura 6.2: Simulação de uma distribuição de dados em forma de quadrado com $\rho = 0.75$ e $\delta = 0.1$. Ao todo são 1000 amostras uniformemente distribuídas dentro da área em cinza claro. (a) Distribuição de densidade dos dados de entrada. (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.

A Figura 6.3 mostra o mapeamento obtido pela rede neural para uma distribuição de dados em forma de círculo. Os padrões de treinamento estão uniformemente distribuídos no interior da região do círculo. Conforme esperado, o mapeamento obtido tende a aproximar a distribuição dos dados de entrada. Nesta simulação foi fixado o valor 0.8 para o parâmetro de plasticidade, ρ . Isto promove a inserção de um maior número de neurônios em relação a simulação anterior. Como pode ser observado na Figura 6.3b, um total de treze neurônios foram incluídos na rede para realizar o mapeamento de características.

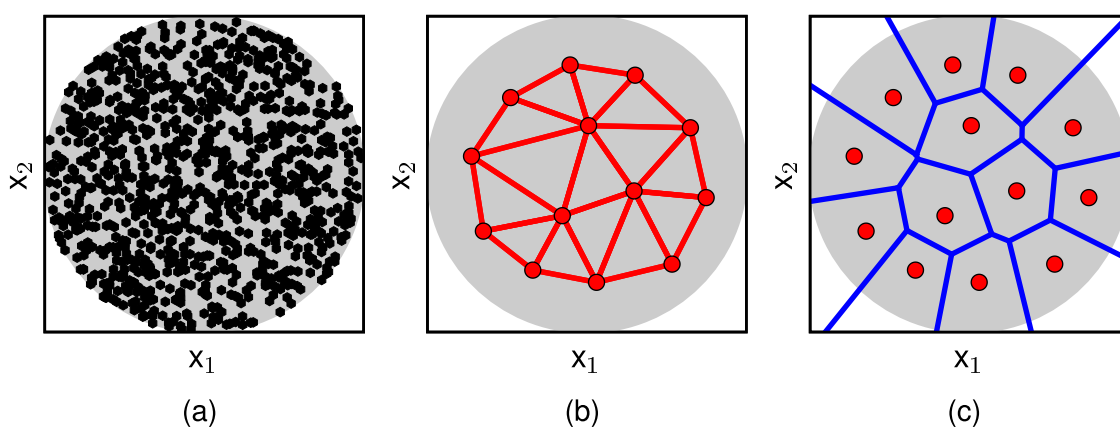


Figura 6.3: Simulação de uma distribuição de dados em forma de círculo com $\rho = 0.8$ e $\delta = 0.1$. No total, são 1000 amostras uniformemente distribuídas sobre a área em cinza claro. (a) Distribuição de densidade dos dados de entrada. (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.

A Figura 6.4 ilustra o mapeamento obtido pela rede neural para uma distribuição de dados em forma de triângulo. Os padrões de treinamento estão uniformemente distribuídos dentro da área triangular. Pode-se perceber que a rede também captura a topologia intrínseca da distribuição uniforme na entrada. Nesta simulação foi fixado o valor de 0.85 para o parâmetro de plasticidade, ρ . Isto causa a inserção de um número ainda maior de neurônios em relação a simulação anterior. No total, dezoito neurônios foram adicionados na rede a fim de realizar o mapeamento de características.

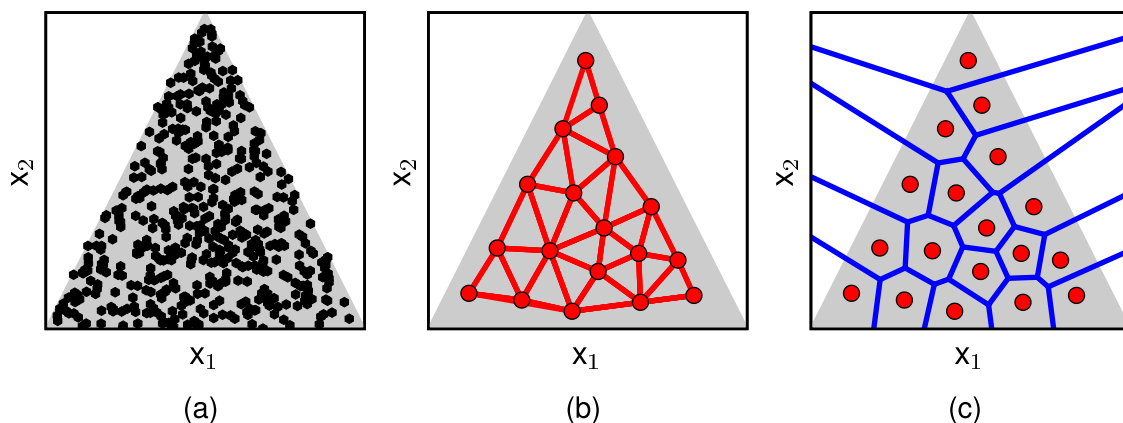


Figura 6.4: Simulação de uma distribuição de dados em forma de triângulo com $\rho = 0.85$ e $\delta = 0.1$. Ao todo são 1000 amostras uniformemente distribuídas ao longo da área em cinza claro. (a) Distribuição de densidade dos dados de entrada. (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.

Em todas as simulações, a rede é inicializada com apenas dois neurônios. Em um primeiro momento, no intuito de preencher o espaço de entrada, novos neurônios são automaticamente adicionados na rede. A inserção de neurônios é realizada conforme os valores fixados para os parâmetros de plasticidade, ρ , e de novidade, δ . Em seguida, os neurônios criados são adaptados à estrutura subjacente do espaço de entrada, de modo a minimizar o erro médio de quantização da rede (vide Eq. 3.11). Adicionalmente, as conexões sinápticas incluídas em demasia são removidas da rede e as restantes são reorganizadas. A distribuição estatística dos neurônios no mapa se aproxima daquela dos vetores de entrada, exceto nas áreas situadas na extremidade do espaço de entrada (e.g., vide Figura 6.2b). A má representação das regiões limites do espaço resulta do *efeito de borda*. Neurônios localizados nas bordas do arranjo possuem vizinhos somente por um lado e, por esse motivo, tendem a ser “puxados” por eles para dentro do espaço.

Durante a fase de aprendizagem, os valores dos pesos sinápticos da rede saem de uma configuração inicial desordenada para uma configuração ordenada. Pode-se fazer aqui uma analogia da evolução do aprendizado da rede com uma *malha elástica* (do inglês *elastic mesh*) que se desdobra à medida que a aprendizagem avança. Em particular, a malha elástica é dinâmica (i.e., com a inclusão e remoção de neurônios) e não tem uma estrutura fixa (e.g., retangular, hexagonal, entre outras). Após um período suficiente de aprendizagem (10 épocas), os valores dos pesos sinápticos e a configuração geométrica das conexões sinápticas se tornam estáveis. A estabilidade na representação da rede ocorre quando o erro médio de quantização da rede é suficientemente pequeno e, deste modo, o valor calculado para a taxa de aprendizagem do neurônio vencedor, ϵ_v , tende a 0.

A rede neural proposta não sofre de um problema que surge ocasionalmente na aplicação do algoritmo de Kohonen convencional. Este problema é representado graficamente

na Figura 6.5 e consiste na criação de um mapa “dobrado” devido a uma falha de ordenação topológica. A criação de um mapa dobrado pode ser vista como uma forma de um *estado metaestável* – mínimo local – do processo de ordenação topológica. Um estado metaestável consiste em uma configuração dos pesos sinápticos da rede com defeito topológico, ou seja, configurações em que relações de vizinhança são violadas. Defeitos topológicos ocorrem, por exemplo, se a largura inicial da função vizinhança é muito pequena ou decresce muito rapidamente. Erwin, Obermayer e Shulten (1992) demonstraram que estados metaestáveis surgem com muito mais frequência quando o algoritmo usa uma função vizinhança não *convexa* como, por exemplo, a função retangular.

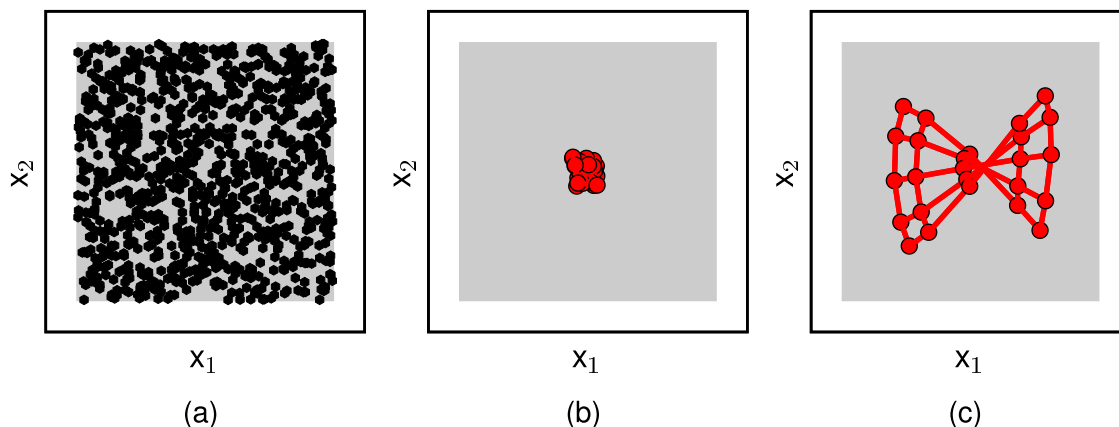


Figura 6.5: Falha de ordenação topológica observada em uma rede de Kohonen convencional com 25 neurônios dispostos regularmente em 5 linhas \times 5 colunas. (a) Distribuição dos dados de entrada (1000 amostras). (b) e (c) Valores iniciais e finais, respectivamente, dos pesos sinápticos da rede. O arranjo de neurônios encontra-se em uma configuração metaestável e, neste caso, a proximidade espacial dos neurônios é violada.

A falha no processo de ordenação topológica não ocorre na rede neural proposta, uma vez que a função vizinhança é dinâmica. A técnica empregada para criar, remover e ajustar as conexões sinápticas é baseada no método de aprendizado hebbiano competitivo (MARTINETZ; SCHULTEN, 1994). Esse método determina que a vizinhança topológica de um neurônio de índice i é o conjunto de neurônios que tem conexão sináptica direta com ele no instante de tempo t , i.e., N_i . Ou seja, a vizinhança não tem uma estrutura fixa e nem é constante durante o processo de aprendizagem.

Apesar da aparente vantagem de utilizar uma vizinhança dinâmica, existe uma grande limitação intrínseca nesta abordagem no que se refere à *redução de dimensionalidade*. Em uma rede de Kohonen convencional, o mapa de características é frequentemente capaz de formar uma representação topológica correta quando a dimensão do espaço de entrada é maior que a dimensão do espaço de saída. Na rede neural proposta, a construção do mapa de características tende a seguir a estrutura intrínseca do espaço de entrada e, por essa razão, o algoritmo *não* realiza uma redução de dimensionalidade. No entanto, as conexões entre neurônios podem criar tanto malhas regulares como diversas figuras geométricas e, por isso, uma maior capacidade de aprendizado é conferida à rede neural.

O procedimento de inserção de novos neurônios na rede é realizado de forma automática e em qualquer instante de tempo, sem que isto obrigue a reinicialização do processo de aprendizagem. O mecanismo que governa o crescimento da rede adiciona um novo neurônio se e somente se dois critérios de vigilância forem satisfeitos: o critério de plasticidade, $0 < \rho < 1$, e o critério de novidade, $0 < \delta < 1$ (vide Figura 6.6). As três

simulações computacionais discutidas até então – distribuição quadrangular, circular e triangular – utilizaram o valor de 0.1 para o parâmetro de novidade e diferentes valores para o parâmetro de plasticidade, a saber: 0.75, 0.8 e 0.85, respectivamente.

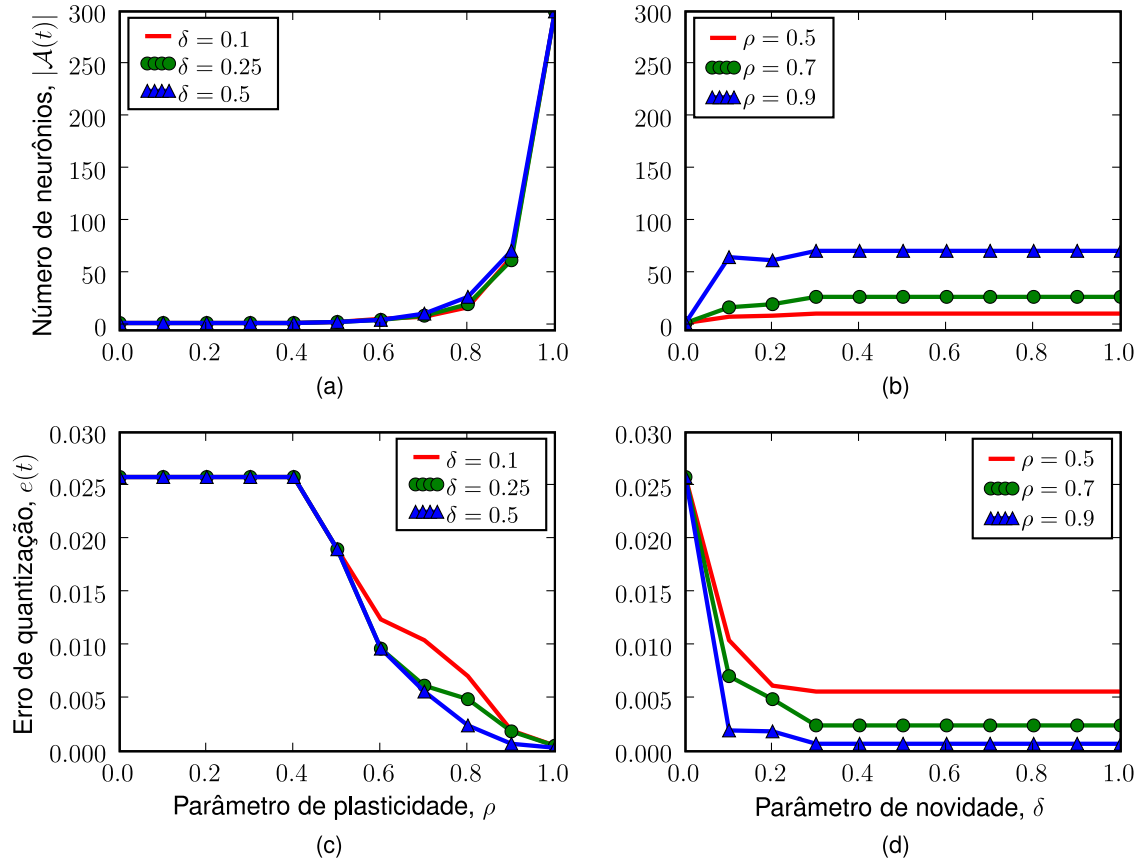


Figura 6.6: (a) Número de neurônios em função do parâmetro de plasticidade. (b) Número de neurônios em função do parâmetro de novidade. (c) Erro médio quadrado em função do parâmetro de plasticidade. (d) Erro médio quadrado em função do parâmetro de novidade.

Nesse momento, talvez seja apropriado refletir sobre a influência dos parâmetros de vigilância na convergência da rede neural. No intuito de ter-se uma noção de quais as repercussões de atribuir um valor pequeno ou grande para esses dois parâmetros, foram realizadas diversas simulações computacionais variando-se os valores do parâmetro de novidade $\delta \in \{0.1, 0.25, 0.5\}$ e do parâmetro de plasticidade $\rho \in \{0.5, 0.7, 0.9\}$. O resultado está sintetizado na Figura 6.6. Cada combinação é treinada com o mesmo conjunto de pesos aleatórios iniciais e o mesmo conjunto de 300 amostras uniformemente distribuídas, de modo que os resultados do experimento possam ser comparados diretamente. O número mínimo de neurônios permitido é preestabelecido em 2. O número máximo de neurônios é governado pelo parâmetro n_{max} e, neste caso, vale 300.

A partir da Figura 6.6 pode-se concluir que o parâmetro de plasticidade, ρ , está diretamente relacionado com o número de neurônios inseridos na rede. Quanto maior o valor fixado para o parâmetro de plasticidade, mais específico será o critério de plasticidade e uma quantidade maior de neurônios será adicionada na rede (i.e., $\rho \rightarrow 1 \therefore |\mathcal{A}| \rightarrow n_{max}$). Em contrapartida, o parâmetro de novidade, δ , não afeta significativamente a quantidade de neurônios criados, mas, por outro lado, determina o momento em que novos neurônios são adicionados à rede (vide Tabela 6.1). Quanto maior o valor fixado para o parâmetro

de novidade, mais rapidamente os neurônios serão inseridos na rede.

Tabela 6.1: Relação do parâmetro de novidade com a gênese da rede

Novidade δ	Plasticidade ρ	Número de neurônios $ \mathcal{A} $	Iteração da última inserção
0.10	0.7	8	1700
0.25	0.7	9	1564
0.50	0.7	11	1123
0.75	0.7	10	484
1.00	0.7	11	229

Na Figura 6.6, o fato de que o erro médio quadrado final de cada curva não varia significativamente no intervalo de ρ e δ entre as curvas sugere uma superfície de erro “bem comportada” (i.e., relativamente suave) para o problema. Como regra geral, a busca por valores adequados para os parâmetros de vigilância é guiada por tentativa-e-erro, após alguma experimentação inicial com os dados e a rede neural. Contudo, as curvas de aprendizagem experimentais mostradas na Figura 6.6 sugerem as seguintes tendências:

1. Com ρ próximo de 1, um número elevado de neurônios é adicionado na rede e pode ocorrer um fenômeno chamado sobre-ajuste (do inglês *overfitting*), tal que a rede neural perde sua capacidade de *generalização*;
2. Utilizando ρ menor que 0.5, a rede cria um número bem reduzido de neurônios e pode ocorrer um fenômeno chamado sub-ajuste (do inglês *underfitting*), tal que a rede se torna rígida demais para lidar com variados padrões de entrada;
3. Com δ próximo de 1, o tempo para adaptar os novos neurônios à estrutura da rede existente é significativamente reduzido, gerando redes visivelmente deformadas; e
4. Com δ próximo de zero, a quantidade de neurônios criados em um período de tempo é menor, entretanto, o tempo de treinamento da rede é consideravelmente maior.

6.2.2 Distribuição não-linearmente separável de dados

A Figura 6.7 apresenta o mapeamento obtido pela rede neural para a distribuição de densidade de duas espirais concêntricas². O problema de classificação de padrões das duas espirais, originalmente proposto por Alexis Wieland, é um *benchmark* da área de métodos de classificação e, até recentemente, tem recebido atenção especial da comunidade científica por se tratar de um problema *não-linearmente separável*³ de difícil solução. Este problema de classificação consiste em determinar em qual das duas espirais entrelaçadas pertence um determinado ponto do espaço. Cada espiral representa uma classe (ou categoria) e cada neurônio representa o protótipo de uma determinada classe. Na versão não-supervisionada do problema de classificação das duas espirais, o propósito é determinar se a rede é capaz de preservar a topologia subjacente do espaço de entrada.

²Esta distribuição de dados em forma de duas espirais concêntricas já foi utilizada em experimentações com outras redes neurais auto-organizáveis construtivas, dentre elas GDCS (BRUSKE; SOMMER, 1995), GNG (KUNZE; STEFFENS, 1995) e GWR (MARSLAND; SHAPIRO; NEHMZOW, 2002). As amostras foram obtidas através do repositório oficial de redes neurais artificiais da Universidade de Carnegie Mellon (CMU).

³Um problema de classificação não-linearmente separável é aquele que possui padrões em determinadas posições do espaço de características que não podem ser separados simplesmente por uma linha reta.

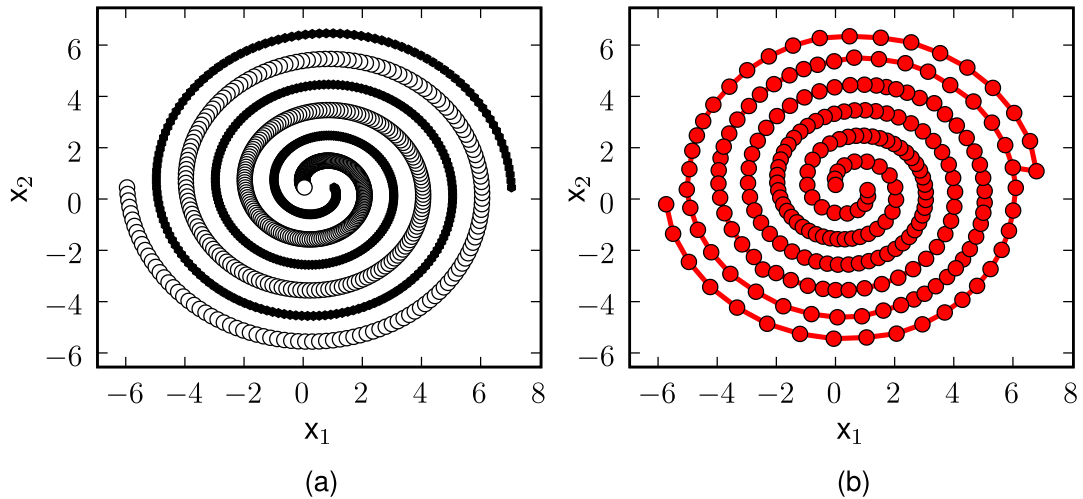


Figura 6.7: Simulação de uma distribuição de dados em forma de duas espirais com $\rho = 0.7$ e $\delta = 0.1$. (a) Distribuição de densidade de duas espirais concêntricas. Ao todo são 960 amostras. (b) Condição da rede após 10 épocas de treinamento.

6.2.3 Distribuição multimodal de dados

A rede neural proposta é capaz de representar estruturas bastante complexas, inclusive com áreas alongadas e regiões desconexas. A Figura 6.8 mostra um exemplo do mapeamento obtido pela rede neural para uma distribuição multimodal de dados. Em particular, os padrões de treinamento estão distribuídos em quatro áreas distintas e cada área está adequadamente representada por um conjunto de protótipos.

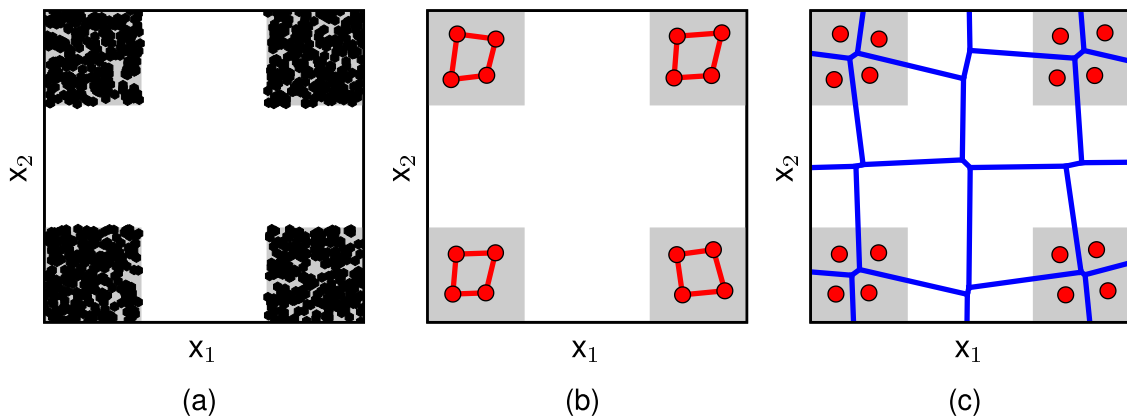


Figura 6.8: Simulação de uma distribuição multimodal de dados com $\rho = 0.85$ e $\delta = 0.1$. (a) Distribuição de entrada (1000 amostras distribuídas em quatro regiões desconexas). (b) Condição da rede após 10 épocas de treinamento. (c) Diagrama de Voronoi resultante.

A simulação acima pode ser compreendida como uma tarefa de identificação de agrupamentos (clusterização). A identificação de agrupamentos é uma *tarefa descritiva* que procura agrupar dados usando a *similaridade* dos valores de seus atributos como fator de decisão quanto à sua *pertinência* num entre vários agrupamentos possíveis. Este processo é realizado de forma não-supervisionada, pois a similaridade entre os atributos é uma característica intrínseca dos dados, não necessitando de um arquivo de treinamento com classes pré-definidas. A Figura 6.9 ilustra o comportamento da rede nesta tarefa.

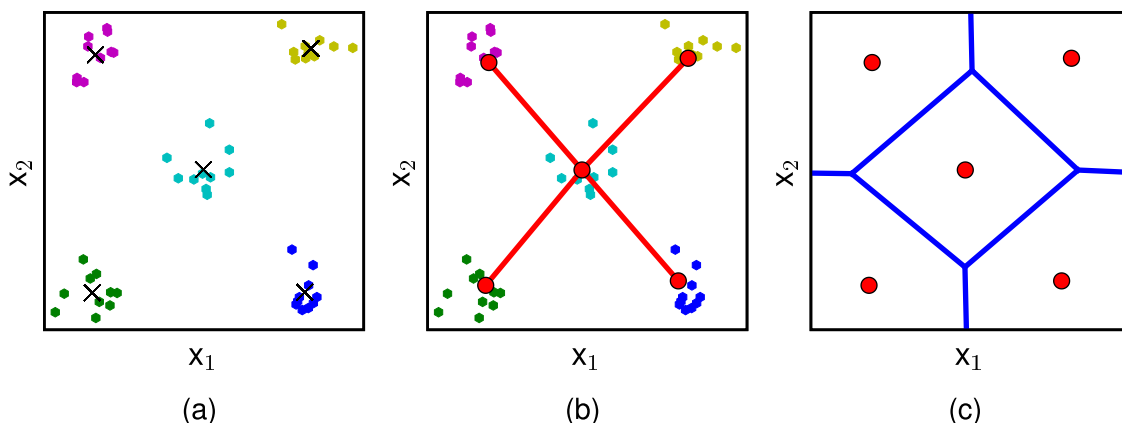


Figura 6.9: Identificação de agrupamentos com $n_{max} = 5$. (a) Modelo gerador com cinco agrupamentos. O símbolo \times representa o centróide de um determinado agrupamento. (b) Os cinco protótipos calculados pela rede neural se aproximam dos cinco centróides do modelo gerador. (c) Diagrama de Voronoi resultante.

6.2.4 Distribuição não-estacionária de dados

A Figura 6.10 mostra a distribuição não-estacionária de dados utilizada para treinar a rede neural. Inicialmente, as amostras são retiradas de dois quadrados posicionados no canto inferior esquerdo e superior direito da área da Figura 6.10a. Após 5 épocas – na cinco milésima primeira iteração – as amostras são extraídas de dois quadrados localizados no canto inferior direito e superior esquerdo (vide Figuras 6.10b–c). No total, a simulação compreende 10 épocas de aprendizagem, cada qual com 1000 iterações.

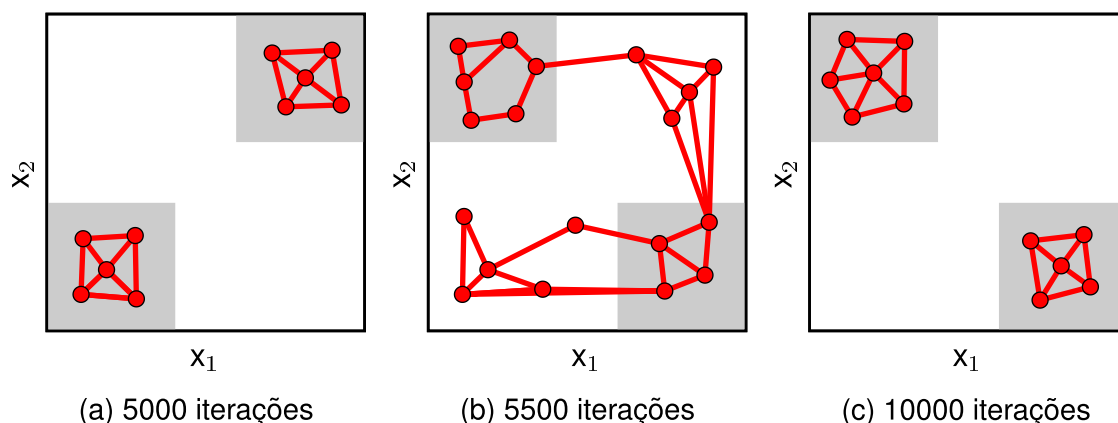


Figura 6.10: Simulação de uma distribuição não-estacionária de dados com $\rho = 0.8$ e $\delta = 0.1$. Ao todo são 1000 amostras uniformemente distribuídas dentro das áreas em cinza claro. (a) Configuração inicial do mapa de características. (b) A distribuição de dados é drasticamente modificada e se inicia um processo de inclusão e migração de neurônios. (c) Todos os neurônios são realocados para regiões de alta densidade probabilística.

O mapa de características calculado pela rede neural fornece uma boa aproximação à topologia subjacente do espaço de entrada. Após a distribuição sofrer uma drástica alteração, alguns neurônios são criados ou realocados para áreas de alta densidade probabilística, enquanto outros são removidos da rede. A Figura 6.11 mostra o erro de quantização e o número de neurônios criados no transcorrer do processo de aprendizagem.

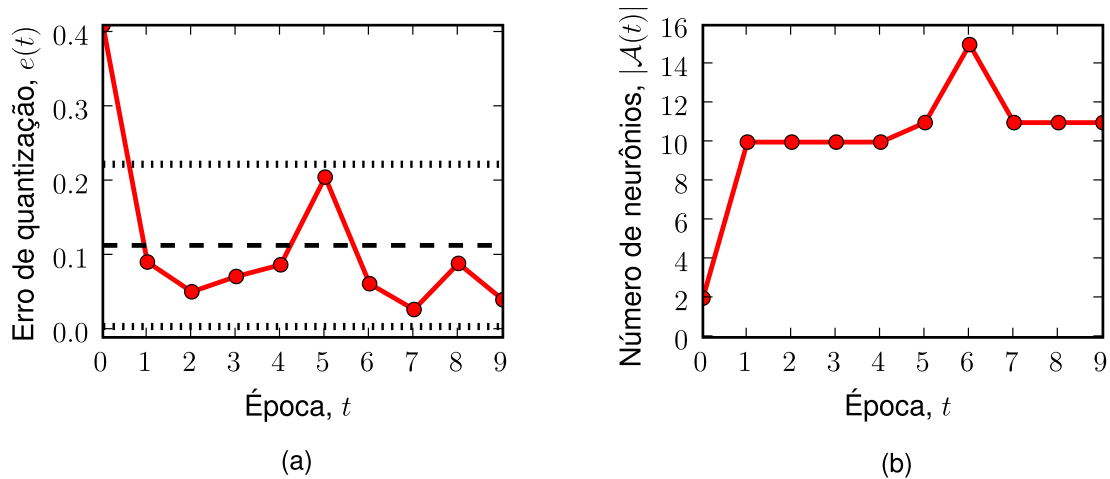


Figura 6.11: (a) Erro de quantização da rede (EMQ=0.00559). (b) Número de neurônios em função do tempo. O número mínimo de neurônios permitido na rede é fixado em 2. O número máximo de neurônios é governado pelo parâmetro n_{max} e, neste caso, vale 300.

Como regra geral, o algoritmo proposto remove neurônios não-representativos (ou supérfluos) para a distribuição de entrada e adiciona novas unidades em posições mais significativas para a diminuição do erro de quantização da rede. Um neurônio é considerado supérfluo caso não vença a competição por um longo período de tempo. Esse período é definido de forma implícita através da escolha das taxas de sensibilização e habituação da sinapse de habituação do neurônio (parâmetros α e β , respectivamente). A taxa de sensibilização faz aumentar paulatinamente o valor da sinapse de habituação até o valor 1. Já a taxa de habituação faz diminuir gradativamente o valor da sinapse de habituação até o valor zero. À medida que o processo de aprendizagem se desenvolve, cada neurônio tem sua sinapse de habituação, w^h , ajustada, de acordo com os valores fixados para α e β . Por convenção, um neurônio de índice i é removido da rede se a sua sinapse de habituação ultrapassar o valor 0.95. A Figura 6.12 apresenta o valor do peso da sinapse de habituação de um neurônio arbitrário no decorrer do processo de aprendizagem.

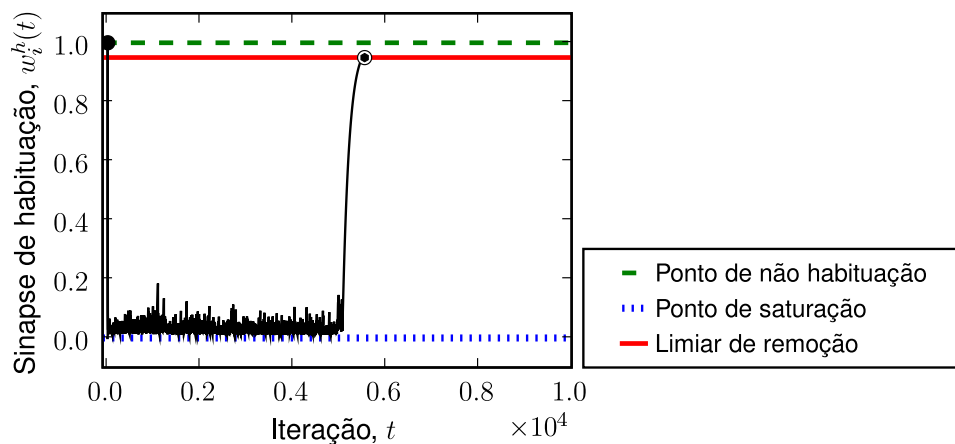


Figura 6.12: Valor do peso da sinapse de habituação de um neurônio arbitrário em função do tempo. O neurônio é adicionado à rede no instante de tempo 11. A distribuição de dados é alterada após 5 épocas. Neste momento, o neurônio se torna não-representativo para a distribuição de densidade e é removido da rede no instante de tempo 5542.

Na simulação da Figura 6.10, neurônios que não vencem a competição por um longo período são removidos da rede porque a taxa de sensibilização é igual a 0.1. Já na simulação da Figura 6.13, nenhum neurônio supérfluo é removido da rede porque a taxa de sensibilização é feita igual a zero. Neste caso, a rede possui uma *memória de longo prazo* (LTM, do inglês *long-term memory*) que retém de forma definitiva a informação aprendida. A situação ilustrada na Figura 6.13c também ocorre com uma rede de Kohonen convencional. Inicialmente, a rede de Kohonen consegue representar bem a distribuição de dados. No entanto, quando a distribuição é significativamente modificada, alguns neurônios da rede não podem ser retreinados pois a taxa de aprendizagem possui um valor muito pequeno e próximo de zero (i.e., $\eta(t) \approx 0$) e, conseqüentemente, se tornam neurônios inoperantes. Em contrapartida, a rede neural proposta consegue resolver esta problemática, desde que seja fixado um valor maior que 0.0 para a taxa de sensibilização⁴.

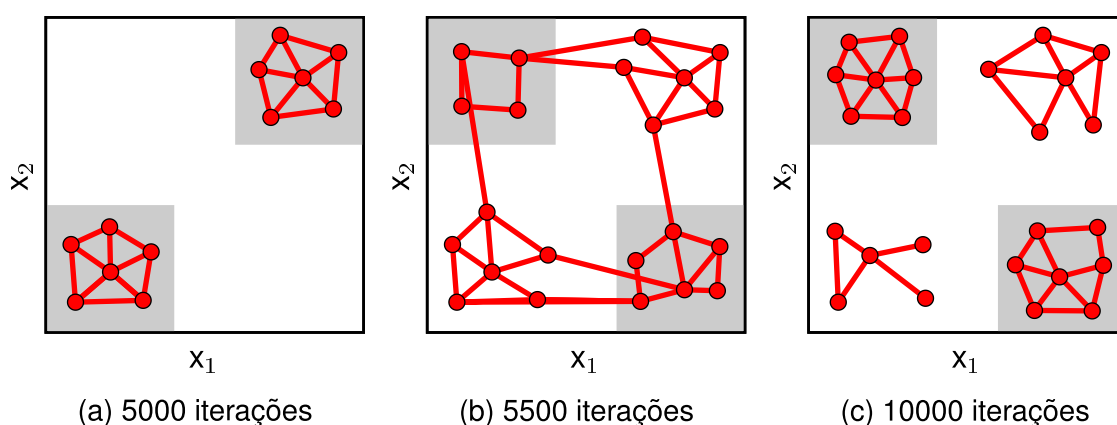


Figura 6.13: Simulação de uma distribuição não-estacionária de dados com $\rho = 0.8$ e $\delta = 0.1$. Em particular, o parâmetro da taxa de sensibilização, α , foi fixado em 0.0. No total, são 1000 amostras uniformemente distribuídas sobre as áreas em cinza claro. (a–c) Configuração do mapa antes, durante e depois da distribuição sofrer uma alteração.

6.2.5 Distribuição não-uniforme de dados

O mapa de características calculado pela rede neural proposta reflete variações na estatística da distribuição de dados de entrada. Em outras palavras, regiões no espaço de entrada de onde as amostras sejam retiradas com uma alta probabilidade de ocorrência são mapeadas em regiões maiores do espaço de saída, e vice-versa. A Figura 6.14 ilustra um exemplo da estimação pontual da densidade de probabilidades. Assim como na rede de Kohonen convencional, a alocação de recursos não é proporcional à densidade de pontos da distribuição. Esta constatação pode ser verificada pelo número de neurônios criados para cada região da Figura 6.14. Como regra geral, a rede tende a subrepresentar regiões com alta densidade de entrada e sobrerrepresentar regiões com baixa densidade de entrada. Ou seja, o mapa fornece apenas uma representação pontual “subótima” da distribuição de probabilidade intrínseca dos dados de entrada.

⁴Esta distribuição não-estacionária de dados já foi utilizada em experimentações com outras redes neurais auto-organizáveis construtivas, tais como GNG com fator de utilidade (FRITZKE, 1997a), GWR (MARS-LAND; SHAPIRO; NEHMZOW, 2002) e, mais recentemente, DASH (HUNG; WERMTER, 2003).

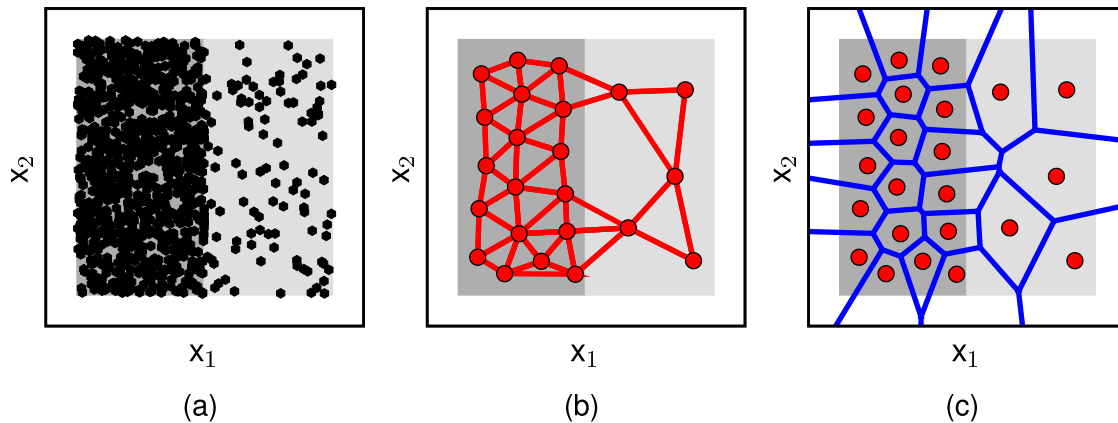


Figura 6.14: Simulação de uma distribuição não-uniforme de dados com $\rho = 0.85$, $\delta = 0.1$. (a) Distribuição dos dados de entrada. Ao todo são 1000 amostras distribuídas sobre as áreas em cinza: 900 dentro do retângulo da esquerda e apenas 100 ao longo do retângulo da direita. (b) Alocação de mais recursos (i.e., neurônios) para a região com maior densidade de amostras. (c) Diagrama de Voronoi resultante.

6.3 Segmentação de seqüências temporais

Nesta seção utiliza-se um experimento para ilustrar o comportamento da rede neural proposta em uma tarefa de segmentação temporal⁵. Nesta tarefa, o objetivo principal é verificar se a rede neural da Figura 5.3 é capaz de segmentar corretamente a seqüência temporal em trechos significativos e identificar posições de ambigüidade espaço-temporal.

6.3.1 Distribuição espaço-temporal de dados em forma de “8”

A Figura 6.15 apresenta a seqüência temporal de duas dimensões em forma de “8”.

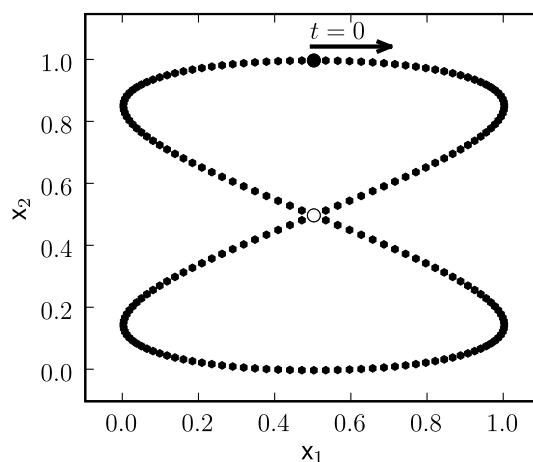


Figura 6.15: Ilustração da seqüência temporal em forma de “8”. O círculo fechado (‘●’) simboliza o ponto inicial da seqüência ($t = 0$), a seta indica o sentido do fluxo (da esquerda para a direita) e o círculo aberto (‘○’) representa os dois pontos ambíguos da seqüência temporal ($t = 50$ e $t = 150$), alocados na mesma posição espacial (0.5, 0.5).

⁵Este experimento de segmentação de seqüências temporais já foi feito com outras redes neurais auto-organizáveis temporais, dentre elas TDSOM (KANGAS, 1990), rede de Araújo e Barreto (ARAÚJO; BARRETO, 2002) e, mais recentemente, rede de Kohonen com memória diferenciadora-integradora (MOSER, 2004).

A seqüência temporal da Figura 6.15 é gerada a partir das Equações 6.1 e 6.2 para $1 \leq t \leq 200$. Os 200 pontos obtidos são submetidos à rede neural em uma determinada ordem, de tal maneira que a seqüência possa ser interpretada como a trajetória de um robô móvel – ou agente de *software* – realizada dentro de um ambiente. Nesta seqüência existe um único ponto de ambigüidade espacial, posicionado na coordenada $(0.5, 0.5)$. Os dois instantes em que a seqüência temporal passa por este ponto são $t = 50$ e $t = 150$.

$$x_1 = 0.5 \sin\left(\frac{t}{200}4\pi\right) + 0.5 \quad (6.1)$$

$$x_2 = 0.5 \cos\left(\frac{t}{200}2\pi\right) + 0.5 \quad (6.2)$$

A rede neural foi submetida a 10 épocas de aprendizagem, cada qual com 200 iterações. A cada instante, a camada de entrada da rede recebe um padrão de entrada espaço-temporal $\mathbf{z}(t)$ de dimensão 4-por-1, tal que $\mathbf{z}(t) = [x_1(t), x_2(t), y_1(t), y_2(t)]^T$. Na Figura 6.16 é apresentado o resultado obtido pela rede proposta de parâmetros $n_{max} = 20$, $c_{max} = 60$, $i_{max} = 50$, $\rho = 0.7$, $\delta = 0.1$, $\mu = 0.7$, $\eta = 0.7$, $\alpha = 0.01$, $\beta = 0.9$ e $\gamma = 0.7$.

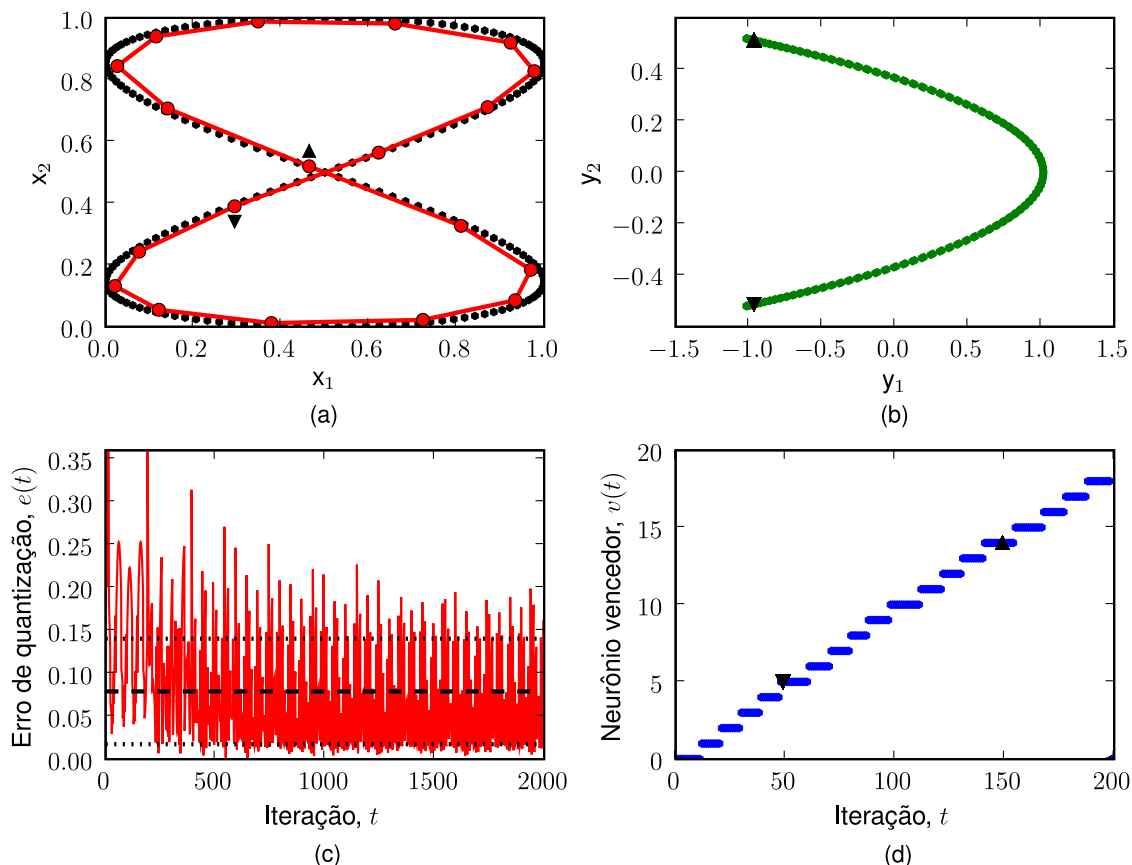


Figura 6.16: Simulação de uma distribuição de dados em forma de “8” com $\rho = 0.7$, $\delta = 0.1$, $\mu = 0.7$, $\eta = 0.7$, $\alpha = 0.01$, $\beta = 0.9$ e $\gamma = 0.7$. (a) Condição da rede após 10 épocas de aprendizagem. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01001). (d) Atividade na saída da rede na última época. Os símbolos \blacktriangledown e \blacktriangle indicam os instantes $t = 50$ e $t = 150$, respectivamente, representados pelos neurônios de índice 5 e 14, respectivamente.

A Figura 6.16a mostra os valores dos pesos sinápticos após as 10 apresentações da seqüência temporal. Os vetores de pesos sinápticos são traçados como pontos no espaço de entrada. As linhas desenhadas representam a vizinhança dos neurônios. A Figura 6.16b apresenta o espaço de representação gerado pela unidade de memorização Ψ . Os parâmetros empregados na unidade de memorização foram $\mu = 0.7$ e $\eta = 0.7$, com a saída da unidade multiplicada por 10 para compensar a redução de amplitude do sinal e aumentar a importância da informação de contexto⁶. A Figura 6.16c mostra o erro de quantização da rede. A Figura 6.16d apresenta a atividade de saída de todos os neurônios da rede.

O arranjo de neurônios da Figura 6.16a encontra-se em uma configuração estável e a proximidade espaço-temporal dos neurônios não é violada. No gráfico da Figura 6.16d o primeiro neurônio tem índice 0 e o último neurônio tem índice 18. A transição entre os neurônios de índices 0 e 18 não corresponde a um salto abrupto, pois os neurônios são vizinhos no mapa de características. A rede neural classifica a posição de ambigüidade – localizada na coordenada (0.5, 0.5) – com dois neurônios diferentes. Os neurônios de índice 5 e 14 representam os instantes 50 e 150, respectivamente (vide Figura 6.16d). Esta distinção somente é possível pois os dois instantes de tempo encontram-se em locais distintos no subespaço de entrada contextual \mathcal{Y} (vide Figura 6.16b). Mais especificamente, a distinção é realizada com base no sinal da componente y_2 do vetor de entrada contextual $\mathbf{y}(t)$: $\mathbf{z}(50) = [0.5, 0.5, -0.96394, 0.51279]$ e $\mathbf{z}(150) = [0.5, 0.5, -0.96394, -0.51279]$.

A Tabela 6.2 apresenta um resumo dos desempenhos da rede na tarefa de segmentação.

Tabela 6.2: Resumo dos desempenhos de segmentação da seqüência temporal “8”

Tentativa	Ambigüidade na representação	Erro médio quadrado final
1	Não	0.01286
2	Não	0.01096
3	Não	0.01151
4	Não	0.01172
5	Não	0.01050
6	Não	0.01139
7	Não	0.01205
8	Não	0.01089
9	Não	0.01278
10	Não	0.01001
Média	100%	0.011467
Desvio padrão	0.0	0.0009264

Na Figura 6.17 é ilustrado o resultado obtido por uma rede de Kohonen convencional com 20 neurônios dispostos num mapa circular unidimensional de parâmetros $\sigma = 1.5$ e $\eta = 1$. Para aumentar a convergência e a estabilidade dos resultados, a largura da vizinhança topológica, σ , tem seu valor reduzido linearmente até 1.0 e o valor da taxa de aprendizagem, η , decai monotonicamente até 0.0 conforme o gráfico da Figura 6.17b. Em todas as 10 tentativas realizadas a rede de Kohonen convencional falhou na representação da posição de ambigüidade e, em média, obteve um valor quadrado do erro de quantização igual a 1.9118×10^{-2} e desvio padrão igual a 0.11374×10^{-2} . É importante ressaltar que o erro de quantização da rede de Kohonen não leva em conta a distância temporal,

⁶A normalização dos dados de saída da unidade de memorização Ψ é realizada em tempo real por uma constante multiplicativa. A determinação da amplitude ideal para o sinal incidente é feita a partir da análise de sua função *kernel* (vide Eq. 4.12). Para maiores detalhes, consulte (MOSER, 2004, p. 65–66).

mas apenas a componente espacial do vetor de entrada toma parte neste erro. A representação da seqüência temporal classifica os dois instantes de tempo com o mesmo neurônio protótipo (vide neurônio de índice 7 na Figura 6.17d). A atividade na saída da rede sofre saltos abruptos no decorrer da seqüência e, por isso, o mapa de características não reflete espacialmente a contigüidade temporal da seqüência. Esta ambigüidade de representação existe independente dos parâmetros escolhidos para σ e η , pois a rede carece de uma unidade de memorização (e.g., memória de curto prazo ou laços de realimentação).

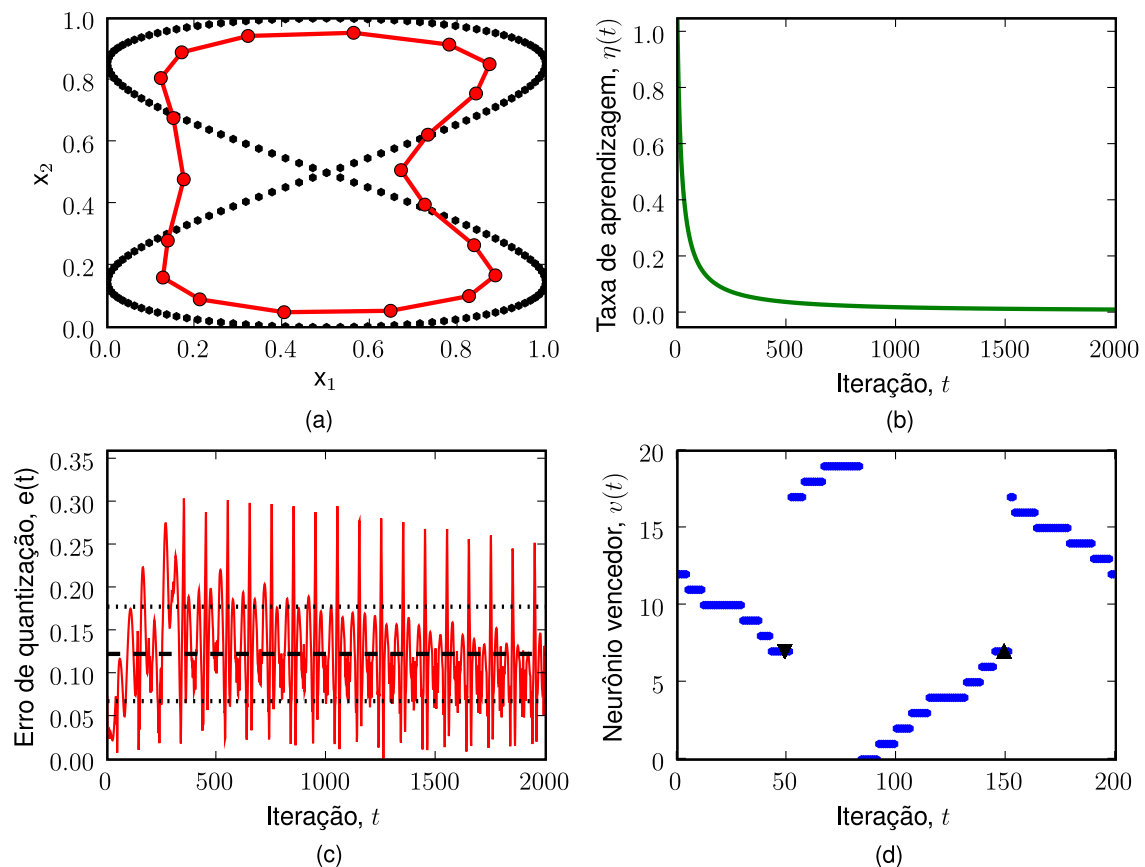


Figura 6.17: Simulação de uma distribuição de dados em forma de “8” com 20 neurônios dispostos num mapa de Kohonen unidimensional circular de parâmetros $\sigma = 1.5$ e $\eta = 1$. (a) Condição da rede após 10 épocas de treinamento. (b) Heurística usada para decrescer monotonicamente o valor de η com o transcorrer do treinamento. (c) Erro de quantização da rede (EMQ=0.019118). (d) Atividade na saída da rede na última época.

Para ampliar a discussão sobre o tema, esta mesma tarefa de segmentação temporal foi submetida às redes neurais auto-organizáveis temporais discutidas no Capítulo 4. Em geral, as redes temporais classificam o ponto de cruzamento com neurônios diferentes. A razão para isso é a inclusão de uma informação de contexto ao espaço de representação da rede, tanto na forma de uma memória de curto prazo (e.g., linha de atrasos de tempo, integradores e/ou diferenciadores) como na forma de laços de realimentação (i.e., por recorrência da atividade neural). A Figura 6.18 mostra a freqüência de convergência dos neurônios de cada rede temporal estudada. Os resultados apresentados são a média de 20 episódios da seqüência temporal em forma de “8”. Os seguintes parâmetros foram usados para as redes temporais: (i) TKM com constante de retenção de atividade, λ , igual a 0.8; (ii) RSOM com profundidade de memória, λ , igual a 0.05; (iii) CSOM com constantes de

esquecimento, α e β , iguais a 1.0 e 0.5; (iv) SOMSD de parâmetros $\alpha = 1.0$ e $\beta = 0.08$; (v) MSOM de parâmetros $\alpha = 0.5$ e $\beta = 0.5$; (vi) SARDNET com taxa de decaimento, τ , igual a 0.95; e (vii) DIN com constantes de integração, η e μ , iguais a 0.7 e 0.7.

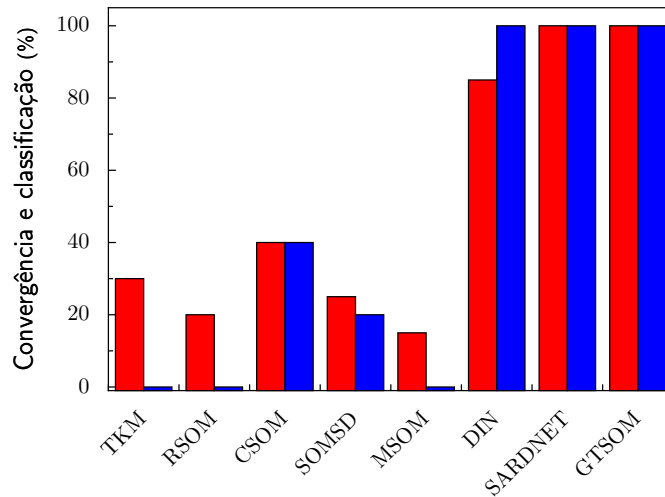


Figura 6.18: Frequências de convergência e de classificação de cada rede neural auto-organizável na tarefa de segmentação temporal. A barra da esquerda indica a porcentagem de convergência dos neurônios protótipos, ou seja, a frequência de representação estável da seqüência temporal. A barra da direita aponta a porcentagem de acerto na classificação da seqüência temporal, isto é, a frequência de identificação da posição de ambigüidade.

Com exceção da rede neural proposta que realiza a inserção e a remoção de neurônios ao longo do processo de aprendizagem, a representação temporal das redes temporais é sensível à configuração inicial dos pesos sinápticos. O melhor resultado de classificação foi obtido pela rede neural proposta, pois em 100% dos experimentos houve convergência estável e não ambígua de representação da seqüência temporal e o valor médio do erro médio quadrado de quantização foi o mais baixo de todos (1.001×10^{-2}). A Figura 6.19 apresenta o erro médio quadrado de quantização de cada rede temporal estudada.

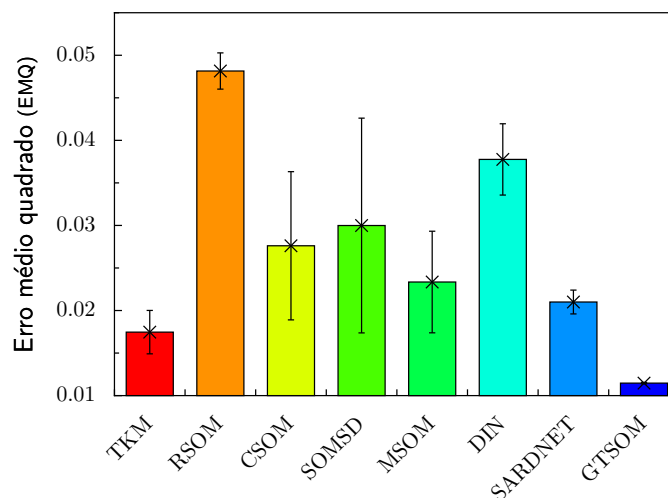


Figura 6.19: Erro médio quadrado de quantização de cada rede neural auto-organizável na tarefa de segmentação temporal. A linha vertical sobre cada barra representa a margem de erro, isto é, o desvio padrão para mais e para menos em torno da média.

No intuito de verificar a estabilidade da representação gerada pela rede neural aplicou-se a seqüência temporal em forma de “8” com perturbações de ruído uniforme no intervalo $[-0.1, 0.1]$, translação e rotação. Como será analisado a seguir, a classificação da seqüência temporal ruidosa é invariante às perturbações testadas, principalmente a perturbações de ruído e translação da seqüência. Os resultados obtidos nas Figuras 6.20, 6.21 e 6.22 são semelhantes aos ilustrados na Figura 6.16, isto é, a rede não apresenta defeitos topológicos na formação do mapa de características e a posição de ambigüidade é classificada com neurônios protótipos distintos.

A Figura 6.20 apresenta os resultados de classificação da seqüência temporal com ruído de distribuição uniforme no intervalo $[-0.1, 0.1]$. A aplicação do ruído sobre a seqüência original “8” não prejudica a classificação final, pois a propriedade de redução de ruído por integração temporal da unidade de memorização Ψ garante invariância na segmentação da seqüência ruidosa e a propriedade de preservação de topologia da rede assegura a corretude no mapeamento de características Φ . Neste problema específico, a tolerância máxima de ruído uniforme compreende o intervalo de -0.25 a 0.25 . Para valores fora desta faixa, a rede neural gera um mapa de características bastante distorcido e classifica erroneamente a seqüência temporal. A tolerância a ruído é de fundamental importância para problemas robóticos, pois os sinais provindos dos sensores e os comandos enviados aos atuadores dos robôs geralmente são medidas imprecisas dotadas de ruído.

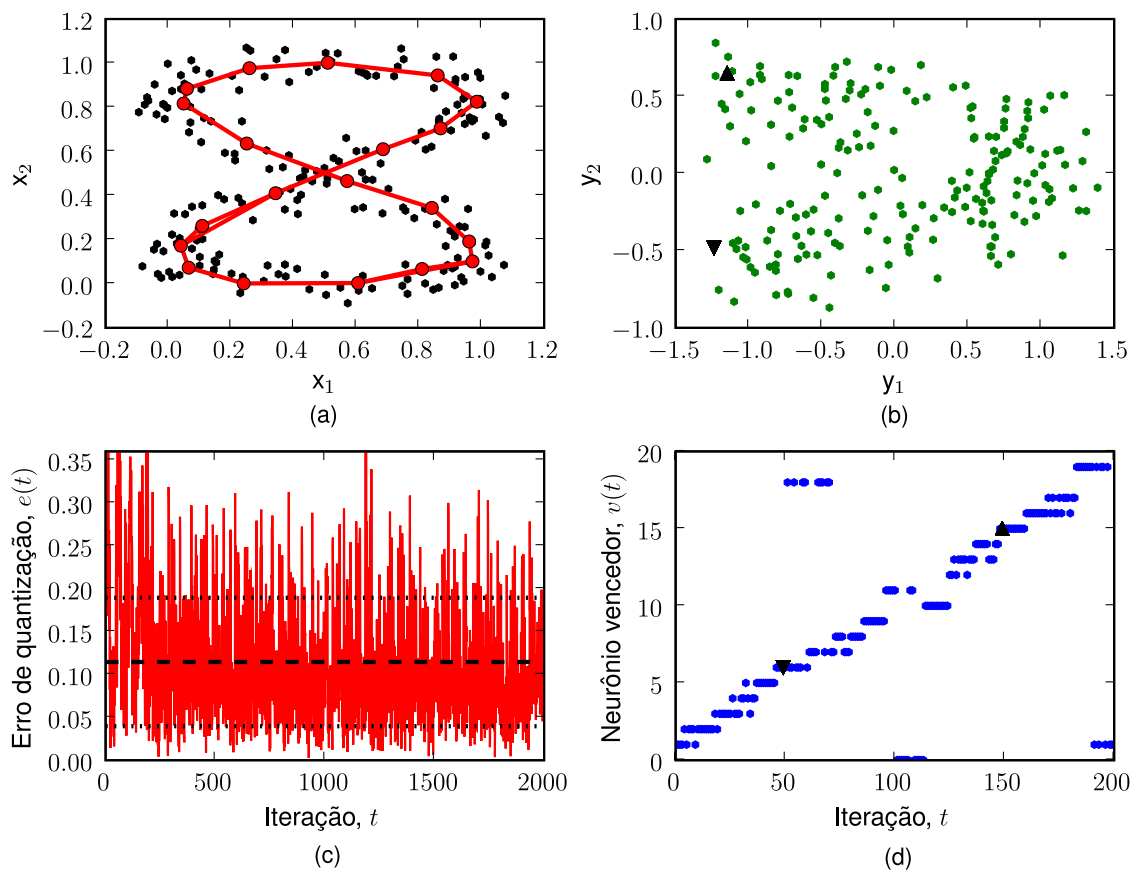


Figura 6.20: Simulação de uma distribuição de dados em forma de “8” com ruído de distribuição uniforme no intervalo $[-0.1, 0.1]$. (a) Condição da rede após 10 épocas. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01872). (d) Atividade na saída da rede na última época.

A Figura 6.21 apresenta os resultados de classificação da seqüência temporal com perturbação de translação. Como pode ser observado na Figura 6.21, a classificação obtida é completamente invariante à translação e bastante similar à representação da seqüência temporal sem translação da Figura 6.16. A invariância à translação, também denominada na literatura por invariância ao deslocamento, é uma propriedade que pode ser aplicada em uma série de problemas robóticos relacionados ao processamento de imagens e sinais. Nestes problemas, deseja-se que alguma característica do padrão apresentado seja extraída ou reconhecida, independente da translação ao longo de uma dimensão. Por exemplo, considere um filtro para detecção de bordas em uma imagem. É extremamente desejável que o filtro efetue a mesma operação sobre todos os pontos da imagem, independente se uma parte da imagem foi transladada.

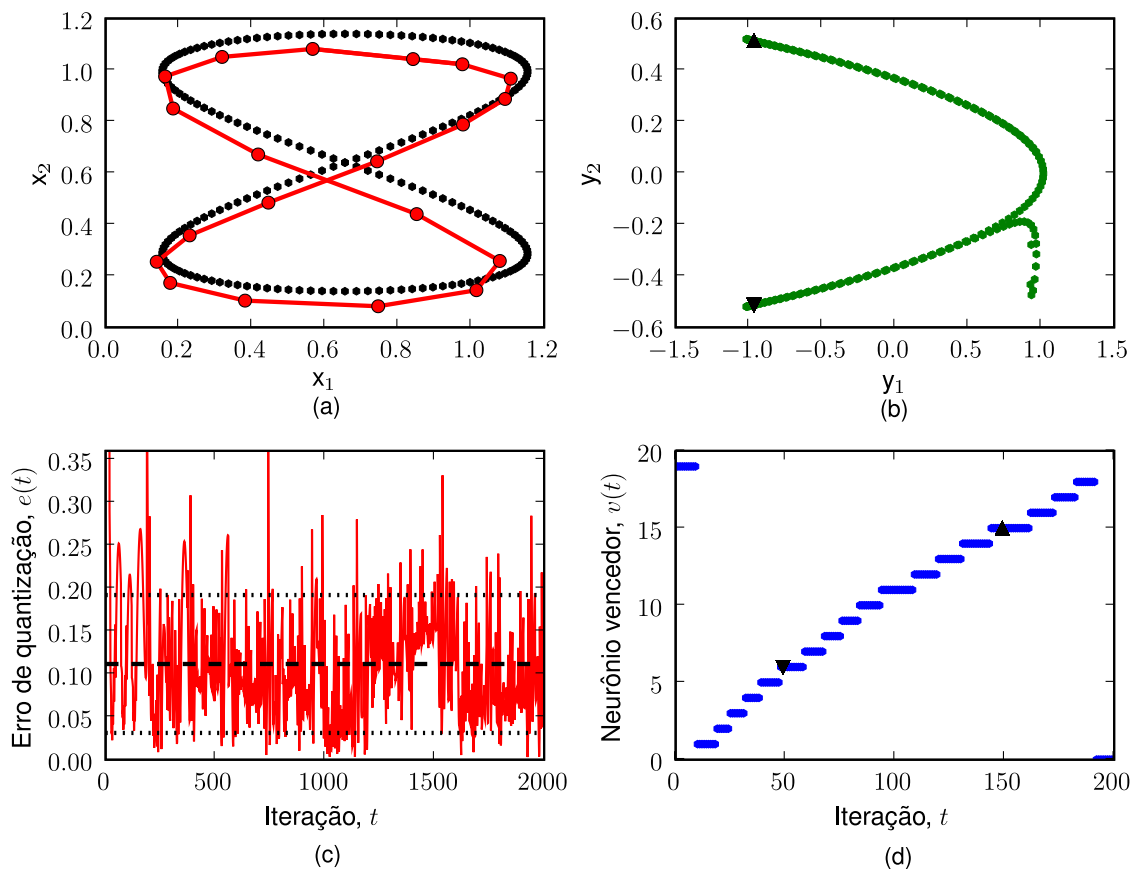


Figura 6.21: Simulação de uma distribuição de dados em forma de “8” com ruído de translação. (a) Condição da rede após 10 épocas de aprendizagem. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01891). (d) Atividade na saída da rede na última época.

A Figura 6.22 apresenta o resultado da perturbação de rotação do sinal. A rotação é uma perturbação que afeta a representação da unidade de memorização Ψ (vide gráfico da Figura 6.22b). No entanto, os resultados ainda assim podem ser considerados satisfatórios, pois a classificação final identifica a posição de ambigüidade com dois neurônios protótipos diferentes. Constata-se nas Figuras 6.22a e 6.22b, que o subespaço de entrada contextual \mathcal{Y} é mais estável que o subespaço de entrada sensorial \mathcal{X} . Isto implica a redução da influência desta perturbação e o aumento da estabilidade da classificação. Em muitos problemas robóticos, a invariância à rotação é imprescindível para o robô catego-

rizar os objetos físicos presentes no meio pois uma pequena rotação pode levar à oclusão ou ao aparecimento de uma parte substancial do objeto em questão. Categorizar objetos significa realizar distinções entre diferentes tipos de objetos e reconhecer um mesmo objeto sob diferentes ângulos de visão. Em um mecanismo de *software* ou *hardware* sem qualquer robustez à rotação, os objetos para serem reconhecidos têm que estar exatamente no mesmo local e na mesma orientação com que foram inicialmente observados.

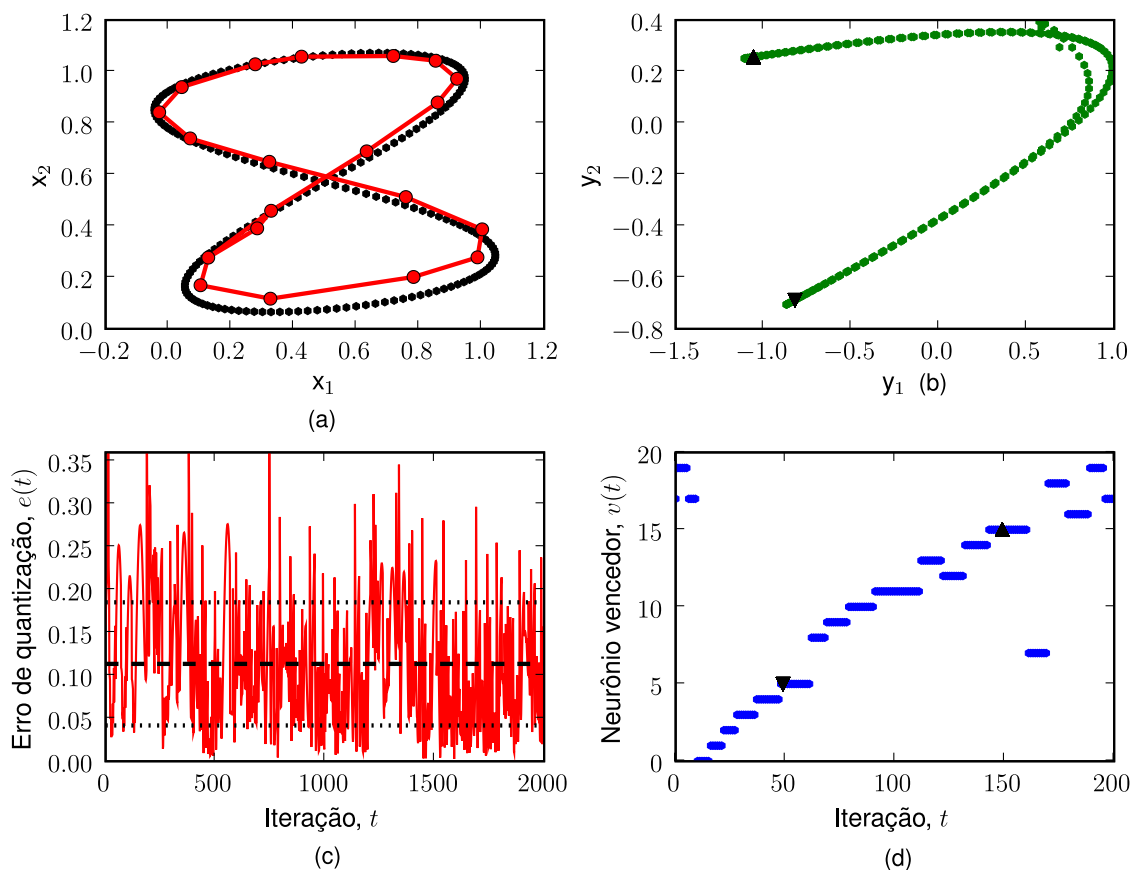


Figura 6.22: Simulação de uma distribuição de dados em forma de “8” com ruído de rotação. (a) Condição da rede após 10 épocas de aprendizagem. (b) Espaço de representação gerado pela unidade de memorização Ψ . (c) Erro de quantização da rede (EMQ=0.01803). (d) Atividade na saída da rede na última época.

6.3.2 Distribuição espaço-temporal de dados em forma de circunferência

Para finalizar, considera-se uma generalização do problema de classificação da seqüência temporal em forma de “8”, onde todos os pontos da seqüência têm ambigüidade espaço-temporal. A seqüência é definida sobre uma circunferência de raio 0.5 e centro (0.5, 0.5), conforme ilustra a Figura 6.23. Os 50 pontos igualmente espaçados da circunferência são submetidos à rede neural em uma determinada ordem, de tal forma que a seqüência possa ser interpretada como a trajetória de um robô móvel – ou agente de *software* – realizada dentro de um ambiente. O robô pode percorrer o circuito, partindo de um ponto inicial simbolizado por \bullet , em dois sentidos opostos: horário (vide Figura 6.23a) ou anti-horário (vide Figura 6.23b). Por convenção, o sentido anti-horário (contrário aos ponteiros de um relógio) é adotado como sentido negativo. No total, a simulação compreende 20 épocas de aprendizagem, cada qual com 50 iterações. Entre a primeira e a quinta

época e entre a décima e a décima quinta época, as amostras são retiradas no sentido de rotação horário. Entre a quinta e a décima época e entre a décima quinta e a vigésima época, as amostras são extraídas no sentido de rotação anti-horário.

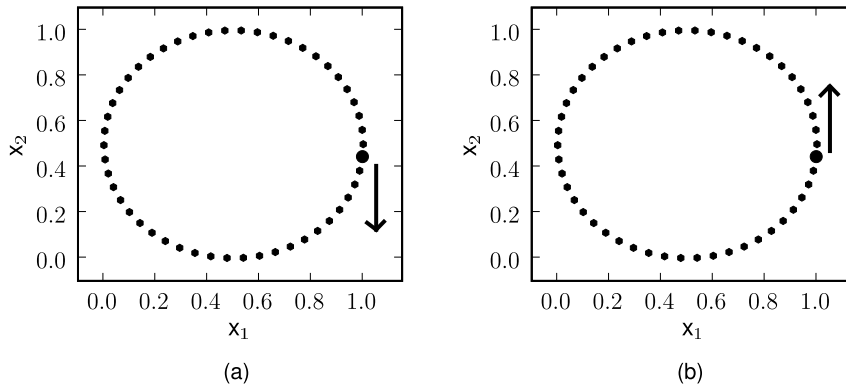


Figura 6.23: Ilustração da seqüência temporal definida sobre uma circunferência de raio igual a 0.5 e centro igual a (0.5, 0.5). No total, são 50 amostras uniformemente distribuídas ao longo da circunferência. O símbolo \bullet representa o ponto inicial da seqüência e a seta indica o sentido de rotação do fluxo: (a) horário ($t = \{0..249; 500..749\}$) ou (b) anti-horário ($t = \{250..499; 750..999\}$).

Nesta tarefa, o objetivo principal é verificar se a rede neural proposta é capaz de identificar o sentido de rotação da seqüência. Os seguintes valores foram fixados para os parâmetros do modelo: $n_{max} = 50$, $c_{max} = 150$, $i_{max} = 20$, $\rho = 0.7$, $\delta = 0.1$, $\mu = 0.6$, $\eta = 0.4$, $\alpha = 0.05$, $\beta = 0.9$ e $\gamma = 0.7$. A saída da unidade de memorização foi multiplicada por 2 para compensar a redução de amplitude do sinal e aumentar a importância da informação de contexto. Cabe ressaltar que não houve nenhuma seleção exaustiva de parâmetros e que outros valores para os parâmetros do modelo geram resultados semelhantes aos relatados a seguir. A Figura 6.24 traz os resultados.

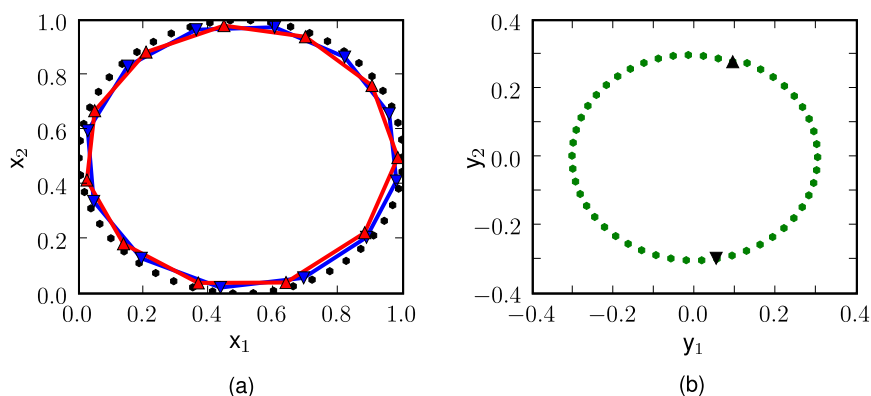


Figura 6.24: Simulação de uma distribuição de dados em forma de circunferência com $\rho = 0.7$, $\delta = 0.1$, $\mu = 0.6$, $\eta = 0.4$, $\alpha = 0.05$, $\beta = 0.9$, $\gamma = 0.7$. (a) Condição da rede após 10 épocas de aprendizagem. O símbolo \blacktriangledown indica os neurônios ativos durante o sentido de rotação horário. O símbolo \blacktriangle aponta os neurônios ativos durante o sentido de rotação anti-horário. (b) Espaço de representação gerado pela unidade de memorização Ψ . Todos os 50 pontos são ambíguos. Os símbolos \blacktriangledown e \blacktriangle representam os pontos iniciais da seqüência temporal nos sentidos de rotação horário e anti-horário, respectivamente.

A cada instante de tempo, a camada de entrada da rede recebe um padrão de entrada espaço-temporal $\mathbf{z}(t)$ de dimensão 4-por-1, tal que $\mathbf{z}(t) = [x_1(t), x_2(t), y_1(t), y_2(t)]^T$. Da Figura 6.24 identifica-se dois agrupamentos (*clusters*) de neurônios superpostos. Cada agrupamento mapeia um sentido de rotação da seqüência temporal. Não existe nenhuma ligação entre os dois agrupamentos e, dessa forma, pode-se afirmar que a rede neural identificou um *contexto* para cada sentido de rotação. Esta observação pode ser melhor visualizada no gráfico em três dimensões da Figura 6.25.

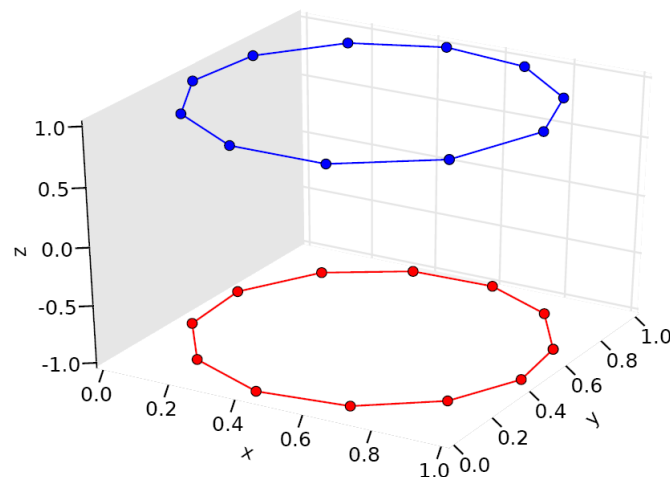


Figura 6.25: Identificação de dois agrupamentos distintos na distribuição de dados. Os neurônios posicionados no eixo $z = 1.0$ mapeiam o sentido de rotação horário. Os neurônios alocados no eixo $z = -1.0$ mapeiam o sentido de rotação anti-horário.

Note que o subespaço de entrada contextual \mathcal{Y} da Figura 6.24 tem raio 0.3 e centro $(0, 0)$. A distinção entre os dois sentidos de rotação é feita com base na magnitude e, principalmente, no sinal do vetor de contexto \mathbf{y} . Mais especificamente, a distinção é realizada de acordo com o sinal da componente y_2 do vetor de entrada contextual $\mathbf{y}(t)$. Por exemplo, no início da apresentação da seqüência temporal no sentido de rotação horário, o vetor de entrada espaço-temporal \mathbf{z} vale $[1.0, 0.5, 0.09358, 0.27831]$ e, no sentido de rotação anti-horário, o vetor de entrada espaço-temporal \mathbf{z} é igual a $[1.0, 0.5, 0.09833, -0.28678]$.

Na Figura 6.26 pode-se verificar que os dois sentidos de rotação da seqüência temporal foram mapeados com neurônios distintos para toda extensão de amostras ($1 \leq t \leq 1000$). Por um lado, o sentido de rotação horário da seqüência temporal foi classificado pelos neurônios de índice 0 a 11 (vide Figura 6.26b). Por outro lado, o sentido de rotação anti-horário foi identificado pelos neurônios de índice 12 a 22 (vide Figura 6.26d). Com este resultado, conclui-se que a rede neural resultante da aplicação do algoritmo proposto identifica casos extremos de ambigüidade espacial como este.

Nos experimentos realizados com as redes neurais auto-organizáveis temporais do Capítulo 4, verificou-se que um único agrupamento de neurônios foi criado para representar toda a extensão de amostras ($1 \leq t \leq 1000$). Os dois sentidos de rotação da seqüência temporal foram mapeados com os mesmos neurônios protótipos (vide Figura 6.27). É importante ressaltar que esta ambigüidade de representação existe independente dos parâmetros escolhidos, pois se deve principalmente a rígida estrutura topológica da rede que impossibilita desmembrar o arranjo bidimensional de neurônios em dois agrupamentos.

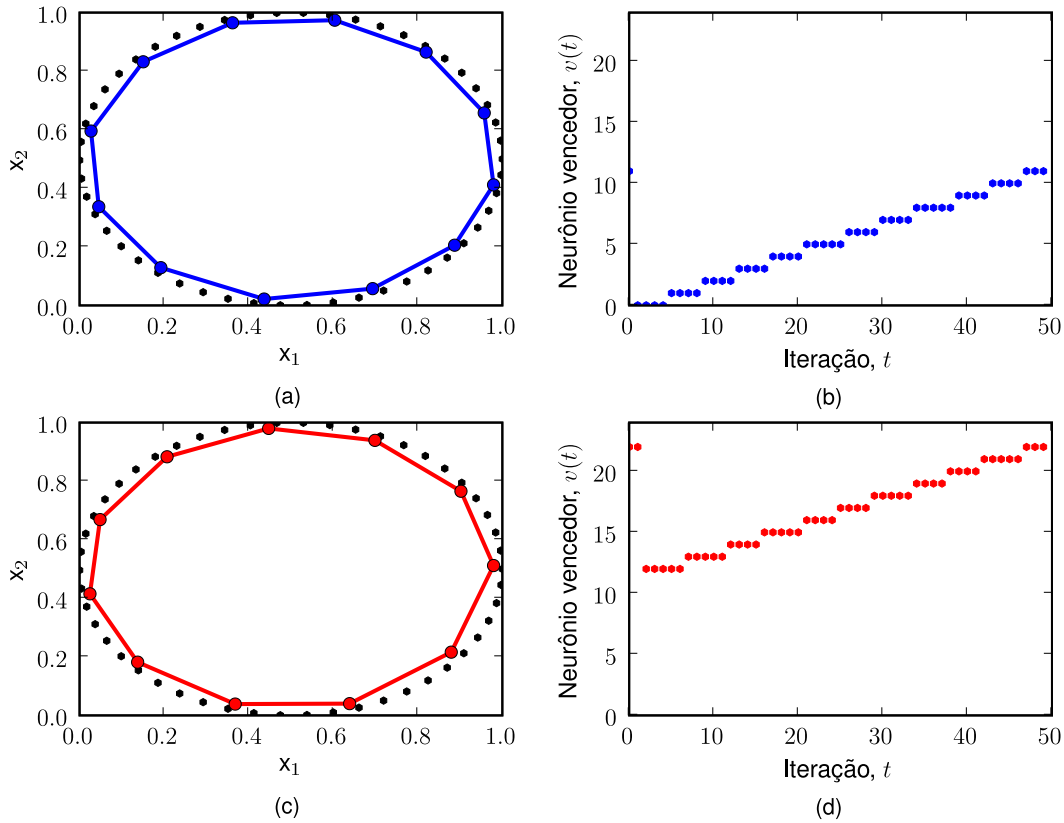


Figura 6.26: (a) Neurônios inseridos entre a primeira e a quinta época (rotação horária). (b) Atividade na saída da rede na quinta época. (c) Neurônios adicionados entre a quinta e a décima época (rotação anti-horária) (d) Atividade na saída da rede na décima época.

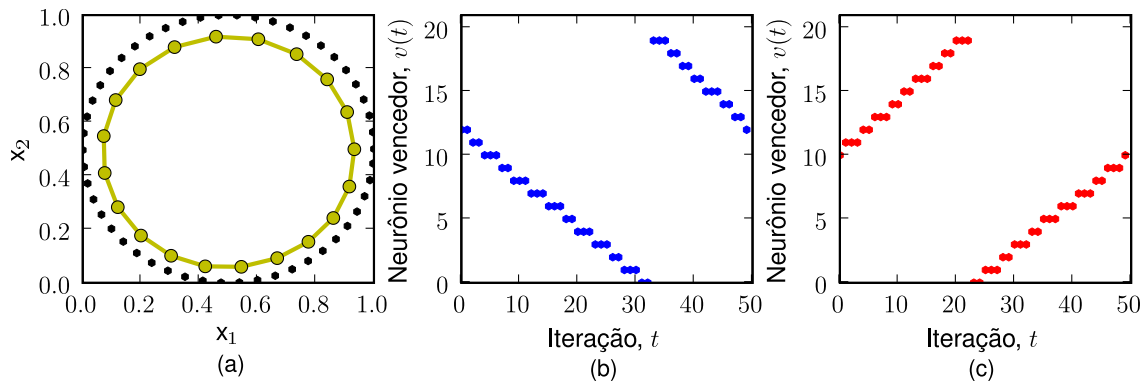


Figura 6.27: Simulação de uma distribuição de dados em forma de circunferência com uma rede TKM de parâmetro $\lambda = 0.8$. (a) Condição da grade após 10 épocas de aprendizagem. (b-c) Atividade na saída da rede na quinta época e na décima época, respectivamente. Note que todos os neurônios estão ativos em ambos os sentidos de rotação.

6.4 Considerações do capítulo

As simulações apresentadas neste capítulo são somente uma pequena amostra da vasta agenda de pesquisa que se inaugura a partir do tema de pesquisa sugerido nesta dissertação. No próximo capítulo é feito um apanhado geral dos principais avanços científicos e tecnológicos apresentados neste documento, bem como são discutidos possíveis tópicos para futuros desenvolvimentos nos domínios de aplicação aqui estudados.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 Sobre o trabalho

Este trabalho apresentou as etapas de projeto, teste e comparação de um novo modelo de rede neural artificial, denominado GTSOM (*Growing Temporal Self-Organizing Map*). O modelo foi concebido de acordo com três aspectos julgados essenciais em aplicações robóticas de natureza espaço-temporal e de horizonte infinito:

- *Auto-organização*, através da *competição global*, *cooperação local* e *auto-amplificação seletiva*. A auto-organização diz respeito à capacidade da rede neural alterar a sua estrutura de interconexão exclusivamente em função de sua experiência e de seu relacionamento com o meio ambiente;
- *Representação temporal*, por meio de uma memória de curto prazo diferenciadora e integradora aplicada na camada de entrada da rede. A representação temporal está relacionada com a maneira de incluir o tempo na operação da rede neural, que pode ser por linha de atrasos de tempo, diferenciadores, integradores e/ou laços de realimentação (recorrência); e
- *Aprendizado construtivo*, via inserção e remoção de neurônios e conexões sinápticas. O aprendizado construtivo está relacionado com a construção automática da arquitetura da rede neural, através de procedimentos de validação cruzada, técnicas de regularização, métodos de poda e/ou métodos construtivos.

No intuito de projetar o modelo de rede neural artificial, se fez necessária uma ampla revisão bibliográfica do estado da arte em princípios de projeto para agentes de *software* inteligentes, técnicas de processamento de sinais temporais para redes neurais artificiais e métodos de construção automática de arquitetura para redes neurais artificiais. Com base neste estudo, apresentado no decorrer dos Capítulos 2, 3 e 4, o modelo de rede neural artificial foi devidamente idealizado, projetado, construído e testado.

A formulação da rede neural proposta é baseada no processamento de dados espaço-temporais, ou seja, dados que estão relacionados de forma espacial e que possuem uma correlação temporal entre si. No entanto, a rede é *flexível* o suficiente para lidar com dados puramente espaciais, como é o caso de tarefas de quantização vetorial. Por exemplo, nas simulações computacionais da Seção 6.2, a arquitetura da rede neural foi ligeiramente modificada, simplesmente omitindo-se a unidade de memorização Ψ .

A unidade de memorização acoplada à rede neural é uma sugestão de Moser (2004) e tem como função realizar uma derivação e/ou integração da informação de entrada. O funcionamento da memória exige poucos recursos computacionais e apresenta grande

potencialidade na área da robótica, onde é necessário tratar diretamente o fluxo sensório-motor de robôs de forma rápida e econômica. Segundo Moser (2004, p. 93), problemas temporais que possuem alta correlação temporal (e.g., sinais sensoriais de robôs), com ou sem ruído, provavelmente podem se aproveitar das informações geradas pela memória. Em contrapartida, problemas com baixa correlação temporal ou simbólicos (e.g., cadeias de DNA) provavelmente se agravariam com o uso da memória. Cabe ressaltar que a arquitetura da rede neural é *modular*. Por essa razão, quando desejado, pode-se facilmente substituir a memória diferenciadora-integradora por outra unidade de memorização.

Diferentemente da maioria das redes construtivas que necessitam especificar previamente uma taxa de crescimento, a arquitetura da rede neural proposta é construída automaticamente ao longo do processo de aprendizagem. Uma unidade de habituação regula tanto o crescimento quanto a poda do número de neurônios da rede neural. As conexões sinápticas são criadas, removidas ou ajustadas de acordo com o método de aprendizado hebbiano competitivo. Por essa razão, a rede neural proposta oferece uma boa solução para o *dilema da estabilidade-plasticidade*, apontado por Stephen Grossberg em 1976 (HAYKIN, 2001, p. 30), e faz uso eficiente de memória se comparada com outras redes equivalentes (i.e., economia de neurônios e de conexões sinápticas).

A série de experimentos realizados neste trabalho comprovou o potencial do modelo neural proposto. Os cenários abrangeram tarefas simples de classificação de padrões puramente espaciais e segmentação de seqüências temporais. Os resultados preliminares obtidos no Capítulo 6 demonstraram a eficiência do modelo neural proposto frente às arquiteturas conexionistas existentes e foram considerados bastante satisfatórios com relação aos parâmetros avaliados. No entanto, os experimentos certamente não foram exaustivos e diversas outras aplicações poderiam ser testadas, como, por exemplo, tarefas de condicionamento e formação de conceitos em agentes de *software* inteligentes.

A nosso ver, a grande dificuldade e, ao mesmo tempo, a principal contribuição deste trabalho residiu em integrar os princípios de auto-organização, representação temporal e aprendizado construtivo em uma arquitetura neural de maneira viável e eficiente à solução de determinados problemas teóricos/práticos de agentes de *software* inteligentes. Dentre as limitações da rede neural proposta pode-se incluir a inexistência de uma estratégia de redução de dimensionalidade, que é uma característica inerente de redes neurais que preservam a topologia do espaço de entrada (*topology-representing networks*).

7.2 Trabalhos futuros

No desenvolvimento deste trabalho inúmeras possibilidades de ampliação da proposta inicial surgiram. Devido à complexidade dos objetivos traçados e à abrangência de áreas da ciência a serem exploradas, tais ampliações foram transformadas em propostas de possíveis trabalhos futuros. Além disso, por se tratar de um trabalho consideravelmente exploratório, existe um extenso campo de pesquisas a ser explorado. A seguir, são discutidos os principais desdobramentos e direções para o encaminhamento de pesquisas futuras:

- Agregar à informação armazenada no mapa de características um *segmento sensório-motor*. Cada segmento sensório-motor representaria a percepção e a respectiva tomada de decisão motora do agente de *software* num instante de tempo qualquer e poderia ser interpretado como um *conceito* adquirido pelo agente de *software* (e.g., se detectar um determinado evento sensório selecionar uma determinada ação motora). Esse mecanismo para formação de conceitos foi proposto originariamente por Verschure e Pfeifer (1992), cujo trabalho inspirou diversas arquiteturas neurais para

agentes de *software* situados e corporificados. Acredita-se que a informação temporal e o aspecto incremental do mapa de características auto-organizável podem ser de grande utilidade na descoberta de conceitos em ambientes não-estacionários.

- Ampliar os experimentos computacionais descritos no Capítulo 6 com tarefas robóticas de condicionamento. A idéia principal é utilizar o modelo neural proposto para extrair as informações perceptivas do agente de *software* e, juntamente com um sistema de valores previamente definido pelo projetista, associá-las com ações apropriadas – por exemplo, condicionar o agente a identificar uma fonte de alimentação como comestível ou venenosa através do sistema olfativo, independentemente do sistema gustativo. Acredita-se que o mapa de características pode ser de grande valia para associar um evento sensorio temporal com um evento motor característico. Em meados da década de 1990, Zrehen e Gaussier (1994) argumentaram que mapas topológicos são extremamente úteis na aprendizagem robótica e, mais recentemente, uma série de modelos neurais ontogênicos foi proposta na literatura (HARTER; KOZMA; GRAESSER, 2001) com o intuito de resolver tarefas de condicionamento clássico (e.g., vide (HARTER; KOZMA, 2004)).
- Elaborar um *controlador interno* para ajustar em tempo real os parâmetros de vigilância do modelo. O aprendizado do modelo neural proposto é contínuo e relativamente estável, entretanto, bastante parametrizável. A influência dos parâmetros na aprendizagem é extremamente dependente das propriedades do ambiente e dos sensores do agente de *software*. Não há ainda um algoritmo ou uma heurística para determinar valores adequados para os parâmetros de plasticidade, ρ , e de novidade, δ , do modelo. Logo, a escolha desses valores é baseada na experiência empírica do projetista. Uma pesquisa interessante seria propor um mecanismo que calculasse em tempo real e de forma automática valores ótimos para os parâmetros de vigilância, assegurando uma operação realmente autônoma do agente de *software*.
- Modificar a estrutura da rede neural proposta com a adição de uma segunda camada computacional de neurônios a fim de resolver tarefas de aproximação de funções e de reconhecimento estatístico de padrões. A primeira camada de neurônios seria responsável por identificar de maneira não-supervisionada um conjunto razoavelmente pequeno de padrões, no qual está concentrado o conteúdo de informação essencial dos dados brutos de entrada. Já a segunda camada computacional de neurônios teria um esquema de aprendizagem supervisionada – pela técnica de *quantização vetorial por aprendizagem (LVQ)* ou através de funções de base radial (RBF) – para classificar os padrões extraídos ou aproximar as funções. Acredita-se que este mecanismo de *aprendizagem semi-supervisionada* pode ser utilizado para solucionar uma ampla classe de problemas robóticos – por exemplo, estimar o valor de estados futuros nunca antes visitados com uma precisão razoável.

REFERÊNCIAS

AHRNS, I.; BRUSKE, J.; SOMMER, G. On-line Learning with Dynamic Cell Structures. In: INTERNATIONAL CONFERENCE IN ARTIFICIAL NEURAL NETWORKS, 1995. **Proceedings...** Paris, França, 1995. p. 141–146.

ALAHAKOON, D.; HALGAMUGE, S. K.; SRINIVASAN, B. Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery. **IEEE Transactions on Neural Networks**, New York, v. 11, n. 3, p. 601–614, 2000.

ALPAYDIN, E. GAL: Networks that grow when they learn and shrink when they forget. **International Journal of Pattern Recognition and Artificial Intelligence**, Berkeley, v. 8, n. 1, p. 391–414, 1994.

ARAÚJO, A. F. R.; BARRETO, G. de A. Context in Temporal Sequence Processing: A Self-Organizing Approach and Its Application to Robotics. **IEEE Transactions on Neural Networks**, New York, v. 13, n. 1, p. 45–57, 2002.

ASHBY, W. R. **Design for a Brain: The Origin of Adaptive Behaviour**. 2nd ed. London: Chapman & Hall, 1960. 286 p.

ASHBY, W. R. **Uma introdução à cibernética**. 2nd ed. São Paulo: Perspectiva, 1970. 345 p.

BARONE, D. Da física ao neurônio artificial. In: BARONE, D. et al. (Ed.). **Sociedades Artificiais: A Nova Fronteira da Inteligência nas Máquinas**. Porto Alegre: Bookman, 2003. p. 93–126.

BARRETO, G. D. A.; ARAÚJO, A. F. R. Time in Self-Organizing Maps: An Overview of Models. **International Journal of Computer Research**, Huntington, v. 10, n. 2, p. 139–179, 2001.

BASTOS, E. N. F. **Um estudo sobre sistemas sensório-motores biologicamente plausíveis**. 2005. 53 p. Trabalho Individual (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

BERGLUND, E.; SITTE, J. The Parameterless Self-Organizing Map Algorithm. **IEEE Transactions on Neural Networks**, New York, v. 17, n. 2, p. 305–316, 2006.

BITTENCOURT, G. **Inteligência Artificial: Ferramentas e Teorias**. 2. ed. Florianópolis: Ed. da UFSC, 2001. 362 p.

BLACKMORE, J. **Visualizing High-Dimensional Structure with Incremental Grid Growing Neural Network**. 1995. Dissertação (Mestrado) — University of Texas at Austin, Austin.

BLACKMORE, J.; MIIKKULAINEN, R. Visualizing high-dimensional structure with the incremental grid growing neural network. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 1995. **Proceedings...** [S.l.: s.n.], 1995. p. 55–63.

BLANK, D. et al. The Pyro toolkit for AI and robotics. **AI Magazine**, Menlo Park, v. 27, n. 1, p. 39–51, 2006.

BOEKHORST, R. T.; LUNGARELLA, M.; PFEIFER, R. Dimensionality reduction through sensory-motor coordination. In: JOINT INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS AND NEURAL INFORMATION PROCESSING, ICANN/ICONIP, 2003. **Proceedings...** Berlin: Springer, 2003. p. 496–503. (Lecture Notes in Computer Science, v. 2714).

BRISTOL, A. S.; PURCELL, A. L.; CAREW, T. J. Habituation. In: ARBIB, M. A. (Ed.). **The Handbook of Brain Theory and Neural Networks**. 2nd ed. Cambridge, MA: MIT Press, 2003. p. 504–507.

BROOKS, R. A. A robust layered control system for a mobile robot. **IEEE Journal of Robotics and Automation**, New York, v. 2, n. 1, p. 14–23, 1986.

BRUSKE, J.; SOMMER, G. Dynamic Cell Structure learns Perfectly Topology Preserving Map. **Neural Computation**, Cambridge, MA, v. 7, n. 4, p. 845–865, 1995.

BURZEVSKI, V.; MOHAN, C. K. Hierarchical Growing Cell Structures. In: INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 1996. **Proceedings...** [S.l.: s.n.], 1996. v. 3, p. 1658–1663.

BUSHNELL, E. M.; BOUDREAU, J. P. Motor development in the mind: The potential role of motor abilities as a determinant of aspects of perceptual development. **Child Development**, [S.l.], v. 64, n. 1, p. 1005–1021, 1993.

CARPENTER, G. A.; GROSSBERG, S. Adaptive resonance theory. In: ARBIB, M. A. (Ed.). **The Handbook of Brain Theory and Neural Networks**. 2nd ed. Cambridge, MA: MIT Press, 2003. p. 87–90.

CHAPPELL, G. J.; TAYLOR, J. G. The Temporal Kohonen Map. **Neural Networks**, New York, v. 6, n. 3, p. 441–445, 1993.

CLANCEY, W. J. **Situated Cognition: on human knowledge and computer representations**. Cambridge, U.K.: Cambridge University Press, 1997. 426 p.

CLARK, A.; THORNTON, C. Trading spaces: Computation, representation, and the limits of uninformed learning. **Behavioral and Brain Sciences**, [S.l.], v. 20, n. 1, p. 57–90, 1997.

CONNORS, B. W.; BEAR, M. F.; PARADISO, M. A. **Neurociências: Desvendando o Sistema Nervoso**. 2. ed. [S.l.]: Artmed, 2002. 855 p.

- COSTA, F. A.; ROCHA, L. M. Embodied and Situated Cognition. **Artificial Life**, Cambridge, v. 11, n. 1-2, p. 5–11, 2005.
- DENNETT, D. C. **Kinds of Minds**: Towards an Understanding of Consciousness. [S.l.]: Basic Books, 1996. 192 p.
- DESIENO, D. Adding a Conscience to Competitive Learning. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 1988. **Proceedings...** Los Alamitos: IEEE Computer Society, 1988. 1 CD-ROM.
- DITTENBACH, M.; MERKL, D.; RAUBER, A. The Growing Hierarchical Self-Organizing Map. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 2000. **Proceedings...** Los Alamitos: IEEE Computer Society, 2000. 1 CD-ROM.
- DOWNING, K. L. The Predictive Basis of Situated and Embodied Artificial Intelligence. In: CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, GECCO, 2005, Washington. **Proceedings...** New York: ACM Press, 2005. p. 43–50.
- DRESCHER, G. L. **Made-Up Minds**: A Constructivist Approach to Artificial Intelligence. [S.l.]: The MIT Press, 1991. 236 p.
- DUCH, W.; JANKOWSKI, N. Survey of Neural Transfer Functions. **Neural Computing Surveys**, Mahwah, NJ, v. 2, p. 163–212, 1999.
- DUPUY, J.-P. **Nas origens das ciências cognitivas**. São Paulo: Ed. da Universidade Estadual Paulista, 1996. 228 p.
- EDELMAN, G. Building a Picture of the Brain. **Annals of the New York Academy of Sciences**, New York, v. 78, p. 68–89, 1999.
- EDELMAN, G. M. Group selection and phasic reentrant signalling: a theory of higher brain function. In: EDELMAN, G. M.; MOUNTCASTLE, V. B. (Ed.). **The Mindful Brain**: Cortical Organization and the Group-Selective Theory of Higher Brain Function. Cambridge, MA: MIT Press, 1978. p. 51–100.
- EDELMAN, G. M. **Neural Darwinism**: The Theory of Neuronal Group Selection. [S.l.]: Basic Books, 1987.
- EDELMAN, G. M. **The Remembered Present**: A Biological Theory of Consciousness. [S.l.]: Basic Books, 1989.
- EDELMAN, G. M. **Biologia da Consciência**: as Raízes do Pensamento. [S.l.]: Instituto Piaget, 1995.
- EDELMAN, G. M. **Wider Than the Sky**: the phenomenal gift of consciousness. [S.l.]: Yale University Press, 2004.
- ELMAN, J. L. Finding Structure in Time. **Cognitive Science**, Norwood, v. 14, n. 2, p. 179–211, 1990.

ENGEL, P. M. Redes Neurais Artificiais: Uma Visão Geral das suas Potenciais Aplicações. In: FÓRUM DE INTELIGÊNCIA ARTIFICIAL DA REGIÃO SUL, 2., 2001. **Anais...** Canoas: Universidade Luterana do Brasil, 2001. 1 CD-ROM.

ENGEL, P. M. **Conceitos básicos da Inteligência Computacional**. Notas de aula da disciplina CMP121 Redes Neurais do Mestrado em Ciência da Computação, UFRGS. 2005.

ERWIN, E.; OBERMAYER, K.; SCHULTEN, K. Models of Orientation and Ocular Dominance Columns in the Visual Cortex: A Critical Comparison. **Neural Computation**, [S.l.], v. 7, p. 425–486, 1995.

ERWIN, E.; OBERMAYER, K.; SHULTEN, K. Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions. **Biological Cybernetics**, [S.l.], v. 67, p. 47–55, 1992.

FAHLMAN, S. E.; LEBIERE, C. The Cascade-Correlation Learning Architecture. In: TOURETZKY, D. S. (Ed.). **Advances in Neural Information Processing Systems**. Denver: Morgan Kaufmann, 1990. v. 2, p. 524–532.

FIESLER, E. Comparative Bibliography of Ontogenic Neural Networks. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, ICANN, 1994, Sorrento, Italy. **Proceedings...** London: Springer-Verlag, 1994. p. 793–796.

FIKES, R. E.; NILSSON, N. J. **Strips**: a New Approach to the Application of Theorem Proving to Problem Solving. Menlo Park, CA: Stanford Research Institute, 1971.

FLORES, R. Z.; SCHÜLER, J. ao P. S. Estudos sobre o pensamento evolutivo. In: EL-HANI, C. N.; VIDEIRA, A. A. P. **A evolução dos algoritmos mentais**. Rio de Janeiro: [s.n.], 2004.

FLORIAN, R. V. **Autonomous artificial intelligent agents**. 2003. Monografia — Center for Cognitive and Neural Studies, Cluj-Napoca, Romania.

FRANKLIN, S.; GRAESSER, A. Is it an Agent, or Just a Program?: a Taxonomy for Autonomous Agents. In: ECAI, 1996. **Intelligent Agents III, Agent Theories, Architectures, and Languages**: proceedings. Berlin: Springer-Verlag, 1997. p. 21–35. (Lecture Notes in Computer Science, v. 1193).

FRITZKE, B. Growing Cell Structures – a Self-Organizing Network in k Dimensions. In: ALEKSANDER, I.; TAYLOR, J. (Ed.). **Artificial Neural Networks**. Amsterdam, Netherlands: North-Holland, 1992. p. 1051–1056.

FRITZKE, B. Growing Cell Structures – A Self-organizing Network for Unsupervised and Supervised Learning. **Neural Networks**, New York, v. 7, n. 9, p. 1441–1460, 1994.

FRITZKE, B. Growing Grid – a self-organizing network with constant neighborhood range and adaptation strength. **Neural Processing Letters**, Dordrecht, v. 2, n. 5, p. 9–13, 1995.

FRITZKE, B. A Growing Neural Gas Network Learns Topologies. In: TESAURO, G.; TOURETZKY, D. S.; LEEN, T. K. (Ed.). **Advances in Neural Information Processing Systems**. Cambridge, MA: MIT Press, 1995. v. 7, p. 625–632.

FRITZKE, B. Growing Self-organizing Networks – Why? In: EUROPEAN SYMPOSIUM ON ARTIFICIAL NEURAL NETWORKS, 4., 1996. **Proceedings...** Brussels: D-Facto, 1996. p. 61–72.

FRITZKE, B. A self-organizing network that can follow non-stationary distributions. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, ICANN, 1997. **Proceedings...** London: Springer, 1997. p. 613–618.

FRITZKE, B. **Some Competitive Learning Methods**. [S.l.], 1997. 43 p. Disponível em: <<http://www.neuroinformatik.ruhr-unibochum.de/ini/VDM/research/gsn/JavaPaper/>>. Acesso em: 21 out. 2007.

FRITZKE, B. Unsupervised ontogenic networks. In: FIESLER, E.; BEALE, R. (Ed.). **Handbook of Neural Computation**. [S.l.]: Institute of Physics Publishing and Oxford University Press, 1997. p. C2.4:1–C2.4:16.

GIBSON, J. J. **The Ecological Approach to Visual Perception**. Boston: Houghton Mifflin, 1979. 336 p.

GIROSI, F.; JONES, M.; POGGIO, T. Regularization theory and neural networks architectures. **Neural Computation**, Cambridge, v. 7, p. 219–269, 1995.

GUAZZELLI, A. **Do ART 1 ao Fuzzy ARTMAP: Um Estudo Sobre Modelos de Redes Neurais Artificiais Baseados na Teoria da Adaptação Ressonante (ART)**. 1992. 61 p. Trabalho Individual (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

HAGENBUCHNER, M.; SPERDUTI, A.; TSOI, A. C. A Self-Organizing Map for Adaptive Processing of Structured Data. **IEEE Transactions on Neural Networks**, New York, v. 14, n. 3, p. 491–505, 2003.

HARTER, D.; KOZMA, R. Aperiodic Dynamics for Appetitive/Aversive Behavior in Autonomous Agents. In: INTERNATIONAL CONFERENCE ON ROBOTICS & AUTOMATION, 2004. **Proceedings...** [S.l.: s.n.], 2004. v. 3, p. 2147–2152.

HARTER, D.; KOZMA, R. Chaotic Neurodynamics for Autonomous Agents. **IEEE Transactions on Neural Networks**, New York, v. 16, n. 3, p. 565–579, May 2005.

HARTER, D.; KOZMA, R.; GRAESSER, A. C. Models of Ontogenetic Development for Autonomous Adaptive Systems. In: ANNUAL CONFERENCE OF THE COGNITIVE SCIENCE SOCIETY, 23., 2001. **Proceedings...** [S.l.: s.n.], 2001. p. 405–410.

HAYKIN, S. **Redes Neurais: Princípios e Práticas**. 2. ed. São Paulo: Bookman, 2001.

HEBB, D. **The Organization of Behavior**. [S.l.]: John Wiley, 1949.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, Cambridge, v. 9, n. 8, p. 1735–1780, 1997.

HODGE, V. J.; AUSTIN, J. Hierarchical Growing Cell Structures: TreeGCS. **IEEE Transactions on Knowledge and Data Engineering**, New York, v. 13, n. 2, p. 207–218, 2001.

HUNG, C.; WERMTER, S. A Dynamic Adaptive Self-Organising Hybrid Model for Text Clustering. In: INTERNATIONAL CONFERENCE ON DATA MINING, 3., 2003. **Proceedings...** [S.l.: s.n.], 2003. p. 75–82.

HUNG, C.; WERMTER, S. A Constructive Self-Organising Model for A Non-Stationary Environment. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 2005. **Proceedings...** Piscataway, N.J.: IEEE Operation Center, 2005. v. 5, p. 2948–2953.

HURLEY, S. Perception And Action: Alternative Views. **Synthese**, [S.l.], v. 129, n. 1, p. 3–40, 2001.

IZHIKEVICH, E. M. Simple Model of Spiking Neurons. **IEEE Transactions on Neural Networks**, New York, v. 14, n. 6, p. 1569–1572, 2003.

JAMES, D. L.; MIIKKULAINEN, R. SARDNET: A Self-Organizing Feature Map for Sequences. In: TESAURO, G.; TOURETZKY, D. S.; LEEN, T. K. (Ed.). **Advances in Neural Processing Systems**. Cambridge, MA: MIT Press, 1995. v. 7, p. 577–584.

JOLLIFFE, I. T. **Principal Component Analysis**. 2nd ed. New York: Springer, 2002.

K-TEAM: K robots. Yverdon-les-bains, Switzerland: [s.n.], 2002–2006. Disponível em: <www.k-team.com>. Acesso em: 18 ago. 2006.

KANGAS, J. Time-Delayed Self-Organizing Maps. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 1990. **Proceedings...** [S.l.: s.n.], 1990. v. 2, p. 331–336.

KASKI, S.; KANGAS, J.; KOHONEN, T. Bibliography of Self-Organizing Map (SOM) Papers: 1981–1997. **Neural Computing Surveys**, [S.l.], v. 1, p. 102–350, 1998.

KASKI, S.; LAGUS, K. Comparing Self-Organizing Maps. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, ICANN, 1996. **Proceedings...** Berlin: Springer, 1996. p. 809–814. (Lecture Notes in Computer Science, v. 1112).

KOHONEN, T. Physiological Interpretation of the Self-Organizing Map Algorithm. **Neural Networks**, Oxford, U.K., v. 6, n. 7, p. 895–905, 1993.

KOHONEN, T. **Self-Organizing Maps**. New York: Springer-Verlag, 2001.

KOHONEN, T. The Self-Organizing Map. **Proceedings of the IEEE**, Piscataway, v. 78, n. 9, p. 1464–1480, Sept. 1990.

KOVÁCS, Z. L. **Redes Neurais Artificiais: Fundamentos e Aplicações**. 3. ed. São Paulo: Livraria da Física, 2002. 174 p.

KRICHMAR, J. L.; EDELMAN, G. M. Brain-Based Devices for the Study of Nervous Systems and the Development of Intelligent Machines. **Artificial Life**, Cambridge, v. 11, n. 1-2, p. 63–77, Winter–Spring 2005.

KRICHMAR, J. L.; EDELMAN, G. M. Principles Underlying the Construction of Brain-Based Devices. In: KOVACS, T.; MARSHALL, J. A. R. (Ed.). **Adaptation in Artificial and Biological Systems**. Bristol, U.K.: [s.n.], 2006. p. 37–42.

KRUSCHKE, J. K. ALCOVE: An exemplar-based connectionist model of category learning. **Psychological Review**, [S.l.], v. 99, n. 1, p. 22–44, Jan. 1992.

KUNZE, M.; STEFFENS, J. Growing Cell Structure and Neural Gas: Incremental Neural Networks. In: AIHEP WORKSHOP, 4., 1995. **Proceedings...** [S.l.: s.n.], 1995.

LAMBRINOS, D.; SCHEIER, C. **Extended Braitenberg Architectures**. Paris, 1995.

LAMBRINOS, D.; SCHEIER, C. Building Complete Autonomous Agents: A Case Study on Categorization. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS. **Proceedings...** [S.l.: s.n.], 1996. v. 1, p. 170–177.

LANG, R.; WARWICK, K. The Plastic Self Organizing Map. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 2002. **Proceedings....** [S.l.: s.n.], 2002. v. 1, p. 727–732.

LENT, R. **Cem Bilhões de Neurônios: Conceitos Fundamentais de Neurociência**. São Paulo: Atheneu, 2002. 698 p.

LINÅKER, F.; NIKLASSON, L. Time series segmentation using an adaptive resource allocating vector quantization network based on change detection. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 2000. **Proceedings...** [S.l.: s.n.], 2000. v. 6, p. 323–328.

LUNGARELLA, M.; PFEIFER, G. M. R.; SANDINI, G. Developmental robotics: a survey. **Connection Science**, Abingdon, v. 15, n. 4, p. 151–190, 2003.

MARSLAND, S.; NEHMZOW, U.; SHAPIRO, J. A Model of Habituation Applied to Mobile Robots. In: TOWARDS INTELLIGENT MOBILE ROBOTS, 1999, Bristol. **Proceedings...** Bristol: UWE, 1999.

MARSLAND, S.; NEHMZOW, U.; SHAPIRO, J. On-line novelty detection for autonomous mobile robots. **Robotics and Autonomous Systems**, Amsterdam, v. 51, p. 191–206, 2005.

MARSLAND, S.; SHAPIRO, J.; NEHMZOW, U. A self-organising network that grows when required. **Neural Networks**, New York, v. 15, n. 8, p. 1041–1058, 2002.

MARTINETZ, T. M. A neural gas network learns topologies. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, ICANN, 1991. **Proceedings...** Amsterdam: Elsevier Science, 1991. p. 397–402.

MARTINETZ, T. M. Competitive Hebbian learning rule forms perfectly topology preserving maps. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, ICANN, 1993. [S.l.]: Springer, 1993. p. 427–434.

MARTINETZ, T. M.; SCHULTEN, K. J. Topology representing networks. **Neural Networks**, Oxford, v. 7, n. 3, p. 507–522, 1994.

MATURANA ROMECIN, H.; MPODOZIS, J. Percepção: configuração do objeto pela conduta. In: MATURANA ROMECIN, H. et al. **A Ontologia da Realidade**. Belo Horizonte: Ed. da UFMG, 1997. p. 67-76.

MATURANA ROMECIN, H.; VARELA, F. F. **De máquinas e seres vivos: autopoiese : a organização do vivo**. 3. ed. Porto Alegre: Artes médicas, 1997. 138 p.

MATURANA ROMECIN, H.; VARELA, F. F. **A Árvore do Conhecimento: as bases biológicas da compreensão humana**. 2. ed. São Paulo: Palas Athenas, 2001. 288 p.

MCCULLOCH, W. S.; PITTS, W. H. A Logical Calculus of Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, USA, v. 5, p. 115–133, 1943.

MERKL, D. et al. Adaptive Hierarchical Incremental Grid Growing: An architecture for high-dimensional data visualization. In: WORKSHOP ON SELF-ORGANIZING MAPS, 4., 2003. **Proceedings...** [S.l.: s.n.], 2003. p. 293–298.

MICHEL, O. Webots: Professional Mobile Robot Simulation. **International Journal of Advanced Robotics Systems**, Vienna, v. 1, n. 1, p. 39–42, 2004.

MOBILEROBOTS: ActivMedia Robotics. Amherst, NH US: [s.n.], 1995–2006. Disponível em: <www.mobilerobots.com>. Acesso em: 18 ago. 2006.

MÖLLER, R. Perception Through Anticipation: A Behavior-Based Approach to Visual Perception. In: RIEGLER, A. et al. (Ed.). **Understanding Representation in the Cognitive Sciences: does representation need reality?** New York: Kluwer Academic, 1999. p. 184–191.

MOSER, L. D. **Modelo de um Neurônio Diferenciador-Integrador para Representação Temporal em Arquiteturas Conexionistas**. 2004. 98 p. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

MOZER, M. C. Neural net architectures for temporal sequence processing. In: WEIGEND, A.; GERSHENFELD, N. (Ed.). **Predicting the Future and Understanding the Past**. Redwood City: Addison-Wesley, 1993. p. 243–264.

MUÑOZ, M. E. de S. **Proposta de um Modelo de Esquema Cognitivo Sensório-Motor Inspirado na Teoria de Jean Piaget**. 1999. 140 p. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

NEWELL, A.; SIMON, H. A. GPS, a program that simulates human thought. In: FEIGENBAUM, E. A.; FELDMAN, J. (Ed.). **Computers and Thought**. New York: McGraw-Hill, 1963. p. 279–293.

NOLFI, S. Power and the Limits of Reactive Agents. **Neurocomputing**, Amsterdam, v. 42, n. 1-4, p. 119–145, 2002.

NOLFI, S.; PARISI, D. Exploiting the Power of Sensory-Motor Coordination. In: EUROPEAN CONFERENCE ON ADVANCES IN ARTIFICIAL LIFE, ECAL, 5., 1999. **Advances in Artificial Life: proceedings**. Berlin: Springer, 1999. p. 173–182.

OJA, M.; KASKI, S.; KOHONEN, T. Bibliography of Self-Organizing Map (SOM) Papers: 1998–2001 Addendum. **Neural Computing Surveys**, [S.l.], v. 3, p. 1–156, 2002.

OLIVEIRA, F. I. da S. **Affordances**: a relação entre agente e ambiente. 2005. 99 p. Dissertação (Mestrado) — Universidade Estadual Paulista, Marília, São Paulo.

PEROTTO, F. S. **Inteligência Artificial Construtivista**: uma Nova Perspectiva Teórica Para uma Nova Arquitetura de Agente Computacional Inteligente. 2004. 135 p. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

PESSOA JR., O. **Auto-Organização e Complexidade**: Uma Introdução Histórica e Crítica. Notas de aula da disciplina FIS-731, Auto-Organização, Complexidade e Emergência, do Mestrado em Ensino, Filosofia e História das Ciências, UFBA/UEFS. 2001.

PFEIFER, R.; IIDA, F.; BONGARD, J. New Robotics: Design Principles for Intelligent Systems. **Artificial Life**, Cambridge, v. 11, n. 1-2, p. 99–120, Winter–Spring 2005.

PFEIFER, R.; SCHEIER, C. From Perception to Action: The Right Direction? In: FROM PERCEPTION TO ACTION CONFERENCE, 1994. **Proceedings...** Los Alamitos, CA: IEEE Computer Society Press, 1994. p. 1–11.

PFEIFER, R.; SCHEIER, C. Sensory-motor coordination: The metaphor and beyond. **Robotics and Autonomous Systems**, Amsterdam, v. 20, n. 2-4, p. 157–178, 1997.

PFEIFER, R.; SCHEIER, C. **Understanding Intelligence**. Cambridge, MA: The MIT Press, 2001.

PIAGET, J. **A construção do real na criança**. 3. ed. Rio de Janeiro: Zahar, 1979. 360 p.

REEKE, J. G. N.; EDELMAN, G. M. Selective Networks and Recognition Automata. **Annals of the New York Academy of Sciences**, New York, v. 426, p. 181–201, 1984.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. [S.l.]: Campus, 2004. 1040 p.

SANTOS, B. A. **Aspectos Conceituais e Arquiteturais para a Criação de Linhagens de Agentes de Software Cognitivos e Situados**. 2003. 130 p. Dissertação (Mestrado) — Centro Federal de Educação Tecnológica de Minas Gerais.

SCHEIER, C.; LAMBRINOS, D. Categorization in a real-world agent using haptic exploration and active perception. In: INTERNATIONAL CONFERENCE ON SIMULATION OF ADAPTIVE BEHAVIOR, 4., 1996. **From Animals to Animats 4**: proceedings. Cambridge: MIT, 1996. p. 65–74.

SCHEIER, C.; PFEIFER, R. Classification as Sensory-Motor Coordination: A Case Study on Autonomous Agents. In: EUROPEAN CONFERENCE ON ADVANCES IN ARTIFICIAL LIFE, 3., 1995. **Advances in Artificial Life**: proceedings. Berlin: Springer, 1995. p. 657–667.

SCHÜLER, J. P. S. **Inteligência Artificial Popperiana**. 2002. 105 p. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

SETH, A. K.; BAARS, B. J. Neural Darwinism and consciousness. **Consciousness and Cognition**, Homewood, v. 14, n. 1, p. 140–168, 2005.

STRICKERT, M.; HAMMER, B. Self-Organizing Context Learning. In: EUROPEAN SYMPOSIUM ON ARTIFICIAL NEURAL NETWORKS, 2004. **Proceedings...** Bruges: D-Facto, 2004. p. 39–44.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. Cambridge, MA: MIT Press, 1998.

THRUN, S. B.; MITCHELL, T. M. Lifelong robot learning. **Robotics and Autonomous Systems**, Bonn, v. 15, n. 1, p. 24–46, 1995.

VARELA, F. F.; THOMPSON, E.; ROSCH, E. **A mente incorporada: ciências cognitivas e experiência humana**. Porto Alegre: Artmed, 2003. 294 p.

VARSTA, M.; HEIKKONEN, J.; MILLÁN, J. D. R. A Recurrent Self-Organizing Map for Temporal Sequence Processing. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, ICANN, 7., 1997, Lausanne, Switzerland. **Proceedings...** Berlin: Springer, 1997. p. 421–426. (Lecture Notes in Computer Science, v. 1327).

VERSCHURE, P. F. M. J.; ALTHAUS, P. A real-world rational agent: Unifying old and new AI. **Cognitive Science**, Norwood, v. 27, n. 1, p. 561–590, 2003.

VERSCHURE, P. F. M. J.; PFEIFER, R. Categorization, Representations, and the Dynamics of System-Environment Interaction: a case study in autonomous systems. In: INTERNATIONAL CONFERENCE ON SIMULATION OF ADAPTIVE BEHAVIOR, 1992. **From Animals to Animats 2: proceedings**. Cambridge: MIT, 1992. p. 210–217.

VOEGTLIN, T. Context Quantization and Contextual Self-Organizing Maps. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, IJCNN, 2000. **Proceedings...** Los Alamitos: IEEE Computer Society, 2000. 1 CD-ROM.

VOEGTLIN, T. Recursive self-organizing maps. **Neural Networks**, New York, v. 15, n. 8-9, p. 979–991, 2002.

WAZLAWICK, R. **Um Modelo Operatório para a Construção do Conhecimento**. 1993. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis.

WERBOS, P. J. Backpropagation Through Time: What It Does and How to Do It. **Proceedings of the IEEE**, Piscataway, v. 78, n. 10, p. 1550–1560.

WILLIAMS, R. J.; ZIPSER, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. **Neural Computation**, Cambridge, v. 1, p. 270–280, 1989.

ZIEMKE, T. What's that Thing Called Embodiment? **Cognitive Science**, Kidlington, Oxford, v. 27, n. 2, p. 333, 2003. Trabalho apresentado no 25th Annual Meeting of the Cognitive Science Society.

ZREHEN, S.; GAUSSIÉ, P. Why Topological Maps Are Useful for Learning in an Autonomous Agent. In: FROM PERCEPTION TO ACTION CONFERENCE, 1994. **Proceedings...** Los Alamitos, CA: IEEE Computer Society Press, 1994. p. 230–241.

APÊNDICE A FERRAMENTA DE SIMULAÇÃO ROBÓTICA

No decorrer da elaboração desta dissertação, uma ferramenta de simulação robótica foi desenvolvida para auxiliar na modelagem, prototipação e simulação de agentes de *software* situados e corporificados. Esta ferramenta foi usada para elaborar todos os experimentos relatados no Capítulo 6. Neste apêndice, são descritas as características da ferramenta de simulação robótica, bem como os principais componentes oferecidos por ela, a saber: modelos computacionais dos robôs Pioneer e Khepera; uma coleção de sensores proprioceptivos e exteroceptivos; e um conjunto de efetuadores robóticos.

A.1 Introdução

A simulação computacional constitui um passo muito importante na concepção de sistemas inteligentes, pois não há custos para aquisição, implantação e manutenção mecânica ou elétrica de robôs. Ademais, o controle sobre os parâmetros da simulação é total, não há acoplamento entre tempo real e tempo simulado, e os experimentos podem ser repetidos quantas vezes for necessário, permitindo uma melhor análise dos dados. No entanto, apesar dos pontos positivos destacados, é sabido que os simuladores necessariamente realizam simplificações em relação aos fenômenos reais envolvidos, o que impossibilita retratar fidedignamente a dinâmica de um robô físico e de um ambiente realístico.

Como regra geral, a escolha entre utilizar ambientes de simulação ou construir robôs físicos deve ser guiada pelas necessidades de projeto e pelo propósito da pesquisa (PFEIFFER; SCHEIER, 2001, p. 107). Caso o objetivo seja estudar assuntos teóricos (e.g., integração sensorio-motora, formação de conceitos, navegação autônoma, etc.), os simuladores, via de regra, são ferramentas apropriadas por proporcionar um caminho mais rápido, econômico e eficaz na obtenção e avaliação de alguns resultados experimentais. Esse é o foco do presente trabalho e, portanto, optou-se em utilizar um simulador robótico.

Hoje em dia, existe uma grande disponibilidade de aplicações comerciais para simulação robótica. A maioria fornece soluções altamente poderosas, entretanto, com um altíssimo custo associado, tipicamente na ordem de centenas a milhares de dólares (e.g., Webots de Michel, 2004). Muitos pesquisadores e entusiastas da robótica, que não dispõem de recursos financeiros, tentam suprir suas necessidades criando seus próprios simuladores alternativos (e.g., The Pyro Toolkit desenvolvido por Blank et al., 2006). A maior parte deles é concebida sob a Licença Pública Geral (do inglês *General Public License*, ou simplesmente GNU GPL), para que outras pessoas possam usufruir e contribuir para seu desenvolvimento. Contudo, o resultado desses esforços costuma ser restrito a um domínio específico de aplicação e, geralmente, são de difícil uso e expansão.

Em virtude das razões supracitadas, uma nova ferramenta de simulação robótica foi desenvolvida dentro do contexto deste trabalho. O objetivo principal é proporcionar uma

plataforma simples, flexível e de fácil uso, para viabilizar a modelagem, construção e simulação de agentes de *software* situados e corporificados. A ferramenta foi implementada na linguagem de programação Python versão 2.4, a interface gráfica foi codificada em Qt versão 3.3.6 e a geração de gráficos foi realizada através da interface pylab da biblioteca matplotlib versão 0.87. Nenhum componente avançado da linguagem Python foi usado, sendo possível modificar e executar a ferramenta em qualquer sistema operacional que possuir a versão básica do Python instalada (e.g., Windows, Unix e Macintosh OS X).

A.2 Ambiente de simulação

A interface principal da ferramenta de simulação robótica é mostrada na Figura A.1. As principais funcionalidades são fornecidas pelo menu de comandos e pela barra de botões. O menu *File* permite inicializar, carregar e salvar uma simulação. O menu *View* possibilita customizar a área de visualização da simulação. O menu *Options* admite carregar uma *shell* interativa para controlar a simulação por linha de comando (vide Figura A.4) e abrir uma janela para visualizar as mensagens de *log*. Por fim, o menu *Windows* permite selecionar o controlador, os sensores e os efetadores dos robôs para serem inspecionados durante a simulação. Através da barra de botões pode-se iniciar, *resetar*, executar e parar a simulação, determinar a velocidade que a simulação é executada (0 – 1000 ms), e ainda criar vídeos e imagens da simulação realizada para posterior análise ou publicação.

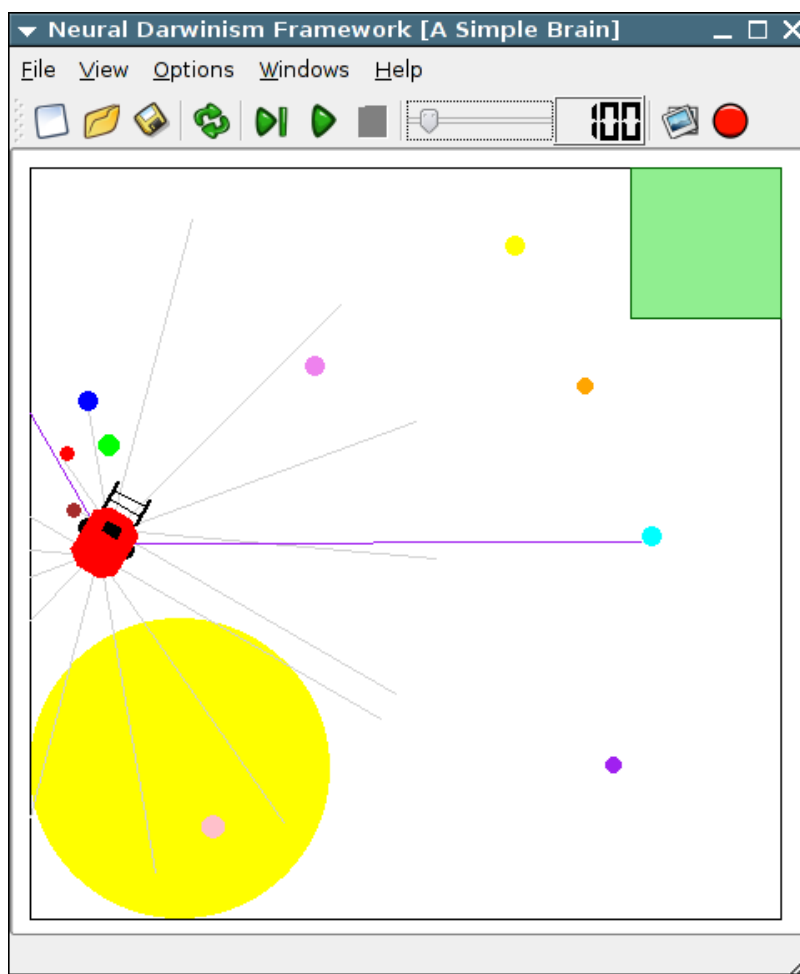


Figura A.1: Interface principal da ferramenta de simulação robótica.

A simulação do ambiente é baseada em um mapa bidimensional. Esse mapa pode ser habitado tanto por recursos quanto por agentes de *software*, referidos doravante como robôs. Os recursos vão desde esferas até fontes de luz. Já os robôs têm determinados sensores, efetadores e uma unidade de controle, onde são estabelecidos os comportamentos ou as tarefas a serem solucionadas pelo robô. A unidade de controle pode ser codificada por uma variedade de formalismos (e.g., redes neurais artificiais, lógica fuzzy, etc.). Antes de iniciar a simulação pode-se calibrar os parâmetros da unidade de controle através de uma interface de usuário e, a cada passo de simulação, pode-se visualizar graficamente as propriedades da unidade de controle. As Figuras A.2 e A.3 mostram um exemplo das interfaces de configuração e de visualização de uma unidade de controle, respectivamente.

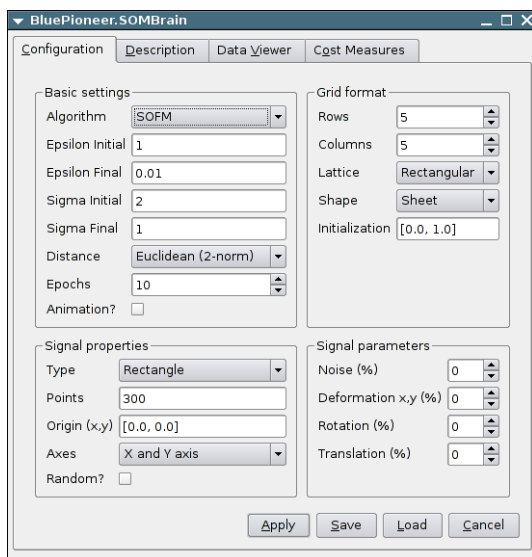


Figura A.2: Exemplo de uma interface de configuração da unidade de controle

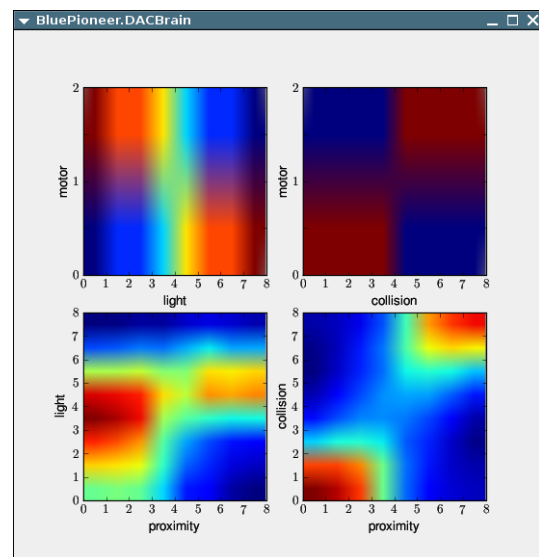


Figura A.3: Exemplo de uma interface de visualização da unidade de controle

Durante a simulação é possível interagir com as propriedades e operações da ferramenta de uma maneira fácil e rápida, conforme é apresentado na Figura A.4. Todas as propriedades e os objetos da simulação (e.g., robôs, recursos) são acessíveis por intermédio da variável `sim`. Por exemplo, o comando `r = sim.getRobot(name='RedPioneer')` atribui à variável `r` a instância de robô cujo nome de identificação é `RedPioneer`.

```

Python shell
Python 2.4.4c0 (#2, Jun 14 2006, 22:35:41)
[GCC 4.1.2 20060613 (prerelease) (Debian 4.1.1-4)] on linux2
Type "sim" for more information.
>>> r = sim.getRobot('RedPioneer')
>>> r.getEnergy()
1000.0
>>> r.isActive()
False
>>> r.stopDevice('camera')
True
>>> r.hasABrain()
True
>>> |

```

Figura A.4: Interface interativa para controle da simulação por linha de comando

A.3 Interface de programação de aplicativos

A ferramenta de simulação robótica fornece uma interface de programação de aplicativos (API) para a concepção de novos robôs, dispositivos, recursos e ambientes. A interface de *software* contém uma série de rotinas, funções e classes, acessíveis somente por programação e na linguagem de programação Python, que permite realizar um aprimoramento da ferramenta e uma expansão de suas funcionalidades básicas.

A.4 Robôs

Na distribuição básica da ferramenta são disponibilizados dois modelos de robôs, juntamente com um conjunto de sensores e efetadores robóticos. As características dos dois modelos de robôs fazem menção às possibilidades encontradas em robôs reais, a fim de reduzir o investimento na etapa de transição da simulação para o experimento prático. A seguir, são descritos em detalhes os principais componentes oferecidos pela ferramenta.

O modelo de robô Pioneer disponibilizado pela ferramenta de simulação robótica (vide Figura A.5) é inspirado na família de robôs móveis Pioneer desenvolvido pela empresa MobileRobots. A Figura A.6 apresenta uma fotografia do Pioneer 3-DX. O robô Pioneer 3-DX é equipado com uma câmera de vídeo, um apanhador de objetos com dois graus de liberdade, dois grupos de oito sonares e um conjunto de sensores de colisão. O robô está montado sobre um eixo de duas rodas independentes que o permite fazer movimentos de translação e rotação. Para maiores informações acerca da família de robôs Pioneer, consulte (MOBILEROBOTS, 1995–2006).

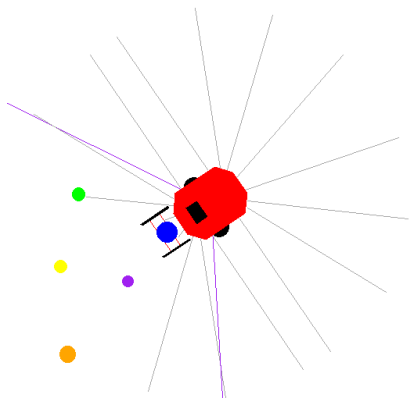


Figura A.5: Modelo de robô Pioneer



Figura A.6: Fotografia do Pioneer 3-DX.
Fonte: (MOBILEROBOTS, 1995–2006)

O modelo de robô Khepera fornecido pela ferramenta de simulação robótica (vide Figura A.7) é inspirado no mini-robô móvel Khepera desenvolvido pela empresa suíça K-Team. A Figura A.8 apresenta uma fotografia do Khepera II. O robô se move através de duas rodas fixas impulsionadas por motores de corrente contínua independentes, e dispõe de módulos suplementares de visão, telemetria e garra mecânica. Em sua circunferência existem oito sensores de infravermelho que detectam a proximidade de obstáculos. Para maiores informações acerca do robô Khepera II, consulte (K-TEAM, 2002–2006).

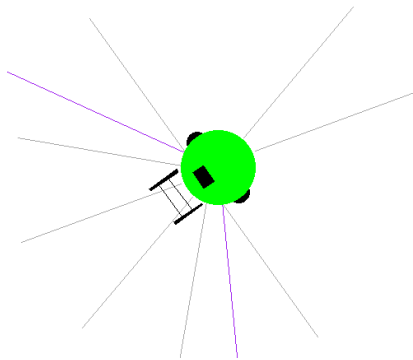


Figura A.7: Modelo de robô Khepera

Figura A.8: Fotografia do Khepera II.
Fonte: (K-TEAM, 2002–2006)

A.5 Sensores robóticos

Uma importante característica de um robô móvel é a capacidade de agir no ambiente a partir de informações obtidas de seu sistema sensorial. A capacidade de sensoriamento é imprescindível para que o robô consiga operar em um ambiente dinâmico. Um sensor é um dispositivo capaz de detectar e quantificar parâmetros específicos através de elementos que transformam uma grandeza física em outra. Basicamente, há dois tipos de sensores: *proprioceptivos* e *exteroceptivos*. A Tabela A.1 apresenta os sistemas de sensoriamento que foram codificados para os dois robôs simulados.

Tabela A.1: Sensores robóticos presentes nos robôs simulados Pioneer e Khepera

Classe	Nome	Descrição
Proprioceptivo	<i>Battery-energy-level</i>	Sensor de carga de bateria para verificar o nível de energia do robô
	<i>Odometry</i>	Sensor para estipular o deslocamento percorrido pelo robô no ambiente
Exteroceptivo	<i>Bumper-sensor</i>	Sensor baseado em contato físico para detectar a colisão do robô com objetos
	<i>Sonar-sensor</i>	Sensor baseado em ultra-som para estimar a distância do robô a objetos
	<i>Light-sensor</i>	Sensor para estimar a intensidade de luz ambiental
	<i>Floor-sensor</i>	Sensor para detectar a superfície de materiais que se encontram no chão do ambiente
	<i>Global-positioning-system</i>	Sistema de posicionamento global para verificar a posição do robô no ambiente
	<i>Camera-sensor</i>	Sistema de visão baseado em uma câmera 2D
	<i>Gripper-sensor</i>	Sensor tátil para detectar objetos situados entre a garra do robô

Os sensores proprioceptivos são usados para medir valores internos do robô como, por exemplo, verificar o nível de carga disponível na bateria ou medir a velocidade e a aceleração das rodas do robô. Já os sensores exteroceptivos são utilizados para determinar

o estado do ambiente – por exemplo, detectar obstáculos, estimar a distância de objetos, determinar a intensidade de luz ambiental ou, até mesmo, verificar a posição real do robô no ambiente através de um sistema de posicionamento global.

A cada passo de simulação é possível visualizar as informações obtidas por cada sensor, conforme ilustrado nas Figuras A.9, A.10, A.11 e A.12 referente ao robô simulado Pioneer. Pode-se constatar que uma colisão à esquerda do robô ocorreu ou está prestes a acontecer, pois existe uma alta ativação nos sensores de colisão (*bumper-sensor*) e de distância (*sonar-sensor*) do grupo lógico *left-all* (vide itens 0, 1, 2, 13, 14 e 15 na Figura A.14). De forma análoga, pode-se verificar uma fonte de luz sendo captada na parte direita do robô, haja vista a forte ativação nos sensores de presença de luminosidade (*light-sensor*) dos grupos lógicos *right* e *back-right* (vide itens 7, 8, 9 e 10 na Figura A.14).

RedPioneer.BumperSensc		
	Distance	Angle
0	0.995983589434	90.0
1	0.0309508918702	65.0
2	-0.023661369066	40.0
3	0.0174073842445	15.0
4	-0.0133075876158	-15.0
5	0.0457526956299	-40.0
6	0.00289389880964	-65.0
7	0.011813829206	-90.0
8	0.0407446226164	-90.0
9	-0.0360384002059	245.0
10	0.0498711380793	220.0
11	0.0256013114031	195.0
12	0.0474866290291	165.0
13	0.0337850555898	140.0
14	-0.0239579136486	115.0
15	0.0273893039895	90.0

Figura A.9: Interface do sensor de contato físico

RedPioneer.SonarSensor		
	Distance	Angle
0	0.921041397045	90.0
1	0.645336864051	65.0
2	0.657713218873	40.0
3	-0.00626828068607	15.0
4	0.047564782875	-15.0
5	0.0144544057641	-40.0
6	-0.00519242357662	-65.0
7	-0.0161508823032	-90.0
8	0.027972172197	-90.0
9	0.009704285868	245.0
10	0.0460891184802	220.0
11	0.410337022262	195.0
12	0.788046560883	165.0
13	0.853727730236	140.0
14	0.856317006248	115.0
15	0.821940043626	90.0

Figura A.10: Interface do sensor de alcance

RedPioneer.LightSensor		
	Distance	Angle
0	-0.00915038148531	90.0
1	0.00713090672311	65.0
2	0.0093072475523	40.0
3	0.024804040868	15.0
4	-0.0292784544074	-15.0
5	-0.046659300806	-40.0
6	-0.0350723124721	-65.0
7	0.435935940158	-90.0
8	0.542994783076	-90.0
9	0.721058347429	245.0
10	0.771146817066	220.0
11	0.732473041303	195.0
12	0.00836246588975	165.0
13	0.0294154704239	140.0
14	0.0100300585667	115.0
15	-0.0325490719513	90.0

Figura A.11: Interface do sensor de detecção de luz

O sistema de visão do robô é composto por duas etapas distintas: aquisição e processamento visual. Em um primeiro momento, o robô obtém uma imagem visual a partir de sua câmera de visão bidimensional (vide Figura A.12). Após, a unidade de controle do robô pode aplicar inúmeros filtros na imagem original (vide Figura A.13), a fim de resolver determinadas tarefas no ambiente (e.g., encontrar esferas da cor verde). O módulo de processamento de imagens foi adquirido do *toolkit* robótico Pyro (BLANK et al., 2006).

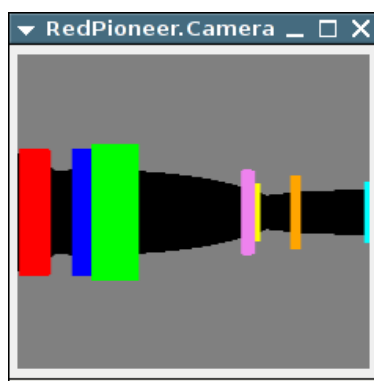


Figura A.12: Interface do sistema de visão baseado em uma câmera 2D

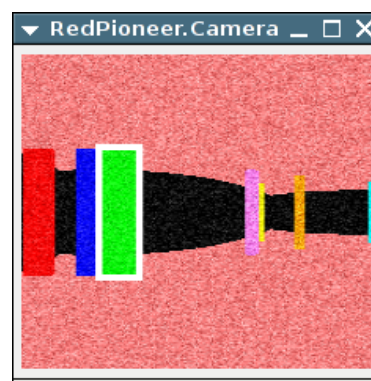


Figura A.13: Imagem obtida após aplicar uma série de filtros arbitrários

As Figuras A.14 e A.15 apresentam um diagrama esquemático da morfologia dos robôs Pioneer e Khepera, respectivamente. Os números indicam as respectivas posições dos sensores exteroceptivos presentes nos robôs simulados.

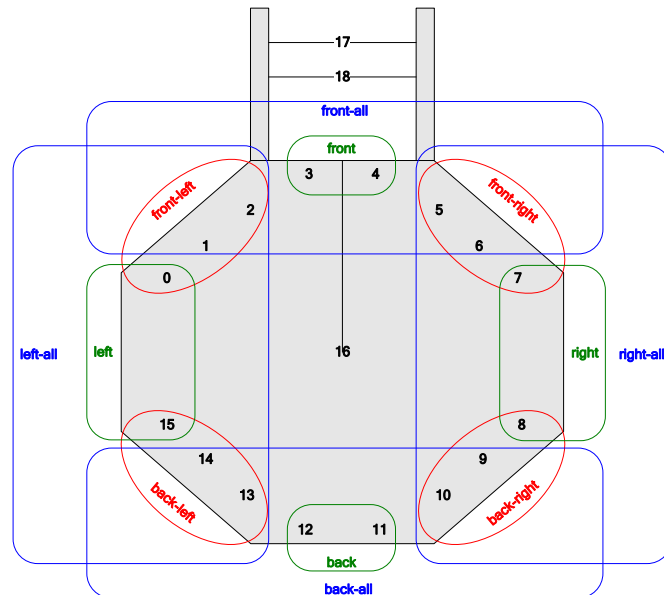


Figura A.14: Morfologia do robô Pioneer. Os sensores *bumper-sensor*, *sonar-sensor* e *light-sensor* estão localizados nas posições $\{0..15\}$. O sistema de visão tem a faixa de operação no intervalo $[1, 6]$. Os sensores *floor-sensor* e *global-positioning-system* estão situados na posição 16. Os sensores da garra estão dispostos nas posições 17 e 18.

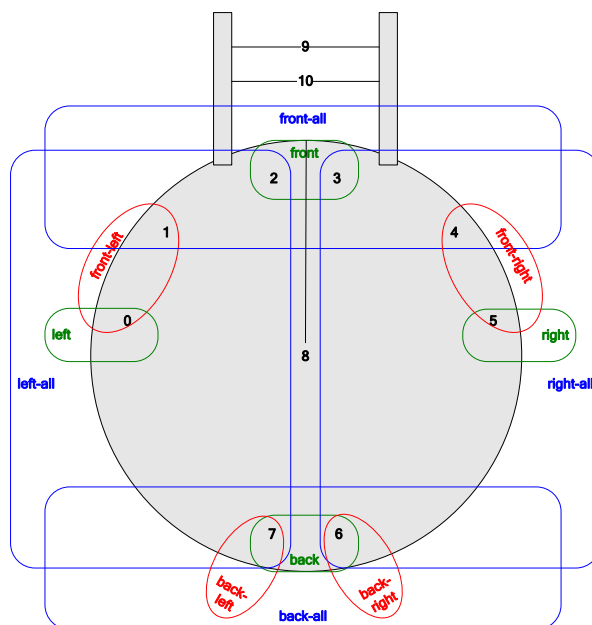


Figura A.15: Morfologia do robô Khepera. Os sensores *bumper-sensor*, *sonar-sensor* e *light-sensor* estão localizados nas posições $\{0..7\}$. O sistema de visão tem a faixa de operação no intervalo $[1, 4]$. Os sensores *floor-sensor* e *global-positioning-system* estão situados na posição 8. Os sensores da garra estão dispostos nas posições 9 e 10.

De acordo com o princípio da redundância discutido na Seção 2.3.6.2, os sensores devem estar posicionados no robô de tal forma que exista uma sobreposição espacial nas informações adquiridas do meio. Por esse motivo, os múltiplos sensores dos robôs simulados estão sobrepostos uns aos outros. Por exemplo, na posição 1 há quatro tipos de sensores exteroceptivos em operação: *bumper-sensor*, *sonar-sensor*, *light-sensor* e *visual-sensor*. Por conveniência, aqueles sensores que têm direções similares são reunidos em grupos lógicos. Por exemplo, no robô simulado Pioneer os sensores de número 0, 1 e 2 pertencem aos grupos lógicos *front-left* e *left-all*.

A.6 Efetadores robóticos

Efetador robótico é um termo genérico que define qualquer componente que permita a um robô exercer alguma ação no ambiente, como, por exemplo, deslocar-se a algum local ou manusear determinados tipos de objetos. Na literatura, o termo efetador robótico muitas vezes é confundido com o termo atuador robótico. Atuador robótico é meramente o mecanismo que habilita o efetador a realizar a sua ação e tipicamente são motores elétricos, pistões hidráulicos e pistões pneumáticos. Os robôs simulados da ferramenta de simulação proposta dispõem de três efetadores, conforme está descrito na Tabela A.2.

Tabela A.2: Efetadores robóticos presentes nos robôs simulados Pioneer e Khepera

Nome	Descrição
<i>Wheel-motors</i>	Dois motores independentes acoplados às rodas do robô que o possibilita realizar movimentos de translação (para frente e para trás) e rotação (giro no sentido horário e anti-horário) ao mesmo tempo
<i>Camera-control</i>	Um controlador de câmera de vídeo que permite o robô ampliar e modificar o ângulo de visão da imagem captada
<i>Gripper-control</i>	Uma garra com uma única articulação (em inglês, <i>Degree of Freedom</i> , DOF) para o robô conseguir manipular os objetos presentes no ambiente

A.7 Considerações do apêndice

Este apêndice descreveu a ferramenta de simulação robótica desenvolvida dentro do contexto deste trabalho de mestrado. Esta ferramenta consiste em um *framework* suficientemente genérico para modelagem, prototipação e simulação robótica e está fortemente embasada nos princípios de agentes de *software* situados e corporificados.

Com o intuito de pesquisar a sua aplicabilidade, foram elaborados variados estudos de caso (não descritos no texto) com os seguintes métodos: controle adaptativo distribuído¹ (DAC), arquitetura estendida de Braitenberg² (EBA), arquitetura de subsunção³ e coordenação sensorio-motora⁴ (SMC). Estas simulações não tiveram caráter validatório, todavia buscaram aplicar as técnicas e mecanismos difundidos na literatura em um ambiente fictício, a título de experimentação e análise de resultados. Os resultados obtidos com os experimentos de simulação foram satisfatórios, demonstrando o potencial e a viabilidade técnica da ferramenta, e serão motivo de uma publicação futura.

¹PFEIFER; SCHEIER, 2001, p. 152–165.

²Ibid., p. 181–197.

³Ibid., p. 199–255.

⁴Ibid., p. 408–417.