Alexandre da Silva Simões

Aprendizado não-supervisionado em redes neurais pulsadas de base radial

Texto apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia Elétrica.

Alexandre da Silva Simões

Aprendizado não-supervisionado em redes neurais pulsadas de base radial

Texto apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Engenharia Elétrica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof.^a Livre-Docente
Anna Helena Reali Costa

Ficha Catalográfica

Simões, Alexandre da Silva

Aprendizado não-supervisionado em redes neurais pulsadas de base radial. São Paulo, 2006. 160 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Redes Neurais. 2. Aprendizado de Máquina. 3. Visão Computacional. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. Título.

Folha de Aprovação

Alexandre da Silva Simões

Aprendizado não-supervisionado em redes neurais pulsadas de base radial

Texto apresentado à Escola Politécnica
da Universidade de São Paulo para
obtenção do Título de Doutor em Enge-
nharia Elétrica.

Área de concentração:
Sistemas Digitais

Aprovado em: 07/04/2006

Banca Examinadora:

Prof. Dr. Adrião Duarte Dória Neto
UFRN – Universidade Federal do Rio Grande do Norte

Prof.^a Dr.^a Anna Helena Reali Costa
EP-USP – Escola Politécnica da Universidade de São Paulo

Prof. Dr. Emilio Del Moral Hernandez
EP-USP – Escola Politécnica da Universidade de São Paulo

Prof. Dr. Hae Yong Kim
EP-USP – Escola Politécnica da Universidade de São Paulo

Prof.^a Dr.^a Roseli Aparecida Francelin Romero
ICMC-USP – Instituto de Ciências Matemáticas e de Computação da
Universidade de São Paulo

“Havendo jardineiro, cedo ou tarde um jardim aparecerá.”

(Rubem Alves)

A Hermínia Russo.

In memoriam.

Agradecimentos

Ao criador de todas as coisas, por permitir a vida e a inteligência, e por me conceder a inquietude e o questionamento.

A meus pais, Maria José e Walsey, que sempre aceitaram incondicionalmente minhas metas, e com amor tornaram possível alcançá-las.

A Iowana e Haíssa, companheiras nas alegrias e dificuldades desta caminhada, pelo afeto, pelo carinho e pela amizade. Da mesma forma a Adalberto e Edna Cardarelli pela acolhida e pelo apoio em todas as ocasiões.

Aos amigos da USP – especialmente a Bianchi, Valguima, Valdinei, Antônio e Diana – por dividirem comigo o prazer da caminhada. Em particular ao amigo Jomi pelo trabalho realizado no modelo Latex para teses e dissertações da Escola Politécnica, aqui utilizado. Da mesma forma a Fátima, que sempre me socorreu nos momentos difíceis.

A Esther, pelas discussões, pelas sugestões, pelo apoio, pela ajuda de todas as formas durante esses últimos passos e, principalmente, por sua estrela. Sem a qual este trabalho não teria sido possível.

Aos amigos do campus de Sorocaba da UNESP – especialmente a Antônio, Fernando, Rogério, José Arnaldo, João, Ronaldo e Márcio – pelo apoio e confiança irrestritos.

À PROPP (pró-reitoria de pós-graduação e pesquisa) da UNESP pelo auxílio parcial através do programa de capacitação docente.

Ao professor Jaime Simão Sichman, pela amizade franca e pelas preciosas contribuições ao longo de quase 8 anos.

A Anna Helena Reali Costa, pelo primor na tarefa de ensinar e pelo excepcional saber. Com quem tive – e tenho sempre – o privilégio de aprender. Que não me censurou nunca e me apoiou sempre. Que me presenteou com conhecimento e paixão pelo conhecimento. Para com quem terei a eterna dívida dos que têm seus horizontes expandidos.

Resumo

Redes neurais pulsadas – redes que utilizam uma codificação temporal da informação – têm despontado como uma nova e promissora abordagem dentro do paradigma conexionista emergente da ciência cognitiva. Um desses novos modelos é a rede neural pulsada de base radial, capaz de armazenar informação nos tempos de atraso axonais dos neurônios e que comporta algoritmos explícitos de treinamento. A recente proposição de uma sistemática para a codificação temporal dos dados de entrada utilizando campos receptivos gaussianos tem apresentado interessantes resultados na tarefa do agrupamento de dados (*clustering*). Este trabalho propõe uma função para o aprendizado não supervisionado dessa rede, com o objetivo de simplificar a sistemática de calibração de alguns dos seus parâmetros-chave, aprimorando a convergência da rede neural pulsada no aprendizado baseado em instâncias. O desempenho desse modelo é avaliado na tarefa de classificação de padrões, particularmente na classificação de pixels em imagens coloridas no domínio da visão computacional.

Abstract

Pulsed neural networks – networks that encode information in the timing of spikes – have been studied as a new and promising approach in the artificial neural networks paradigm, emergent from cognitive science. One of these new models is the pulsed neural network with radial basis function, a network able to store information in the axonal propagation delay of neurons. Recently, a new method for encoding input-data by population code using gaussian receptive fields has showed interesting results in the clustering task. The present work proposes a function for the unsupervised learning task in this network, which goal includes the simplification of the calibration of the network key parameters and the enhancement of the pulsed neural network convergence to instance based learning. The performance of this model is evaluated for pattern classification, particularly for the pixel colors classification task, in the computer vision domain.

Lista de Figuras

2.1	Representação de códigos utilizados em redes neurais pulsadas.	13
2.2	Alguns dos principais modelos de neurônios utilizados em RNPs e sua classificação segundo sua forma mais comum de uso.	17
2.3	Representação esquemática do modelo do neurônio de Hodgkin-Huxley.	18
2.4	Representação esquemática do modelo do neurônio integra-e-dispara.	19
2.5	Diagrama geral do modelo do neurônio pulsado.	21
2.6	Curva típica de ϵ_{ij}	22
2.7	Influência de disparos dos neurônios pré-sinápticos i_1 e i_2 na variável de estado $\mu_j(t)$ de um neurônio j	23
2.8	A refração γ_j do neurônio j após atingir o limiar de disparo ϑ_j	23
2.9	Diagrama esquemático do modelo do neurônio de McCulloch-Pitts.	25
2.10	Diagrama esquemático da variável de estado $u_j(t)$ do neurônio j realizando a comparação de três valores.	26
2.11	Um neurônio SRM implementando a função típica de um neurônio RBF.	27
2.12	Arquitetura básica da rede de SRM-RBFs.	29
2.13	Arquitetura da rede e conexão entre os neurônios.	30
2.14	Campos receptivos gaussianos.	31
2.15	Potencial de um neurônio da camada de saída recebendo contribuições de todas as sinapses.	32
2.16	Modelo da rede neural pulsada.	35
3.1	Disparo de um NP com excitação pré-sináptica linear.	38
3.2	Aproximação linear do potencial do neurônio no momento de seu disparo.	42

3.3	Gráfico da função de aprendizado gaussiana.	45
3.4	Adaptação dos pesos sinápticos e potencial através do algoritmo de aprendizado.	46
3.5	Padrão presente no vetor de pesos sinápticos de um neurônio após o aprendizado.	48
3.6	Diagrama esquemático do neurônio pulsado para implementação do aprendizado por reforço.	50
3.7	Aprendizado por reforço em uma rede neural pulsada.	51
4.1	A superfície de decisão típica do algoritmo KNN.	57
4.2	Evolução típica dos centros dos grupos para o algoritmo <i>k-means</i>	60
4.3	Evolução típica dos centros dos grupos para o algoritmo <i>k-means</i> com um número inadequado de classes.	60
4.4	Arquitetura de uma rede com função de base radial.	63
5.1	Influência dos campos receptivos na separação dos pontos.	73
5.2	Somatória dos valores das latências na geração dos pulsos para campos receptivos gaussianos.	74
5.3	Condições iniciais da RNP-SRM-RBF.	76
5.4	Influências do parâmetro h na discretização dos tempos de disparo do NP.	77
5.5	Generalizações adversas de uma RNP.	78
5.6	Simulação da adaptação dos pesos sinápticos e potencial através do algoritmo de aprendizado.	80
5.7	Simulação da influência do parâmetro b na somatória dos pesos sinápticos e no tempo de disparo dos neurônios.	82
5.8	Implicações de b no processo competitivo dos neurônios.	84
5.9	Implicação de uma soma constante dos pesos sinápticos no processo competitivo.	85
5.10	Falha em processo competitivo de uma RNP com somatória dos pesos sinápticos constante.	86
5.11	Sucesso em processo competitivo de uma RNP com somatória decrescente nos pesos sinápticos.	87

5.12	Valores discretizados da função de aprendizado.	88
5.13	Tempos de disparo de um neurônio da camada de saída para trei- namentos com e sem sucesso.	90
5.14	Padrão de pesos sinápticos de um neurônio após o treinamento. . .	91
6.1	Função de aprendizado proposta para a rede neural pulsada. . . .	95
6.2	Comparativo entre as funções de aprendizado gaussiana e <i>tanh</i> . . .	95
6.3	Integral sobre a função de aprendizado <i>tanh</i>	96
6.4	Curva exibindo os valores de b que tornam nula a integral da função de aprendizado.	98
6.5	Valores máximo e mínimo para a estimativa de ϑ	99
6.6	Comparação do potencial de um neurônio com convergência ina- dequada durante o processo de aprendizado.	102
6.7	Diferentes configurações de W para avaliação do processo de con- vergência.	103
8.1	Processo de convergência para o EXP-01.	114
8.2	Tempos de disparo dos neurônios da rede no EXP-01.	115
8.3	Resultados do EXP-01.	116
8.4	Melhores centros obtidos com o algoritmo <i>k-means</i> no EXP-01. . .	117
8.5	Processo de convergência para o EXP-02.	120
8.6	Tempos de disparo dos neurônios da rede no EXP-02.	121
8.7	Resultados do EXP-02.	122
8.8	Melhores centros obtidos com o algoritmo <i>k-means</i> para o EXP-02.	123
8.9	Processo de convergência para o EXP-03.	126
8.10	Tempo de disparo dos neurônios da rede no EXP-03.	127
8.11	Resultados do EXP-03.	128
8.12	Melhores centros obtidos com o algoritmo <i>k-means</i> para o EXP-03.	129
8.13	Processo de convergência para o EXP-04.	132
8.14	Tempos de disparo dos neurônios da rede no EXP-04.	133
8.15	Resultados do EXP-04.	134

8.16	Melhores centros obtidos com o algoritmo <i>k-means</i> para o EXP-04.	135
9.1	Resultado da classificação de cores no experimento IMG-01.	139
9.2	Dados da simulação do experimento IMG-01.	140
9.3	Imagens classificadas pela RNP no experimento IMG-01.	141
9.4	Resultado da classificação de cores no experimento EXP-02.	145

Lista de Tabelas

4.1	Matriz de confusão para várias classes.	66
5.1	Quadro resumo da RNP-SRM-RBF com campos receptivos gaussianos e aprendizado não-supervisionado sob a ótica do AM. . . .	92
6.1	Sistemática proposta para o aprendizado na RNP-SRM-RBF . . .	94
7.1	<i>Framework</i> utilizado nos experimentos realizados.	109
8.1	Documentação do experimento EXP01.	113
8.2	Quadro comparativo entre a RNP e o <i>k-means</i> no EXP-01.	117
8.3	Documentação do experimento EXP02.	119
8.4	Quadro comparativo entre a RNP e o <i>k-means</i> no EXP-02.	123
8.5	Documentação do experimento EXP03.	125
8.6	Quadro comparativo entre a RNP e o <i>k-means</i> no EXP03.	129
8.7	Documentação do experimento EXP04.	131
8.8	Quadro comparativo entre a RNP e o <i>k-means</i> no EXP04.	135
9.1	Documentação do experimento IMG-01.	138
9.2	Avaliação do desempenho da RNP no experimento IMG-01.	142
9.3	Documentação do experimento IMG-02.	144
9.4	Avaliação do desempenho da RNP no experimento IMG-02.	146

Lista de Algoritmos

4.1	Classificação segundo o <i>k-vizinho mais próximo</i> básico	56
4.2	<i>k-means</i> básico	58
4.3	Agrupamento hierárquico aglomerativo básico	65
4.4	Agrupamento hierárquico divisivo básico	65

Lista de Abreviaturas

- ABI** Aprendizado Baseado em Instâncias
- AM** Aprendizado de Máquinas
- IA** Inteligência Artificial
- CSO** Oscilador Pulsado Caótico – *Chaotic Spiking Oscillator*
- CPCN** Rede Caótica de Pulso Acoplado – *Chaotic Pulse-Coupled Network*
- KNN** K-vizinho mais próximo – *K-nearest neighbor*
- LEGION** Rede Osciladora Localmente Excitatória, Globalmente inibitória – *Locally Excitatory, Globally Inhibitory Oscillatory Network*
- LVQ** Aprendizado da Quantificação Vetorial – *Learning Vector Quantization*
- MLP** Perceptron Multicamada – *Multilayer Perceptron*
- MSE** Erro Médio Quadrático – *Mean Square Error*
- NBR** Neurônio de Base Radial
- NP** Neurônio Pulsado
- PCNN** Rede Neural de Pulso Acoplado – *Pulse-Coupled Neural Network*
- RBF** Função de Base Radial – *Radial Basis Function*
- RNAs** Redes Neurais Artificiais
- RNPs** Redes Neurais Pulsadas
- ROC** Característica de Operação do Receptor – *Receiver Operating Characteristic*
- SN** Neurônio Pulsado – *Spiking Neuron*
- SOM** Mapas Auto-Organizados – *Self-Organized Maps*
- SRM** Modelo de Resposta Pulsada – *Spiking Response Model*
- VC** Visão Computacional

Lista de Símbolos

Conjuntos, vetores e coleções:

$\mathcal{A}$ é uma coleção de ações;

$\mathcal{B}$ é uma coleção de observações;

$\mathcal{E}$ é uma coleção de estados;

$\mathcal{H}$ é o conjunto das hipóteses para aproximar a função-alvo;

I é o conjunto dos neurônios da camada de entrada;

J é o conjunto dos neurônios da camada de saída;

$\mathcal{N}$ é o conjunto de neurônios;

$\mathcal{O}$ é o conjunto dos neurônios da camada oculta;

$\mathcal{P}$ é o conjunto de proposições;

$\mathbb{R}$ é o conjunto dos números reais;

$\mathcal{S}$ é o conjunto de sinapses;

T é o conjunto dos tempos de disparo de um neurônio;

$\mathcal{T}$ é um conjunto de tarefas;

$\mathcal{V}$ é um conjunto de valores;

W é o conjunto de pesos sinápticos;

X é o conjunto de exemplos de entrada em $\mathbb{R}^n$;

Y é o conjunto de exemplos de saída em $\mathbb{R}^m$;

Δ_k é o vetor com os centros do k -ésimo neurônio ou classe;

Δk é a taxa de variação da grandeza k ;

Γ_k é o conjunto dos neurônios pré-sinápticos ao neurônio k ;

Γ^k é o conjunto dos neurônios pós-sinápticos ao neurônio k ;

Γ é a estatística de Hubert;

Letras arábicas:

a é o incremento dos pesos sinápticos na função de aprendizado $\tanh$;

b é o decremento dos pesos sinápticos na função de aprendizado;

c é recuo no tempo da função de aprendizado;

c_k é o centro do grupo k ou o centro da gaussiana k ;

d é o número de sinapses existentes entre o neurônio i_k e o neurônio j ;

$d(.,.)$ é a distância entre os argumentos;

e é um estado, com $e \in \mathcal{E}$;

$f(.)$ é uma função genérica sobre o argumento;

g é o número de gaussianas associadas a uma das dimensões de entrada;

h é a altura das gaussianas presentes no campo receptivo;

$h_{i,j}$ é uma função de define uma vizinhança topológica entre os neurônios i e j ;

i_k é o k -ésimo neurônio da camada de entrada;

j_k é o k -ésimo neurônio da camada de saída;

m é o número de dimensões das instâncias de saída;

n é o número de dimensões das instâncias de entrada;

n_k é um número associado ao conceito k ;

o_k é k -ésimo neurônio da camada oculta $\mathcal{O}$;

p é um padrão;

$\bar{p}_k$ é a taxa média de disparo de pulsos segundo o critério k ;

s é uma sinapse, com $s \in \mathcal{S}$;

t é o tempo;

$t_k^{(u)}$ é o tempo do u -ésimo disparo do neurônio k ;

$\tilde{t}_k^{(u)}$ é o tempo desejado para o u -ésimo disparo do neurônio k ;

w_{ij}^k é o peso sináptico da k -ésima sinapse entre os neurônios i e j , com $w_{ij}^k \in W$;

x é uma instância de entrada $x \in X$;

x_k é o k -ésimo atributo da instância de entrada $x \in X$;

y é uma instância de saída $y \in Y$;

$L(.)$ é uma função de aprendizado;

Letras gregas:

α é uma constante de aprendizado;

β é a taxa de generalização da função de aprendizado $L(t)$;

γ_k é a função que modela a refração do neurônio k ;

$\mathcal{D}$ é uma medida de desempenho;

δ_{ij}^k é o atraso de propagação axonal da k -ésima sinapse entre os neurônios i e j ;

ϵ é uma experiência;

ϵ_{ij}^k é a função que modela a influência dos pulsos recebidos pela k -ésima sinapse entre os neurônios i e j no potencial do neurônio j ;

η é a taxa de aprendizado;

κ é um *Kernel*;

κ_e é a constante de estabilização da somatória de pesos sinápticos;

λ é o intervalo no qual a função de aprendizado $L(t)$ é maior que zero;

$\mu_k(t)$ é o potencial do neurônio k no instante t ;

ω é um conceito por detrás de um conjunto de observações;

ϕ é o ângulo existente no potencial do NP em aprendizagens inadequadas;

$\varphi(.)$ é uma função genérica sobre o argumento;

π é uma política;

ψ é uma medida de desempenho;

$\rho_{ij}(t)$ é o estímulo recebido pelo neurônio j do neurônio i no tempo t ;

σ_k^2 é o desvio padrão da k -ésima curva gaussiana ou de uma distribuição;

σ_k é a variância da k -ésima curva gaussiana ou de uma distribuição;

τ_m é a constante que modela o crescimento da contribuição de uma sinapse na função ϵ_{ij}^k ;

τ_r é a constante que modela o crescimento da refração γ ;

τ_s é a constante que modela o decaimento da contribuição de uma sinapse na função ϵ_{ij}^k ;

θ é o ângulo estimado para o crescimento do potencial de um neurônio da camada de saída;

v é um valor, com $v \in \mathcal{V}$;

ϑ_k é o limiar de disparo do neurônio k ;

$\hat{\vartheta}_k$ é o limiar de disparo estimado do neurônio k ;

Sumário

Lista de figuras

Lista de tabelas

Lista de algoritmos

Lista de abreviaturas

Lista de símbolos

1	Introdução	1
1.1	RNAs na VC	2
1.2	Das RNAs às RNPs	2
1.3	Por que RNPs?	3
1.4	Das RNPs às RNP-RBFs	4
1.5	Objetivos e contribuições	5
1.6	Organização do trabalho	6
I	Conceitos Fundamentais	8
2	Redes Neurais Pulsadas	9
2.1	Códigos	10
2.1.1	Taxa média de disparos	10
2.1.2	Latência (ou atraso)	12
2.1.3	Fase	12
2.1.4	Sincronia	12

2.2	Histórico de RNA	12
2.3	Modelos de neurônios	16
2.3.1	Hodgkin-Huxley	17
2.3.2	Integra-e-dispara	19
2.3.3	SRM	20
2.3.4	Comparativo com outros modelos	24
2.4	Modelos de rede	27
2.5	A RNP-SRM-RBF	28
2.5.1	Arquitetura básica	29
2.5.2	Arquitetura com desdobramento das sinapses	29
2.5.3	Codificação dos dados de entrada	30
2.5.4	Processamento típico	32
2.6	Resumo	33
3	Aprendizado em Redes Neurais Pulsadas	36
3.1	Aprendizado supervisionado	37
3.1.1	Lei de Hebb	37
3.1.2	Retropropagação do erro	39
3.2	Aprendizado não supervisionado	44
3.2.1	Antecipação do disparo	45
3.2.2	Deslocamento do vetor de pesos	47
3.2.3	Padrões presentes no vetor de pesos sinápticos	47
3.2.4	Sensibilidade ao peso máximo	48
3.2.5	Auto-inibição	49
3.3	Aprendizado por reforço	49
3.3.1	Modelo de neurônio	50
3.3.2	Modelo de rede	50
3.3.3	Aprendizado	51

3.4	Resumo	53
4	Aprendizado Baseado em Instâncias	54
4.1	k-vizinho mais próximo	56
4.2	k-means	57
4.3	Aprendizado da Quantificação Vetorial	60
4.4	SOM	61
4.5	Redes de base radial	63
4.6	Algoritmos hierárquicos	64
4.7	Medidas de desempenho	65
4.8	Resumo	68
II	Aprendizado Proposto	69
5	Revisitando a RNP-SRM-RBF	70
5.1	Posicionamento entre as técnicas	71
5.2	Codificação temporal de Bohte	72
5.3	Modelo de Natschläger	78
5.4	RNP-SRM-RBF com campos receptivos gaussianos	81
5.5	Possíveis novos horizontes	89
5.6	Resumo	89
6	Concepção do Aprendizado	93
6.1	Função de aprendizado <i>tanh</i>	94
6.2	Calibração do ponto de estabilidade	96
6.3	Generalização via escalonamento sináptico	98
6.4	Calibração do limiar ϑ	99
6.5	Estimação do peso máximo	100
6.6	Uso de função heurística	101

6.7	Critério de parada	104
III	Resultados Experimentais	105
7	Materiais e Métodos	106
7.1	Implementação	106
7.2	Metodologia de testes	106
7.3	Documentação	108
8	Estudo de casos: agrupamentos 3D	110
8.1	Experimento EXP-01	111
8.2	Experimento EXP-02	118
8.3	Experimento EXP-03	124
8.4	Experimento EXP-04	130
9	Estudo de casos: classificação de cores	136
9.1	Experimento IMG-01	137
9.2	Experimento IMG-02	143
10	Conclusão	147
10.1	Sumário das contribuições	148
10.2	Direções futuras	149

1 Introdução

“O conceito de inteligência é um tanto vago, particularmente com referência a uma máquina. Contudo, definir vários tipos de comportamento geralmente associados com inteligência não é difícil. Algumas características vêm imediatamente à mente:

- i. a habilidade de extrair informação pertinente a partir de um fundo de detalhes irrelevantes;*
- ii. a capacidade de aprender a partir de exemplos e generalizar este conhecimento de forma a aplicá-lo em novas e diferentes circunstâncias;*
- iii. a habilidade de fazer inferências a partir de informações incompletas.”*

A idéia central presente nas definições de comportamentos inteligentes de Gonzales e Woods (1992, p. 571) pode ser colocada da seguinte forma: atribui-se a um agente inteligente a capacidade de realizar observações do ambiente e, a partir delas, inferir conceitos que possam ser utilizados em novas situações. Essas habilidades de *reconhecer padrões* e *aprender* parecem ser inatas nos sistemas biológicos mais complexos.

inteligência
artificial

O papel que o reconhecimento de padrões e o aprendizado desempenham na atividade considerada inteligente justifica seu estudo no âmbito da Inteligência Artificial (IA), que pode ser definida como o campo de estudo da computação que torna possível *percepção*, *raciocínio* e *ação* (WINSTON, 1992).

visão
computa-
cional

A Visão Computacional (VC) tem se preocupado com a tarefa de identificar, extrair e interpretar automaticamente padrões do ambiente a partir de informações visuais. As informações visuais constituem uma das mais ricas formas de percepção, e classificar padrões automaticamente com robustez no domínio da VC tem se mostrado, ao longo dos anos, uma tarefa extremamente complexa.

1.1 RNAs na VC

Um dos paradigmas de maior sucesso nesta empreitada é o assim chamado *paradigma conexionista*. As Redes Neurais Artificiais (RNAs) têm demonstrado aptidão para a classificação devido a algumas características inerentes: *i*) processamento distribuído; *ii*) capacidade de generalização; e *iii*) tolerância a ruídos.

A aplicação de RNAs à visão computacional tem sido objeto de estudo do autor e seu grupo de trabalho nos últimos anos. Em estudos prévios, ferramentas para a classificação de pixels em imagens utilizando o atributo *cor* foram investigadas em diferentes espaços de cores (SIMões, 2000), (SIMões; COSTA, 2000a), (SIMões; COSTA, 2000b), (SIMões; COSTA, 2001). Em particular o Perceptron Multicamada – *Multilayer Perceptron* (MLP) foi objeto de estudo.

Esta modelagem se mostrou bastante apropriada para situações onde as definições de cores – que naturalmente envolvem noções de ordem física e psicológica difíceis de modelar – não são claramente estabelecidas. Particularmente bons resultados foram obtidos em aplicações como o futebol de robôs (SIMões; COSTA, 2000c) e a classificação de frutas baseada em padrões visuais (SIMões, 2002), (SIMões; COSTA, 2003).

Embora a aplicação de RNAs para a classificação da cor de pixels em imagens mostre-se eficiente, o uso dessas redes precisa ser revisto para tarefas mais complexas, como por exemplo o reconhecimento de objetos em imagens baseado em suas características de cor. A abordagem conexionista tem apresentado grandes dificuldades neste contexto, dado principalmente o problema da invariância dos objetos à rotação, translação e escala da imagem (JOHNSON, 1994). Em suma, existe uma impossibilidade prática de se treinar exaustivamente a RNA nesses casos. Assim sendo, existe uma grande motivação na busca de novas técnicas e modelos conexionistas que possam ser aplicados nessa tarefa.

1.2 Das RNAs às RNPs

A proposição de alguns novos modelos conexionistas nos últimos anos vem provocando uma ruptura com os modelos prévios, a ponto de alguns autores cogitarem a divisão das RNAs segundo duas classes distintas: estáticas e pulsadas (MAASS; BISHOP, 1999).

redes
neurais
estáticas

Segundo esta ótica, as *redes neurais estáticas* constituem a grande maioria dos modelos matemáticos adotados para RNAs nos últimos 60 anos (perceptron,

rede de base radial, rede de Kohonen, etc.) e baseiam-se no princípio de utilizar variáveis contínuas para modelar processos que biologicamente ocorrem através de seqüências de pulsos temporais. Algumas das hipóteses para a falta de êxito pleno dessas redes em algumas tarefas – tais como o reconhecimento de objetos em imagens baseado em suas características de cor – são: *i*) os modelos de neurônios podem não capturar aspectos essenciais dos neurônios biológicos necessários para estas tarefas, e *ii*) o uso de variáveis contínuas pode dificultar a realização dessas tarefas.

redes
neurais
pulsadas

As Redes Neurais Pulsadas (RNPs) são redes que utilizam uma codificação temporal da informação. O recente amadurecimento dos modelos associados a essas RNPs tem estimulado o estudo dessa classe de RNAs no domínio do aprendizado de máquinas. Muito embora os limites de aplicação dessas redes ainda não estejam bem estabelecidos na literatura, estudos teóricos apontam que o poder computacional de RNPs pode ser superior ao dos modelos clássicos de RNAs (MAASS; BISHOP, 1999) e algumas RNPs parecem apresentar aptidão para tarefas relacionadas à VC (KINSER, 1997) (BOSH; LABBI; MILANESE, 1998) (DELORME; THORPE, 2001) (WANG, 2001) (KATADA; OHKUDA, 2003) (YU; ZHANG, 2004).

1.3 Por que RNPs?

Há consenso em que nem toda representação é computacionalmente equivalente (MARR, 1982). Um exemplo clássico que ilustra essa colocação é o fato de que é muito mais difícil executar operações aritméticas sobre números romanos do que executar as mesmas operações sobre números arábicos que utilizam a notação posicional (EDELMAN; DUVDEVANI-BAR, 1997).

Alguns pesquisadores vêem as redes neurais estáticas como redes que trabalham com a taxa média de disparos de neurônios (JOHNSON, 1994), um código biologicamente pouco plausível (RULLEN; THORPE, 2001). Em se tratando de RNPs, mais do que um modelo com maior *fidelidade biológica*, trata-se de uma ferramenta capaz de trabalhar com novos tipos de códigos.

Considerando a aplicação dessas redes na VC, alguns autores argumentam que RNPs podem apresentar alguma forma de aptidão para o trabalho no domínio da frequência, historicamente utilizado por algoritmos associados à VC. Investiga-se, então, a hipótese de que novas habilidades sejam possíveis utilizando uma *codificação temporal* dos estímulos visuais (WYSS; KÖNIG; VERSHURE, 2003).

Especificamente, indícios apontam no sentido de que a informação codificada

de forma temporal seria naturalmente invariante a transformações como a rotação, translação e escala (JOHNSON, 1994). Nesse sentido, RNPs podem ser uma abordagem mais propícia à tarefa do reconhecimento de objetos em imagens. Sistemas em *hardware* baseados em pulsos poderiam ainda apresentar *i)* baixo consumo e *ii)* processos transitórios rápidos (TORIKAI; SAITO, 2004).

Neste sentido, é grande o interesse pelo estudo dessas redes, já que vislumbra-se, em um futuro próximo, a possibilidade de aplicar modelos dessa natureza no reconhecimento automático de objetos em imagens utilizando um novo e promissor código neural.

1.4 Das RNPs às RNP-RBFs

Na tarefa de tornar neurônios artificiais capazes de processar sinais temporais, três questões são imediatas:

- i. Qual o código de pulsos mais adequado?;
- ii. Qual o modelo de neurônio mais adequado para processá-lo?; e
- iii. Qual a sistemática de aprendizado mais apropriada para estas redes?

Inúmeros códigos temporais (latência, fase, frequência, sincronia, etc.) e modelos de neurônios que operam sobre esses códigos (integra-e-dispara, ressoa-e-dispara, SRM, etc.) têm sido propostos nos últimos anos. Na grande maioria das redes que utilizam neurônios dessa natureza, contudo, o aprendizado não é apresentado explicitamente na forma de um algoritmo, mas confunde-se com o próprio modelo do neurônio ou da rede. Tipicamente, o processo de sincronização dos pulsos emerge da associação de uma coleção de neurônios (KINSER, 1997)(WANG, 2001)(KATADA; OHKUDA, 2003). Tornar explícita a sistemática de aprendizado de RNPs é importante para estabelecer ou expandir os seus domínios de aplicação.

Um modelo de neurônio pulsado que opera de forma similar a um neurônio com Função de Base Radial – *Radial Basis Function* (RBF) foi proposto por Hopfield (1995). A grande inovação deste modelo é a sua capacidade de armazenar informações nos tempos de atraso axonal dos neurônios. Um algoritmo de aprendizado para um desses neurônios foi proposto por Gerstner (1996). Alguns anos mais tarde um modelo similar em conjunto com uma sistemática de treinamento

não supervisionado foi proposta por Natschläger e Ruf (1998) para uma rede de neurônios pulsados.

O modelo de Natschläger e Ruf (1998) apresenta três problemas principais:

- i. A enorme dificuldade para configurar adequadamente seu grande número de parâmetros livres, tarefa muito mais complexa do que configurar uma RNA estática;
- ii. A dificuldade de convergência apresentada em algumas situações pelo processo de treinamento, sem razões claramente estabelecidas; e
- iii. A falta de uma sistemática para a codificação de dados reais para o domínio temporal, o que limita a aplicação deste modelo para dados reais.

Recentemente, Bohte (2003) propôs a utilização de campos receptivos gaussianos para codificar os dados de entrada. Esta proposição tornou a rede de Natschläger e Ruf (1998) capaz de tratar dados reais, muito embora as implicações de tal codificação não tenham sido profundamente avaliadas pelo autor. Em uma análise mais ampla, o trabalho de Bohte mostrou que a codificação dos dados em pulsos temporais e as RNP-RBFs treinadas de forma não supervisionada constituem um paradigma viável para a classificação de padrões no âmbito computacional.

O estudo da RNP-RBF justifica-se, então, por dois motivos principais: *i)* a rede faz uso da codificação temporal, uma nova forma de representar a informação; e *ii)* a rede encontra-se apta a acolher algoritmos explícitos de aprendizado. Contudo, a dificuldade prática para lidar com o grande número de parâmetros da rede ainda persiste como um entrave ao aprofundamento do estudo desse modelo, agora agravado pela inserção dos campos gaussianos. A reprodução de experimentos realizados em trabalhos prévios tem se mostrado impraticável dada a deficiência na documentação desses experimentos (BOOIJ, 2004), e a literatura não apresenta metodologias para calibração nem ao menos para os parâmetros que são críticos para o processo de convergência da rede.

1.5 Objetivos e contribuições

objetivo O presente trabalho tem por objetivo reduzir o número de parâmetros da rede neural pulsada de base radial com treinamento não supervisionado e codificação via campos receptivos gaussianos, fornecendo metodologias claras para a

calibração de seus parâmetros-chave. Busca-se, em suma, simplificar o uso desta rede.

Além da contribuição principal no processo de configuração da RNP, vislumbra-se como contribuições relevantes:

- i. A proposição de uma função para o aprendizado não supervisionado da rede capaz de permitir uma calibração mais simples de alguns dos seus parâmetros-chave;
- ii. A avaliação das implicações do uso da codificação via campos receptivos gaussianos, ainda pouco estudados;
- iii. A demonstração e documentação da aplicação dessa rede em algumas tarefas (agrupamento e da classificação da cor de pixels em imagens coloridas), de forma a permitir a reprodução dos experimentos realizados;
- iv. O posicionamento deste modelo no contexto das técnicas clássicas para estas tarefas, de forma a facilitar futuras analogias e extensões deste modelo;
- v. A tentativa de organizar diversas teorias e metodologias associadas às RNPs, ainda não sedimentadas na literatura; e
- vi. A maior investigação do potencial de utilização do modelo de neurônio denominado Modelo de Resposta Pulsada – *Spiking Response Model* (SRM) – nas redes neurais pulsadas de base radial, já que esta rede em sua concepção original empregava neurônios do modelo *integra-e-dispara* (*integrate-and-fire*) (NATSCHLÄGER; RUF, 1998).

1.6 Organização do trabalho

O presente trabalho encontra-se dividido em três partes. Na parte I (que compreende os capítulos 2–4), são apresentados os **conceitos fundamentais** para a compreensão do trabalho. O capítulo 2 apresenta uma visão geral das redes neurais pulsadas, com particular ênfase no modelo RNP-SRM-RBF. Os algoritmos e processos de aprendizado para este modelo são discutidos no capítulo 3. O capítulo 4 procura apresentar os algoritmos clássicos do aprendizado baseado em instâncias que mantêm relações com o modelo de RNP adotado.

Na parte II (que compreende os capítulos 5–6), o **aprendizado proposto** é apresentado. O capítulo 5 busca reavaliar a RNP segundo os conceitos ge-

rais do aprendizado apresentados no capítulo anterior. O capítulo 6 apresenta a sistemática de aprendizado idealizada para a RNP.

Em sua parte III (que compreende os capítulos 7–10), são apresentados os **resultados experimentais** obtidos com o modelo proposto. Os materiais e métodos empregados são detalhados no capítulo 7. O capítulo 8 apresenta estudos de caso para a tarefa do agrupamento tridimensional. O capítulo 9 apresenta estudos de caso para a tarefa da classificação de pixels. Finalmente, o capítulo 10 conclui com a discussão dos resultados obtidos e propõe direções para futuras investigações.

Parte I

Conceitos Fundamentais

2 Redes Neurais Pulsadas

“(...) existem muitas complexidades em sistemas neurais biológicos que não são modeladas por RNAs, e muitas características das RNAs que discutimos aqui são sabidamente inconsistentes com sistemas biológicos. Por exemplo, consideramos aqui RNAs cujas saídas das unidades individuais são valores constantes, enquanto as saídas de neurônios biológicos são complexas séries de pulsos temporais” (MITCHELL, 1997, p. 82).

Um dos objetivos da IA ao longo do anos tem sido simular em computadores o funcionamento do cérebro humano, o único modelo de máquina inteligente que ocorre naturalmente (RICH; KNIGHT, 1993). O cenário apresentado por Mitchell (1997), porém, retrata com exatidão a lacuna que existe hoje entre a grande maioria dos modelos de neurônios artificiais e seus análogos biológicos.

Na tarefa de propor modelos computacionais conexionistas mais fiéis ao cérebro humano, algumas questões são de primordial importância: *i)* Quais os modelos de neurônios para processamento baseado em pulsos?; *ii)* Como associar essas unidades neurais de processamento?; e *iii)* Como representar a informação através de pulsos temporais?

Um novo paradigma conexionista preocupado com essas questões tem se delineado nos últimos anos, graças principalmente às contribuições recebidas da neurociência. Esta nova teoria computacional parece não realizar o processamento temporal da mesma forma que os modelos neurais clássicos, justificando sua classificação como uma nova linha de modelos computacionais. Contudo, dado o recente despontar desta teoria, a literatura na área ainda é escassa e controversa, particularmente no que diz respeito à taxonomia para essas redes e seu posicionamento no contexto das RNAs consolidadas.

Visando apresentar um panorama geral das RNAs incluindo os novos modelos pulsados, alguns dos aspectos considerados de maior relevância serão descritos nas seções a seguir.

Na seção 2.1, os principais códigos adotados em RNPs serão brevemente descritos. A seguir, a seção 2.2 apresenta uma revisão dos principais eventos cronológicos da teoria das redes neurais artificiais, apresentando os eventos e códigos relevantes para a teoria das RNPs. Esta revisão foi realizada inspirada na ótica de Maass (1997). Os modelos de neurônios pulsados são abordados na seção 2.3. Finalmente, a seção 2.4 descreve redes desses neurônios.

2.1 Códigos

As discussões sobre o código neural – entenda-se a forma como uma RNP representa a informação através dos neurônios – são vastas na literatura. Os impulsos elétricos gerados na saída dos neurônios biológicos (referenciados neste texto simplesmente como pulsos) podem representar informação em uma ampla gama de formas distintas, cada uma das quais com suas especificidades e aplicabilidades. Um consenso sobre essa questão parece ainda estar distante.

De uma forma geral, a presença ou ausência de pulsos parece ter forte relevância para neurônios biológicos, enquanto sua forma e tamanho não (MAASS, 1996). Com base neste princípio, RNPs têm feito uso de diferentes códigos – via de regra de natureza binária – para representar informações. Algumas das codificações mais freqüentes são descritas a seguir. Para uma melhor compreensão dos códigos aqui apresentados, alguns deles encontram-se visualmente representados na figura 2.1.

2.1.1 Taxa média de disparos

A taxa média de disparos não constitui propriamente uma representação temporal da informação, mas tem sido a grandeza mais utilizada para a codificação das informações transmitidas por neurônios nos últimos anos, e por este motivo é abordada aqui.

Segundo Maas, *“Uma rápida olhada na literatura experimental revela que não há um conceito único e bem definido de taxa média de disparos.”*(MAASS; BISHOP, 1999, p. 7). Ao menos três noções diferentes de taxas são usualmente utilizadas nos trabalhos com neurônios: *i)* média sobre o tempo; *ii)* média sobre repetições; e *iii)* média sobre a população de neurônios.

Média sobre o tempo. Seja um neurônio que dispara diversas vezes. A partir da definição de uma janela de tempo Δt , a média da taxa de disparo sobre

o tempo é dada pela equação 2.1:

$$\bar{p}_t = \frac{n_p}{\Delta t}, \quad (2.1)$$

onde: $\bar{p}_t$ é a taxa média de disparo sobre o tempo; n_p é o número de pulsos que ocorreram no período Δt ; Δt é o intervalo de tempo.

Uma representação dessa situação é mostrada na figura 2.1a.

Média sobre repetições. Seja um neurônio estimulado gerando pulsos em sua saída. Se este estímulo for repetido por diversas vezes, um observador pode, para cada intervalo de tempo Δt , contar o número de disparos ocorridos sobre todas as repetições do experimento. A soma desses disparos representa um histograma de disparos. Uma média sobre as repetições pode ser dada pela equação 2.2:

$$\bar{p}_r = \frac{1}{\Delta t} \cdot \frac{n_p(t; t + \Delta t)}{n_r}, \quad (2.2)$$

onde: $\bar{p}_r$ é a taxa média de disparo sobre as repetições; Δt é o intervalo de tempo; $n_p(t; t + \Delta t)$ é o número de pulsos que ocorreram no período de t até $t + \Delta t$; n_r é o número de repetições que foram executadas.

Uma representação visual desta situação é apresentada na figura 2.1b.

Média sobre populações de neurônios. Sejam duas populações de neurônios i e j com um número muito grande de neurônios em cada uma. Suponha que: *i*) todos os neurônios em uma população têm o mesmo padrão de conexões de entrada e saída; *ii*) cada neurônio da população j recebe entrada de todos os neurônios da população i ; e *iii*) pulsos da população i são enviados para a população j . Do ponto de vista dos neurônios receptores, a quantidade média de informação recebida pode ser dada por:

$$\bar{p}_{pn} = \frac{1}{\Delta t} \cdot \frac{n_p(t; t + \Delta t)}{n_{pop}}, \quad (2.3)$$

onde: $\bar{p}_{pn}$ é a taxa média de disparo sobre a população de neurônios; Δt é o intervalo de tempo; $n_p(t; t + \Delta t)$ é o número de pulsos que ocorreram no período de t até $t + \Delta t$; n_{pop} é o tamanho da população.

2.1.2 Latência (ou atraso)

Trata-se do tempo, a partir de algum evento, para a chegada dos pulsos a alguma das sinapses do neurônio (ver figura 2.1c). Este código tem particular relevância quando estímulos externos são alterados abruptamente. Na visão humana, por exemplo, os olhos freqüentemente desviam de um ponto para o outro através dos movimentos conhecidos como sacádicos (para uma visão geral ver Gregory (1990)). Após um movimento sacádico, uma nova informação visual estará disponível e pode-se imaginar que o código dos tempos de chegada dos pulsos provenientes da retina contenha informações importantes sobre a nova imagem. Particularmente o tempo para a chegada do primeiro pulso tem sido considerado de grande relevância neste contexto. Alguns modelos de RNP como os de Natschläger e Ruf (1998) e Bohte (2003) utilizam esta codificação.

2.1.3 Fase

A idéia da codificação pelo tempo para o primeiro pulso pode ser utilizada para produzir um novo modelo de codificação se o sinal de referência for periódico ao invés de um evento isolado (figura 2.1d). Existem evidências biológicas da utilização desse código para a localização espacial e no sistema olfativo de mamíferos.

2.1.4 Sincronia

Os pulsos provenientes de outros neurônios também podem ser utilizados como sinal de referência para um código de pulsos. O sincronismo (também chamado correlação) entre um par ou um grupo de pulsos de neurônios pode conter informações que certamente não estão contidas na taxa média de disparo de neurônios. Uma representação desta situação é apresentada na figura 2.1e.

2.2 Histórico de RNA

Em 1894, o espanhol Ramòn y Cajal identificava anatomicamente uma célula nervosa em sua publicação “A retina dos vertebrados” (CAJAL, 1894). Muitos anos mais tarde, o célebre artigo “Um cálculo lógico das idéias provenientes da atividade nervosa” de McCulloch e Pitts (1943) inaugurou a teoria das “redes neurais artificiais”, propondo como unidade básica de processamento um discriminador linear.

Códigos: a) média sobre o tempo; b) média sobre repetições; c) latência: os pulsos possuem tempos de atraso de emissão (ou latências) distintos; d) fase: pulsos entrando em fase com um estímulo externo; e) sincronia: pulsos entrando em fase uns com os outros. Extraído de (MAASS; BISHOP, 1999).

Figura 2.1: Representação de códigos utilizados em redes neurais pulsadas.

No final da década de 50, Roseblatt desenvolve uma rede com múltiplos neurônios denominados *perceptron* (ROSENBLATT, 1962). Este modelo também é freqüentemente referenciado como modelo de portas de limiar (*threshold-gates model*) e estabelece uma primeira geração de RNAs, cujos modelos possuem como característica marcante as saídas booleanas.

As contribuições de Widrow (1962) com a *regra delta* e Rumelhart, Hinton e Williams (1986) com o algoritmo da retropropagação do erro levaram ao desenvolvimento de uma segunda geração de RNAs. Suas unidades básicas de processamento eram neurônios com função de ativação diferenciável, suportando assim – entre outros – algoritmos de aprendizado baseados em gradiente. Estas unidades computavam funções com entradas e saídas analógicas e eram capazes de aproximar qualquer função arbitrariamente bem.

Os trabalhos de Kohonen (1972) desenvolvendo o Mapas Auto-Organizados

– *Self-Organized Maps* (SOM), as memórias associativas de Hopfield (1982) e as redes com RBF de Park e Sandberg (1991) são outros pontos altos dessa geração.

Alguns autores como Maass e Bishop (1999) vêem as redes da segunda geração como aquelas que computam utilizando um código que pode ser entendido como a taxa média de disparos de neurônios biológicos em um intervalo de tempo. Considerando desta forma, os modelos de neurônios que processam seqüências de pulsos têm sido referenciados como uma *terceira geração* de RNAs (MAASS, 1997). É preciso observar, contudo, que não existe uma ordem cronológica bem estabelecida entre os modelos das chamadas segunda e terceira geração. Assim sendo, procura-se evitar aqui esta taxonomia, e os modelos pertencentes a esta classe de rede serão tratados adiante simplesmente como RNPs.

Segundo Kistler, Gerstner e Hemmen (1997), os modelos mais simples da dinâmica de pulsos neurais – particularmente o modelo de neurônio *integra-e-dispara* – datam de trabalhos prévios aos de Lapicque (1907). Contudo, a primeira descrição matemática formal do processo dinâmico da atividade neural é atribuída a Hodgkin e Huxley (1952).

O modelo de Hodgkin-Huxley descreve a atividade neural biológica em um nível de abstração microscópico, modelando o comportamento dos canais iônicos dos neurônios, e, portanto, sua dinâmica eletroquímica através de um conjunto de complexas equações não lineares. Nasceram os modelos baseados em condutância e compartilhamento.

A excessiva complexidade desse equacionamento e sua proibitiva complexidade computacional estimularam a proposição de modelos com um nível de abstração mais elevado. Estabeleceram-se então os modelos baseados em limiar de disparo, onde se considera o neurônio como uma entidade homogênea capaz de gerar pulsos se a excitação em suas sinapses for suficientemente elevada.

Os neurônios do tipo *integra-e-dispara* são considerados por alguns como o modelo mais simples desta classe. Gerstner (1991) propôs um modelo denominado SRM, com posteriores contribuições de Kistler, Gerstner e Hemmen (1997), que o consideram uma generalização do modelo *integra-e-dispara*. Mais do que isto, alguns autores classificam o SRM como um modelo genérico de neurônio baseado em limiar de disparo (MAASS; BISHOP, 1999).

Um dos primeiros processos computacionais para tratar com disparos neurais foi realizar a sincronização da atividade dos neurônios, proposta que é atribuída a Milner (1974) e Malsburg e Schneider (1986). Este princípio levou ao surgimento

de uma larga gama de modelos neurais cuja taxonomia é controversa na literatura. Adota-se aqui como princípio para a classificação desses modelos a observação de Maass e Bishop (1999, p. 114) de que “*em alguns casos a sincronização coincide com a oscilação, mas a sincronização não depende da presença de oscilação*”. Nesse sentido, podemos notar ao menos dois tipos de modelos de neurônios: os modelos que permitem sincronização (tais como o modelo genérico do SRM) e os osciladores.

Um dos primeiros modelos com sincronização foi o de Eckhorn (1990), que propôs um gerador de pulsos chamado *neuromine* associado em uma rede de campos conectados (*linking field network*) que exibia descargas de pulsos. A coleção de modelos de RNAs derivados deste princípio tornou-se conhecida na literatura com o nome de Rede Neural de Pulso Acoplado – *Pulse-Coupled Neural Network* (PCNN).

A idéia básica presente nas PCNNs é a de que representações distribuídas no espaço podem ser agrupadas ou dissociadas alterando a sincronia do disparo de um neurônio em relação aos demais. Para uma visão geral das PCNNs e suas aplicações veja Johnson (1994) e Johnson e Padgett (1999).

Outras abordagens recentes seguem na direção da Rede Caótica de Pulso Acoplado – *Chaotic Pulse-Coupled Network* (CPCN), que utiliza como unidade básica de processamento o Oscilador Pulsado Caótico – *Chaotic Spiking Oscillator* (CSO) (NAKANO; SAITO, 2004). Um outro modelo que tem se destacado na tarefa da sincronização das descargas neurais é a Rede Osciladora Localmente Excitatória, Globalmente inibitória – *Locally Excitatory, Globally Inhibitory Oscillatory Network* (LEGION) proposta por Wang e Terman (1995). Um outro representante desta classe é o *ressoa-e-dispara* (IZHIKEVICH, 2001).

Alguns autores como Campbell (1997) classificam os neurônios do tipo integra-e-dispara e as PCNNs como tipos elementares de osciladores. De qualquer forma, segundo Yu e Zhang (2004), as PCNNs e a maioria dos modelos compostos por osciladores compartilham a mesma característica comum: eles representam informação via oscilação de fase ou tempo de pulsos, e os neurônios em diferentes localizações são relacionados pela sincronização de suas atividades.

Uma das principais aplicações desses métodos baseados em osciladores é a segmentação de imagens utilizando características simples, por exemplo, a proximidade espacial e similaridade de brilho dos pixels. Em geral também não se fazem necessários algoritmos explícitos de treinamento: a similaridade emerge do comportamento de seus neurônios.

Uma outra linha de trabalho com RNPs consiste em utilizar um modelo de neurônio pulsado (tipicamente o integra-e-dispara ou o SRM) na forma de um neurônio RBF (HOPFIELD, 1995). O código utilizado nesse tipo de rede tipicamente não é a sincronia ou a fase, mas o tempo de chegada dos pulsos (também denominado *latência*).

Um algoritmo de aprendizado para um desses neurônios foi proposto por Gerstner (1996). Alguns anos mais tarde, o trabalho de Natschläger e Ruf (1998) resultou em um novo modelo similar e em uma sistemática de treinamento não supervisionado para uma rede desses neurônios. Este modelo, no entanto, possui uma grande limitação: a impossibilidade de tratar dados reais, dado que não foi definida uma sistemática para transformação de dados em pulsos temporais.

Recentemente, Bohte (2003) propôs a utilização de campos receptivos gaussianos para codificar os dados de entrada, o que tornou estas redes aptas a tratar dados reais. Além desta contribuição, Bohte também propôs uma extensão do algoritmo da retropropagação do erro para esta rede, desenvolvendo uma metodologia de treinamento supervisionado, gerando uma nova linha de investigações (MOORE, 2002) (BOOIJ, 2004).

Desta forma, além de permitir um novo tipo de classificação baseada nos tempos de chegada dos pulsos em suas entradas, a rede de Natschläger-Ruf-Bohte figura como uma das primeiras RNPs com algoritmos explícitos de treinamento. O panorama apresentado nesta seção configura, então, um momento histórico em que as RNPs – um paradigma comprovadamente viável para classificação – encontram-se particularmente aptas a acolher novas metodologias de treinamento ou mesmo adaptações ou extensões de algoritmos utilizados em redes estáticas.

2.3 Modelos de neurônios

Após a breve revisão do contexto histórico das RNPs, duas observações podem ser realizadas: *i*) os limites de aplicação de cada modelo não se encontram estabelecidos; e *ii*) a classificação dos modelos de neurônios pulsados não é uma questão fechada.

Mais do que um problema de taxonomia, alguns trabalhos sugerem que muitos desses modelos podem ser equivalentes. Maass e Bishop (1999, p. 47), por exemplo, sugerem que o modelo de Hodgkin-Huxley pode ser aproximado pelo SRM através da escolha de um conjunto adequado de núcleos (*kernels*) de resposta. O trabalho de Izhikevich (1999) sugere que qualquer rede de Hodgkin-Huxley pode

ser transformada em uma PCNN através de uma mudança de variáveis. Sugere-se ainda que modelos baseados em equações diferenciais possam vir a ser uma simplificação do modelo de Hodgkin-Huxley (MAASS; BISHOP, 1999). De forma análoga, outros trabalhos procuram estabelecer relações entre esses modelos.

Em uma tentativa de diferenciar as classes de modelos apresentadas, a figura 2.2 apresenta uma proposta de classificação desses modelos segundo sua forma mais comum de uso. As seções a seguir apresentam com maior profundidade três desses modelos: o modelo de Hodgkin-Huxley (primeira descrição matemática formal do processo dinâmico da atividade neural), o modelo integra-e-dispara (modelo baseado em oscilador mais diretamente relacionado com o SRM) e o SRM (modelo genérico de neurônio baseado em limiar de disparo e adotado neste trabalho).

Figura 2.2: Alguns dos principais modelos de neurônios utilizados em RNPs e sua classificação segundo sua forma mais comum de uso.

2.3.1 Hodgkin-Huxley

Os modelos baseados em condutância foram originados pelo trabalho de estudo dos neurônios de lulas gigantes por Hodgkin e Huxley (1952), e preocupam-se com a transmissão do potencial de ação pelo axônio da célula nervosa, basicamente controladas por um conjunto de equações diferenciais. Este modelo será descrito através de uma analogia com o circuito elétrico apresentado na figura 2.3.

Em uma célula nervosa, uma membrana semipermeável separa o interior da célula do líquido extracelular. Devido à permeabilidade seletiva da membrana e devido ao transporte de íons através da mesma, a concentração dos íons no meio

Figura 2.3: Representação esquemática do modelo do neurônio de Hodgkin-Huxley.

intra e extracelular é diferente. Na figura 2.3, a membrana celular é representada pelas constantes R e C , e os resistores Na e K modelam respectivamente os canais iônicos de sódio e potássio. Uma corrente $I(t)$ injetada no circuito flui através dos canais da membrana celular e simultaneamente carrega o capacitor C . Quando o potencial da membrana aumenta significativamente, os canais iônicos abrem-se. Primeiramente o canal Na (gerando um pulso positivo) e depois o canal K (gerando um pulso negativo denominado refração).

$$C \frac{d\mu}{dt} = - \sum_k I_k + I(t), \quad (2.4)$$

onde: C é a capacitância da membrana; $\mu(t)$ é a tensão sobre o resistor e o capacitor; $\sum_k I_k$ é a somatória de correntes iônicas através da membrana da célula; $I(t)$ é a corrente externa injetada no circuito.

Em cada modelo baseado em condutância e compartilhamento, diferentes quantidades de canais iônicos são modelados, promovendo uma descrição mais rica de características como a dinâmica do neurônio e a transmissão sináptica. No caso do modelo de Hodgkin-Huxley, três tipos de correntes iônicas são indexadas: Na , K e L . Desta forma:

$$\sum_k I_k = g_{Na} \cdot m^3 \cdot h (\mu - V_{Na}) + g_K \cdot n^4 (\mu - V_K) + g_L (\mu - V_L), \quad (2.5)$$

onde: g_{Na}, g_K, g_L são as condutâncias máximas dos canais; a ação combinada de m e h controla o canal de Na ; n controla o canal de K ; μ é o potencial da membrana; V_{Na}, V_K, V_L são constantes chamadas potenciais reversos, já que a corrente I_k muda quando o potencial μ supera V_k .

Finalmente, as variáveis m , n e h evoluem segundo as equações diferenciais:

$$\dot{m} = \alpha_m(\mu)(1 - m) - \beta_m(\mu).m \quad (2.6)$$

$$\dot{n} = \alpha_n(\mu)(1 - n) - \beta_n(\mu).n \quad (2.7)$$

$$\dot{h} = \alpha_h(\mu)(1 - h) - \beta_h(\mu).h \quad (2.8)$$

onde: $\dot{m}, \dot{n}, \dot{h}$ são respectivamente as derivadas de m , n e h com relação ao tempo; α, β são funções empíricas de μ que são ajustadas para os dados do axônio da lula gigante.

O conjunto de equações diferenciais 2.4 a 2.8 definem este modelo.

2.3.2 Integra-e-dispara

Um modelo baseado em limiar de disparo e baseado no princípio da oscilação é o integra-e-dispara. Este modelo também é usualmente descrito em termos de equações diferenciais. O circuito equivalente de um neurônio integra-e-dispara é o mostrado na figura 2.4. Um capacitor C encontra-se em paralelo com uma resistência R e drenando uma corrente $I(t)$. A corrente divide-se entre o resistor e a carga do capacitor. Matematicamente, teremos:

$$I(t) = \frac{\mu(t)}{R} + C \cdot \frac{d\mu}{dt}, \quad (2.9)$$

onde: $I(t)$ é a corrente no circuito; $\mu(t)$ é a tensão sobre o resistor e o capacitor; R é o resistor; C é o capacitor.

Figura 2.4: Representação esquemática do modelo do neurônio integra-e-dispara.

Multiplicando-se todos os termos por R chegamos à expressão:

$$\tau_m \cdot \frac{d\mu}{dt} = -\mu(t) + R \cdot I(t), \quad (2.10)$$

onde: τ_m é a constante de tempo RC; $I(t)$ é a corrente no circuito; $\mu(t)$ é a tensão sobre o resistor e o capacitor; R é o resistor; C é o capacitor.

A equação 2.10 é usualmente suplementada por uma condição de limiar. A condição de limiar $\mu(t_i) = \vartheta$ define os instantes de tempo de disparo do neurônio i , dado por t_i . A forma do disparo não é descrita explicitamente, e após o disparo o potencial do neurônio retorna a um valor μ_0 , e a dinâmica é novamente descrita pela equação 2.10.

O modelo integra-e-dispara tem sido um dos mais empregados nos últimos anos para o estudo das RNPs, sobretudo devido a sua extrema simplicidade. Pela mesma razão também é o modelo mais implementado em *hardware*. Dentre suas aplicações usuais estão a segmentação de imagens (CHOE; MIIKKULAINEN, 1998) (BOSH; LABBI; MILANESE, 1998), o agrupamento de dados (NATSCHLÄGER; RUF, 1998) e o reconhecimento de faces (DELORME; THORPE, 2001). Outros estudos de natureza teórica podem ser encontrados em (AMEMORI; ISHII, 1998) (ABE, 2002) (MURESAN, 2002). Alguns modelos como Choe e Miikkulainen (1998) podem ser considerados derivados do integra-e-dispara.

2.3.3 SRM

SRM O SRM (*spiking response model*) é um modelo de neurônio pulsado baseado em limiar de disparo mais genérico que o integra-e-dispara (GERSTNER, 1995) (MAASS; BISHOP, 1999). “*Em contrapartida ao modelo integra-e-dispara, que é usualmente representado com equações diferenciais, o spiking response model é baseado em núcleos de resposta que descrevem o efeito integrado da recepção ou emissão de pulsos no potencial da membrana*” (GERSTNER, 1995, p 739). A modelagem da refração também é um ponto importante neste modelo. A figura 2.5 o apresenta esquematicamente.

O estado do neurônio j é descrito pela variável de estado μ_j , análogo do potencial elétrico da membrana do neurônio biológico, e será tratado no decorrer do texto simplesmente como *potencial do neurônio*.

O neurônio biológico é dito disparar quando o potencial da membrana atinge o *potencial de ação*, instante onde ocorre uma rápida reversão na situação usual de ação

Figura 2.5: Diagrama geral do modelo do neurônio pulsado.

dos meios intra e extracelular do neurônio, provocando o que é denominado pulso, impulso nervoso ou descarga (BEAR; CONNORS; PARADISO, 2006). No modelo computacional, o neurônio é dito disparar se μ_j atinge o limiar ϑ_j . O conjunto dos tempos de disparo de um neurônio é denotado:

$$T_j = \{t_j^{(f)}; 1 \leq f \leq n_d\} = \{t \mid \mu_j(t) = \vartheta_j\}, \quad (2.11)$$

onde: T_j é o conjunto dos tempos de disparo do neurônio j ; $t_j^{(f)}$ é o momento do f -ésimo disparo do neurônio j ; f é o índice do disparo do neurônio; n_d é o número máximo de disparos; t é a variável tempo; $\mu_j(t)$ é o potencial do neurônio j no tempo t ; ϑ_j é o limiar do neurônio j .

A variável de estado μ_j tem seu valor alterado por dois processos distintos: *i*) estímulos de neurônios pré-sinápticos e *ii*) refração.

O primeiro processo capaz de alterar o valor da variável de estado μ_j é o estímulo recebido de neurônios pré-sinápticos. Seja o conjunto dos neurônios pré-sinápticos ao neurônio j dado por $\Gamma_j = \{i \mid i \text{ pré-sináptico a } j\}$. Um pulso pré-sináptico recebido do neurônio i no tempo $t_i^{(f)}$ incrementa (pulso excitatório) ou decrementa (pulso inibitório) o valor do estado μ_j do neurônio j segundo a expressão (MAASS; BISHOP, 1999):

$$\rho_{ij}(t) = w_{ij} \cdot \epsilon_{ij}(t - t_i^{(f)}), \quad (2.12)$$

onde: $\rho_{ij}(t)$ é o valor do estímulo recebido pelo neurônio j do neurônio i no tempo t ; w_{ij} é o peso sináptico associado à i -ésima sinapse do neurônio j ; $\epsilon_{ij}(s)$ é uma

função que descreve o comportamento do neurônio j quando do recebimento de um pulso através da i -ésima sinapse; t é a variável tempo, com $t > t_i^{(f)}$; $t_i^{(f)}$ é o momento do f -ésimo disparo da sinapse i .

Uma função usualmente adotada para modelar o comportamento de neurônios com relação aos pulsos pré-sinápticos é (MAASS; BISHOP, 1999):

$$\epsilon_{ij}(s) = \left[\exp\left(-\frac{s - \delta_{ij}}{\tau_m}\right) - \exp\left(-\frac{s - \delta_{ij}}{\tau_s}\right) \right] \mathcal{H}(s - \delta_{ij}), \quad (2.13)$$

onde: $\epsilon_{ij}(s)$ é a função que modela o comportamento do neurônio j com relação aos pulsos recebidos pela i -ésima sinapse; δ_{ij} é o atraso na propagação axonal da conexão ij ; τ_m, τ_s são respectivamente as constantes de descida e subida da função, com $0 < \tau_s < \tau_m$; $\mathcal{H}(s)$ é a função degrau de *Heaviside*¹. A figura 2.6 apresenta um gráfico desta função.

Figura 2.6: Curva típica de ϵ_{ij} .

A figura 2.7 apresenta esquematicamente a influência de disparos de neurônios pré-sinápticos na variável de estado de um neurônio, segundo o modelo apresentado nas equações 2.12 e 2.13. Na figura, os pulsos pré-sinápticos não foram suficientes para que o neurônio atinja seu limiar de disparo ϑ_j .

refração

O segundo processo capaz de alterar a variável de estado $u_j(t)$ é a refração. Tipicamente, depois que um pulso de saída é gerado no neurônio j em $t_j^{(f)}$, a variável μ_j é levada a zero. Matematicamente, a variável de estado μ_j recebe uma

¹A função degrau de *Heaviside* $\mathcal{H}(s)$ – homenagem ao engenheiro eletricitista inglês Olivier Heaviside (1850-1925) – não é definida para $s < 0$ e possui valor $\frac{1}{2}$ para $s = 0$ e 1 para $s > 0$.

Figura 2.7: Influência de disparos dos neurônios pré-sinápticos i_1 e i_2 na variável de estado $\mu_j(t)$ de um neurônio j .

contribuição negativa, dada usualmente por:

$$\gamma_j(s) = -\vartheta_j \exp\left(-\frac{s}{\tau_r}\right) \cdot \mathcal{H}(s), \quad (2.14)$$

onde: $\gamma_j(s)$ é a função que modela a refração no neurônio j ; ϑ_j é o limiar do neurônio j ; τ_r é uma constante de tempo; $\mathcal{H}(s)$ é a função degrau de *Heaviside*.

A figura 2.8 apresenta esquematicamente a refração de um neurônio após seu disparo. Observe-se que a função de refração $\gamma_j(s)$ modelada segundo a definição da equação 2.14 não é definida para $s < 0$ e decai a zero para $s \rightarrow \infty$.

Figura 2.8: A refração γ_j do neurônio j após atingir o limiar de disparo ϑ_j .

Uma vez modeladas a refração e a influência dos neurônios pré-sinápticos, finalmente pode-se apresentar a equação completa que modela o estado $u_j(t)$ de um neurônio j no tempo t , dada pela superposição linear das equações 2.12 e

2.14.

$$u_j(t) = \underbrace{\sum_{t_j^{(f)} \in T_i} \gamma_j(t - t_j^{(f)})}_{(1)} + \underbrace{\sum_{i \in \Gamma_j} \sum_{t_i^{(f)}} w_{ij} \cdot \epsilon_{ij}(t - t_i^{(f)})}_{(2)}, \quad (2.15)$$

onde: (1) modela a refração do neurônio j para todo disparo de j ; (2) modela o conjunto de todos os estímulos recebidos pelo neurônio j de todos os neurônios pré-sinápticos a ele.

Entre as aplicações do SRM estão o agrupamento de dados (*clustering*) (BOHTE; KOK; la POUTRÉ, 2000) (BOHTE; KOK; la POUTRÉ, 2002a) (BOHTE; KOK; la POUTRÉ, 2002b) (BOHTE; la POUTRÉ; KOK, 2002) (BOHTE, 2003), a classificação de padrões de locomoção (KATADA; OHKUDA, 2003) e o reconhecimento de locutores (TIMOSCZCZUK, 2004).

modelo
do NP

O modelo apresentado nesta seção será denotado no decorrer deste trabalho como modelo do Neurônio Pulsado – NP. Para definições adicionais veja (MAASS, 1996) e (MAASS; BISHOP, 1999).

2.3.4 Comparativo com outros modelos

Nesta seção busca-se apresentar de forma introdutória a capacidade de computação do SRM realizando um comparativo com modelos clássicos de redes neurais.

Comparativo com McCulloch-Pitts. O neurônio de McCulloch-Pitts, apresentado esquematicamente na figura 2.9, pode ser definido da seguinte maneira:

$$y = \begin{cases} 1, & \text{se } \sum_{k=1}^n w_k \cdot x_k \geq \vartheta; \\ 0, & \text{caso contrário.} \end{cases} \quad (2.16)$$

onde: y é a saída do neurônio; n é o número de entradas do neurônio; w_k é o peso sináptico associado à k -ésima entrada do neurônio; x_k é o valor presente na k -ésima entrada do neurônio; ϑ é o limiar de disparo do neurônio.

Considere a figura 2.7, onde as funções pré-sinápticas ϵ_{ij} são todas idênticas – considere, por exemplo, $\epsilon_{ij} = \max \epsilon_{ij}(s)$ (no caso de pulsos excitatórios) ou $\epsilon_{ij} = \min \epsilon_{ij}(s)$ (no caso de pulsos inibitórios) – e onde os neurônios pré-sinápticos

Figura 2.9: Diagrama esquemático do modelo do neurônio de McCulloch-Pitts.

disparam (se for o caso) todos simultaneamente. Neste caso, o neurônio dispara se e somente se:

$$\sum_j w_{ij} \cdot \epsilon_{ij} \geq \vartheta_j. \quad (2.17)$$

A equação 2.17 comparada com a equação 2.16 demonstra que o modelo SRM tem o poder computacional equivalente ao neurônio de McCulloch-Pitts. De forma análoga, é possível mostrar que uma RNP pode simular uma rede composta por neurônios de McCulloch-Pitts, desde que exista um mecanismo de sincronização entre os neurônios (MAASS; BISHOP, 1999).

Comparativo com portas sigmoidais. Considere que números naturais são codificados através do tempo de chegada de pulsos, e que se deseja uma RNA capaz de identificar quando dois ou mais desses números têm valores similares. Em outros termos, deseja-se uma rede capaz de identificar quando dois pulsos forem recebidos quase simultaneamente pela RNA. A figura 2.10 apresenta a variável de estado $\mu_j(t)$ de um único neurônio j com n_p predecessores imediatos $\Gamma_j = \{1, \dots, n_p\}$ modelado para realizar esta tarefa.

Em outros termos, para quaisquer valores das constantes de tempo τ_s, τ_m de $\epsilon_{ij}(s)$ é possível determinar valores $0 < c_1 < c_2$ e ϑ_j de forma que $\mu_j(t) < \vartheta_j$ para qualquer entrada com um número arbitrário de pulsos pré-sinápticos excitatórios com distância $> c_2$, enquanto que $\mu_j(t) > \vartheta_j$ para excitações com distância $\leq c_1$ (MAASS; BISHOP, 1999).

O fato desta tarefa ser realizada com relativa facilidade por um único neurônio SRM permite efetuar um comparativo entre esse modelo e portas sigmoidais da seguinte maneira: pode-se mostrar que, para computar a mesma tarefa, uma RNA com portas sigmoidais arbitrárias precisa ter ao menos $\frac{n-4}{2}$ portas, onde n é o número de entradas (a prova desta afirmação encontra-se em Maass (1997)). Estes resultados corroboram com a noção intuitiva de que esta é uma tarefa difícil

a) As distâncias entre os disparos de entrada são grandes e o neurônio não dispara; b) A distância c entre dois dos disparos é pequena ocasionando o disparo do neurônio.

Figura 2.10: Diagrama esquemático da variável de estado $u_j(t)$ do neurônio j realizando a comparação de três valores.

de ser realizada por uma rede neural tradicional.

Comparativo com RBFs. Um modelo de neurônio pulsado implementando uma função de base radial (RBF - *radial basis function*) foi proposto por Hopfield (1995). Classicamente, esse modelo de neurônio tem como característica principal a ocorrência de funções de ativação – usualmente gaussianas – sintonizadas em torno de uma região. A resposta de um neurônio de base radial é tipicamente incrementada à medida que um vetor de entrada $X = \{x_1, x_2, \dots, x_n\}$ se aproxima do vetor que representa o centro da função de ativação $\Delta = \{\delta_1, \delta_2, \dots, \delta_n\}$.

Seja, então, um neurônio SRM j estimulado em suas n sinapses por disparos de seus n predecessores $i_1..i_n \in \Gamma_j$, com as seguintes características:

- O limiar ϑ_j do neurônio j é alto em relação às contribuições recebidas de cada sinapse, e somente é atingido quando todos os pulsos pré-sinápticos excitatórios são disparados simultaneamente;
- Associado a cada uma das n sinapses de j existe um atraso de propagação $\delta_{i_1j}.. \delta_{i_nj} \in \Delta_j$, de forma que o neurônio j recebe o pulso proveniente do neurônio i_k somente em $t_{i_k} + \delta_{i_kj}$.

Este processo é ilustrado pela figura 2.11, onde a aplicação dos pulsos $t_{i_1}, \dots, t_{i_n}$

nas sinapses do neurônio j faz com que o seu potencial $\mu_j(t)$ atinja o limiar de disparo ϑ_j se e somente se:

$$X_i = \{x_{i_1}, x_{i_2}, \dots, x_{i_n}\} \approx \Delta_j = \{\delta_{j_1}, \delta_{j_2}, \dots, \delta_{j_n}\}, \quad (2.18)$$

onde: X_i é o vetor que contém intervalos de tempo associados aos pulsos emitidos pelos neurônios $i \in \Gamma_j$; x_{i_k} é o intervalo de tempo associado à emissão do pulso pelo neurônio i_k ; Δ_j é o vetor que contém os atrasos axonais de j ; δ_{j_k} é o atraso axonal associado à k -ésima sinapse do neurônio j .

Desta forma, estabelece-se que um neurônio SRM pode ser modelado para implementar um neurônio RBF.

Figura 2.11: Um neurônio SRM implementando a função típica de um neurônio RBF.

2.4 Modelos de rede

Dada a ampla gama de modelos de neurônios pulsados (ver figura 2.2), é natural que redes desses neurônios tenham sido propostas com uma enorme variedade de modelos. Além do tipo de neurônio envolvido, alguns dos parâmetros relevantes na concepção de RNPs têm sido: a topologia da rede, a presença e o formato do estímulo ou inibição entre os neurônios, os códigos de entrada e saída, entre outros. Face à impossibilidade prática de apresentar aqui todas essas arquiteturas, abordaremos apenas uma definição formal válida para redes de neurônios pulsados, e com maior profundidade na seção 2.5 o modelo da rede neural pulsada com neurônios SRM-RBF, utilizada neste trabalho.

Do ponto de vista matemático, uma rede de neurônios pulsados computa uma função que mapeia um vetor de algumas séries temporais $\langle T_i \rangle_{i \in I}$ em um vetor

de algumas outras séries temporais $\langle T_j \rangle_{j \in J}$ (MAASS; BISHOP, 1999, p. 56). Uma definição matemática com relativa generalidade para redes dessa natureza foi proposta por Ruf (1998). Seja então uma rede de neurônios pulsados (RNP)

definição
de RNP

definida como contendo:

- Um grafo direto finito $(\mathcal{N}, \mathcal{S})$, onde $\mathcal{N}$ são neurônios e $\mathcal{S}$ são sinapses;
- Dois subconjuntos $I \subseteq \mathcal{N}$ e $J \subseteq \mathcal{N}$;
- Para cada neurônio $i \in \mathcal{N} - I$ uma função limiar: $\vartheta_i(t) : \mathbb{R}_+ \rightarrow \mathbb{R} \cup \infty$;
- Para cada sinapse $s \in \mathcal{S}$ uma função-resposta $\epsilon_s : \mathbb{R}_+ \rightarrow \mathbb{R}$ e um peso $w_s \in \mathbb{R}$.

Assume-se ainda que:

- O conjunto $T_i \subseteq \mathbb{R}_+$ de disparos dos neurônios de entrada $i \in I$ é fornecido como entrada da rede;
- Para um neurônio $i \in \mathcal{N} - I$ os tempos de disparo T_i são definidos recursivamente, ou seja, referindo-se a tempos de disparo de outros neurônios.

2.5 A RNP-SRM-RBF

Um modelo de rede que tem despertado particular atenção – principalmente por aceitar algoritmos explícitos de treinamento – é a rede de neurônios pulsados do tipo SRM implementando funções de base radial, apresentada nesta seção. A exposição da arquitetura da rede encontra-se dividida em duas partes. Inicialmente apresenta-se a **arquitetura básica** da rede de forma similar à proposta por Hopfield (1995) e Gerstner (1996), que fornece um modelo elementar que pode ser implementado de diferentes formas. A seguir, apresenta-se a **arquitetura com o desdobramento das sinapses**, destacando especificidades da implementação da arquitetura básica segundo Natschläger e Ruf (1998). Após a apresentação da arquitetura da rede, apresenta-se a **codificação dos dados de entrada** utilizando os campos receptivos gaussianos propostos por Bohte (2003). A rede de Natschläger e Ruf (1998) com a codificação de Bohte (2003) será utilizada no decorrer deste trabalho.

2.5.1 Arquitetura básica

Sejam duas camadas I e J de neurônios totalmente conectados como o ilustrado na figura 2.12. Essas duas camadas são denominadas camadas de entrada e saída, respectivamente. Os neurônios da camada de entrada apenas transmitem pulsos aos neurônios da camada de saída, que são os únicos a realizar processamento sobre esses sinais. Muito embora nos trabalhos originais de Gerstner (1996) e Natschläger e Ruf (1998) tenha-se adotado um neurônio de saída do tipo integra-e-dispara, adota-se neste trabalho – assim como em Bohte (2003) – uma rede com neurônios de saída modelados como SRM operando como neurônios RBF, como o apresentado na seção 2.3.4. Desta forma, tipicamente um neurônio de saída dispara se o seu vetor de atrasos axonais Δ se aproximar do vetor de entrada X . É importante observar que nesta rede os neurônios da camada de saída tipicamente não fazem uso da modelagem de refração e cada neurônio da camada de entrada é permitido disparar uma única vez.

a) Arquitetura da rede; b) Detalhamento do processamento interno do neurônio j_1 . O neurônio dispara sempre que o vetor de tempos de entrada X for similar ao vetor de atrasos axonais Δ .

Figura 2.12: Arquitetura básica da rede de SRM-RBFs.

2.5.2 Arquitetura com desdobramento das sinapses

O aprendizado da RNP-SRM-RBF (que será discutido no capítulo 3) tem por objetivo sintonizar adequadamente os parâmetros $\delta \in \Delta$ dos neurônios, o que não é um processo trivial. Para facilitar esta sistemática, Natschläger e Ruf (1998) propuseram uma alteração na arquitetura da rede desdobrando as conexões sinápticas. Nesta abordagem, uma conexão entre dois neurônios i e j (com $i \in I$ e $j \in J$) é sempre implementada através de d sinapses com diferentes atra-

sos de propagação e pesos sinápticos associados. Esta sistemática é apresentada esquematicamente na figura 2.13. Segundo esta abordagem, o processo de sintonizar o parâmetro δ_{ij} pode ser revisto como o processo de deixar ativo um único atraso axonal δ_{ij}^k (com $1 \leq k \leq d$) em cada conexão entre i e j , ou, em outros termos, fazer com que apenas um w_{ij}^k (com $1 \leq k \leq d$) seja diferente de zero em cada conexão entre dois neurônios i e j . Nessa abordagem é usual que os atrasos assumam os valores:

$$\{\delta_{ij}^1, \delta_{ij}^2, \dots, \delta_{ij}^d\} = \{0, 1, \dots, d-1\} \quad (2.19)$$

a) Arquitetura da rede; b) Detalhamento da conexão entre os neurônios i_n e j_m assinalada em “a” e implementada com d sinapses: um pulso gerado em i_n será recebido d vezes em j , com diferentes latências.

Figura 2.13: Arquitetura da rede e conexão entre os neurônios.

2.5.3 Codificação dos dados de entrada

A rede proposta por Natschläger e Ruf (1998) apresenta como sua principal limitação para aplicações práticas a falta de uma sistemática para codificar dados reais para a forma pulsada. Bohte (2003) propôs a utilização de campos receptivos gaussianos como forma de codificar dados de entrada, que tipicamente originam um grande número de disparos. Esta sistemática de codificação é descrita nesta seção.

Seja cada uma das componentes do vetor de entrada de dimensão n aplicada a um conjunto de g gaussianas cobrindo toda a extensão dos dados de entrada, como o mostrado na figura 2.14. A camada de entrada I fica agora composta por $n.g$ neurônios, e as gaussianas determinam o atraso no disparo de cada um dos $n.g$ neurônios da camada de entrada. Se a gaussiana k for estimulada de forma

ótima, o neurônio i_k gerará um pulso na primeira iteração. Se a gaussiana k quase não sofrer estímulo, i_k disparará após h iterações, onde h é a altura da gaussiana. Em outros termos, o k -ésimo neurônio da camada de entrada disparará de acordo com o estímulo da gaussiana k .

A codificação de um valor contínuo (130) em pulsos temporais para os 8 neurônios da camada de entrada. O espaço de entrada $[0:255]$ é coberto por $g=8$ campos receptivos gaussianos. A latência do k -ésimo neurônio encontra-se vinculada à excitação da gaussiana k .

Figura 2.14: Campos receptivos gaussianos.

Considerando $x : [x_{min} x_{max}]$, o centro de cada uma das g gaussianas pode ser expresso por (BOHTE, 2003):

$$c_k = x_{min} + \frac{2k-3}{2} \cdot \frac{x_{max} - x_{min}}{k-2}, \text{ para } k = 3, 4, \dots, g \quad (2.20)$$

onde: c_k é o centro da k -ésima gaussiana; x_{min} é o valor mínimo assumido pela variável de entrada; x_{max} é o valor máximo assumido pela variável de entrada; g é o número de gaussianas presentes no campo receptivo.

A metodologia aqui apresentada permitiu a codificação de dados temporais para a rede de Natschläger e Ruf (1998) e a utilização de redes com neurônios estimulados por gaussianas com desvios padrão distintos mostrou-se capaz de ampliar a precisão e capacidade de RNPs atuando como classificadores de padrões bidimensionais (BOHTE, 2003).

2.5.4 Processamento típico

Considere o modelo proposto por Natschläger e Ruf (1998) com o desdobramento das sinapses. Seja, por simplificação, uma rede com as seguintes características:

- Um neurônio da camada de saída j iniciado com potencial $\mu_j = 0$;
- j possui conexões com os $n.g$ neurônios da camada de entrada;
- O número de sinapses d entre as conexões de i e j é muito grande;
- Os neurônios da camada de entrada emitem pulsos nos tempos t_i ;
- Todos os pesos w_{ij}^k associados às sinapses da rede são iguais a 1;
- As funções ϵ_{ij} possuem constante de tempo de descida muito maior que a constante de tempo de subida ($\tau_m \gg \tau_s$).

Nessas condições, as contribuições de cada uma das $n.g.d$ sinapses podem ser aproximadas por funções degrau unitário. Esta situação é ilustrada na figura 2.15. O disparo do primeiro pulso (no caso ilustrado, em t_{i_2}) fará com que o neurônio j receba uma seqüência de d pulsos provenientes das d sinapses entre o neurônio que disparou (no caso ilustrado, i_2) e j . Como d é muito grande, estas contribuições incrementarão o potencial de j na forma aproximada de uma reta durante todo o processo competitivo. A inclinação desta reta permanecerá constante até que um novo pulso (no caso ilustrado, t_{i_1}) inicie a contribuição de uma nova conexão (no caso a conexão entre i_1 e j) a u_j . Após a chegada do último pulso, a inclinação do incremento em u_j passa a ser constante e da ordem de $\theta = \arctg(n.g)$.

Figura 2.15: Potencial de um neurônio da camada de saída recebendo contribuições de todas as sinapses.

Pode-se estimar a inclinação desta reta com pesos iniciados aleatoriamente por:

$$\theta = \arctg(n.g.w_{medio}) \quad (2.21)$$

onde: n é o número de dimensões de entrada; g é o número de gaussianas; w_{medio} é o peso sináptico médio adotado no início do processo de aprendizado.

Após o treinamento da rede, uma constante τ_m baixa terá importante implicação no processo de evitar o disparo para instâncias distantes do centro do neurônio. Em tempo de aprendizado, a redução na constante τ_m tende a reduzir θ , e constantes excessivamente pequenas podem impedir que o neurônio dispare.

O modelo completo da RNP-SRM-RBFs utilizando campos receptivos gaussianos é apresentada na figura 2.16. Esta figura procura concentrar todos os parâmetros relevantes deste modelo, de forma a facilitar a leitura das futuras seções. O modelo RNP-SRM-RBF será tratado no decorrer do texto simplesmente como modelo da RNP.

2.6 Resumo

Este capítulo apresentou uma visão geral dos principais modelos de redes neurais pulsadas, com foco nos modelos de neurônios pulsados. De forma geral este texto procurou salientar que:

- i. Diversos modelos de rede têm sido propostos para a computação baseada em pulsos, e a taxonomia desse conjunto de modelos – denominado RNPs – ainda não se encontra sedimentada na literatura; e
- ii. Esforços têm sido feitos no sentido de estabelecer a equivalência entre alguns dos modelos de neurônios pulsados, mas suas relações ainda não se encontram formalmente estabelecidas na maioria dos casos.

Apresentada uma visão geral desses modelos, particular atenção foi reservada ao modelo SRM, que tem se firmado como um modelo genérico de neurônio pulsado, com reconhecido poder computacional e possibilidade de emprego em vários dos modelos de rede. Em especial as RNP-SRM-RBFs emergem como um dos primeiros modelos de RNP – senão o primeiro – com possibilidade de acolher algoritmos explícitos de aprendizado.

Esse conjunto de fatores aliado à possibilidade de computar utilizando um código não usual no domínio das RNAs (o código temporal) fez da RNP-SRM-RBF a rede adotada neste trabalho.

Figura 2.16: Modelo da rede neural pulsada.

3 Aprendizado em Redes Neurais Pulsadas

“Aprendizado é o processo pelo qual parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizado é determinado pela maneira através da qual as mudanças nos parâmetros acontecem” (HAYKIN, 1999, p. 50).

O contexto do aprendizado colocado por Haykin (1999) tem sido aplicado às RNPs de pelo menos duas formas distintas. Em algumas redes baseadas em osciladores, por exemplo, o aprendizado não é apresentado explicitamente na forma de um algoritmo, mas encontra-se embutido no modelo do neurônio. Tipicamente o processo de sincronização dos pulsos emerge da associação de uma coleção de neurônios (KINSER, 1997) (WANG, 2001) (KATADA; OHKUDA, 2003).

Em se tratando de RNPs com algoritmos explícitos de aprendizado – tipicamente a maior parte das RNPs que utilizam o modelo do SRM –, três formas distintas de aprendizado têm se destacado:

- Aprendizado supervisionado;
- Aprendizado não-supervisionado; e
- Aprendizado por reforço.

No aprendizado supervisionado, os parâmetros internos da rede são determinados através da diminuição de uma medida de erro definida entre valores de saída dados em exemplos de mapeamentos entrada-saída e as saídas atingidas pela rede para as mesmas entradas (DELORME; THORPE, 2001) (BOHTE; KOK; la POUTRé, 2002a) (MOORE, 2002) (BOOIJ, 2004). Já no aprendizado não-supervisionado, a rede determina seus parâmetros sem o conhecimento da resposta desejada, tipicamente com base nas informações de número de classes desejadas e topologia

da rede (GERSTNER, 1996) (WALLIS; BADDELEY, 1997) (AMEMORI; ISHII, 1998) (CHOE; MIIKKULAINEN, 1998) (NATSCHLÄGER; RUF, 1998) (RUF; SCHMITT, 1998) (BOHTE; KOK; la POUTRÉ, 2000) (BOHTE; la POUTRÉ; KOK, 2002) (KUROYANAGI; IWATA, 2002) (BOHTE, 2003).

No aprendizado por reforço alguma função heurística é utilizada para descrever a qualidade da resposta da rede a uma dada entrada. A qualidade está representada por recompensas ou penalização. A maximização (ou minimização) da recompensa (ou penalização) oferecida à rede ao longo do aprendizado conduz para a recalibração adequada de seus parâmetros (TAKITA; OSANA; HAGIWARA, 2000).

Esta seção aborda em linhas gerais alguns dos algoritmos propostos para o aprendizado das redes neurais pulsadas, focando aqueles mais diretamente relacionados com as redes que possam utilizar neurônios com o modelo SRM. Para tanto, em alguns casos será realizada uma revisão de alguns princípios do aprendizado clássico, transportando esses conceitos ao universo das redes pulsadas. É importante ressaltar que cada algoritmo é específico para uma arquitetura de RNP, que será detalhada em cada caso à medida da necessidade.

A seção 3.1 apresenta algoritmos para o aprendizado supervisionado. A seção 3.2 aborda o aprendizado não supervisionado. Finalmente, a seção 3.3 aborda o aprendizado por reforço.

3.1 Aprendizado supervisionado

Esta seção apresenta as abordagens temporais da lei de Hebb e da retropropagação do erro propostas para as RNPs.

3.1.1 Lei de Hebb

O postulado do aprendizado de Hebb é a mais antiga e conhecida regra de aprendizado (HAYKIN, 1999). Donald Olding Hebb (1904-1985) propôs como base do aprendizado associativo em nível celular que *“quando um axônio da célula A está perto o suficiente para excitar a célula B e repetidamente ou persistentemente toma parte em seu disparo, algum processo crescente ou mudança metabólica toma parte em uma ou ambas as células de forma que a eficiência de A como uma das células que dispara B é aumentada”* (HEBB, 1949, p.62).

sinapse i de um neurônio j com sinais pré e pós-sinápticos denotados por x_i e y_j , respectivamente. A correção aplicada sobre o peso sináptico w_{ij} no tempo k é expressa por (HAYKIN, 1999):

$$\Delta w_{ij}(k) = f\left(y_j(k), x_i(k)\right), \quad (3.1)$$

onde: $\Delta w_{ij}(k)$ é a taxa de variação do i -ésimo peso sináptico do neurônio j na iteração k ; $f(\cdot, \cdot)$ é uma função sobre os sinais pré e pós-sinápticos; $y_j(k)$ é o sinal pós-sináptico do neurônio j na iteração k ; $x_i(k)$ é o sinal pré-sináptico presente na sinapse i do neurônio j na iteração k ; k é a iteração.

No contexto das RNAs, uma forma usual da lei de Hebb é:

$$\Delta w_{ij}(k) = \eta \cdot (y_j(k) \cdot x_i(k)), \quad (3.2)$$

onde: $\Delta w_{ij}(k)$ é a taxa de variação do i -ésimo peso sináptico do neurônio j na iteração k ; η é uma constante positiva denominada taxa de aprendizado; $x_i(k)$ é o sinal pré-sináptico na iteração k ; $y_j(k)$ é o sinal pós-sináptico na iteração k .

Admita que um neurônio i , predecessor imediato do neurônio j , dispare duas vezes nos tempos t_i^1 e t_i^2 , fazendo com que o neurônio j dispare no tempo t_j . Admita ainda que a excitação pré-sináptica do neurônio j é linear. A figura 3.1a ilustra a situação onde $t_j < t_i^2$, e a figura 3.1b ilustra o caso onde $t_j > t_i^2$.

a) O neurônio j dispara antes do segundo pulso de i ; b) O neurônio j dispara após o segundo pulso do neurônio i .

Figura 3.1: Disparo de um NP com excitação pré-sináptica linear.

Considere agora a tarefa de treinar o neurônio j de forma que $t_j = t_i^2$. A lei de Hebb pode então ser colocada da seguinte maneira: “o objetivo é que a sinapse altere seus pesos de forma que um pulso pré-sináptico faça o neurônio pós-sináptico disparar depois de um tempo determinado. Então, o peso é modificado

de acordo com o sinal de erro dado pela diferença de tempo entre o segundo disparo pré-sináptico e o pulso pós-sináptico” (RUF, 1998, p. 53).

Lei de
Hebb em
um NP

Nesse contexto, uma formulação natural para a lei de Hebb aplicada ao neurônio pulsado é dada por (MAASS, 1997) (RUF, 1998):

$$\Delta w_{ij} = \eta \cdot (t_j - t_i^2), \quad (3.3)$$

onde: Δw_{ij} é a taxa de variação do i -ésimo peso sináptico do neurônio j ; η é a taxa de aprendizado; t_j é o instante do disparo do neurônio j ; t_i^2 é o instante do segundo disparo do neurônio i ; $t_j - t_i^2$ é o erro do disparo do neurônio j .

Note que a equação 3.3 é um subcaso da equação 3.1 aplicável ao domínio dos neurônios pulsados. Vale ressaltar que a aplicação sucessiva da lei de Hebb tem assegurada uma convergência exponencial (HAYKIN, 1999), que, no caso de sua versão pulsada, levará a um peso sináptico com valor dado por (RUF, 1998):

$$w_{ij} \rightarrow \frac{\vartheta_j - \mu_j^0(t)}{t_i^2 - t_i^1}, \quad (3.4)$$

onde: w_{ij} é o peso sináptico da i -ésima sinapse do neurônio j ; ϑ_j é o limiar de disparo do neurônio j ; $\mu_j^0(t)$ é o valor inicial do potencial do neurônio j ; t_i^2 é o instante do segundo disparo do neurônio i ; t_i^1 é o instante do primeiro disparo do neurônio i .

A sistemática aqui apresentada é válida para o aprendizado de um neurônio com uma única sinapse. No entanto, freqüentemente um neurônio recebe estímulos de um número maior de sinapses. Segundo Ruf (1998), uma possibilidade para o aprendizado polissináptico é a aplicação sucessiva desta regra de aprendizado de forma que o neurônio j receba estímulos em apenas uma de suas sinapses a cada ciclo de aprendizado.

3.1.2 Retropropagação do erro

retropro-
pagação
em RNPs

O primeiro trabalho aplicando a retropropagação do erro a uma rede de neurônios pulsados é atribuído a Bohte, Kok e la Poutre (2002a). Esse algoritmo conhecido como *SpikeProp* é descrito nesta seção. Seja uma RNP onde existem três camadas de neurônios: I (camada de entrada), $\mathcal{O}$ (camada oculta) e J (camada de saída). As camadas de saída e oculta possuem neurônios do tipo SRM e adota-se a sistemática do desdobramento da sinapses. O objetivo do treinamento é ensinar à rede um conjunto de tempos de disparo $T_j =$

$\{t_{j1}, t_{j2}, \dots, t_{jm}\}$ nos neurônios $j \in J$ para um dado conjunto de padrões $T_i = \left\{ \{t_{i1}^1, t_{i2}^1, \dots, t_{in}^1\}, \{t_{i1}^2, t_{i2}^2, \dots, t_{in}^2\}, \dots, \{t_{i1}^{n_p}, t_{i2}^{n_p}, \dots, t_{in}^{n_p}\} \right\}$, composto pelos tempos de disparo de cada neurônio $i \in I$, onde n_p é o número de padrões a serem apresentados.

Uma possível função para a avaliação do desempenho do sistema é a função de erro dada por:

$$E = \frac{1}{2} \sum_{j \in J} (t_j - \tilde{t}_j)^2, \quad (3.5)$$

onde: E é a função erro; j é um neurônio da camada de saída J ; t_j é o tempo de disparo atual do neurônio j ; $\tilde{t}_j$ é o tempo de disparo desejado do neurônio j .

Para a retropropagação do erro, cada sinapse é tratada como uma conexão k independente das demais com peso w_{ij}^k . Dessa forma, um passo sobre a curva convexa do erro quadrático de um ponto arbitrário em direção ao ponto de mínimo global pode ser dado utilizando:

$$\Delta w_{oj}^k = -\eta \cdot \frac{\partial E}{\partial w_{oj}^k}, \quad (3.6)$$

onde: w_{oj}^k é o peso sináptico atribuído à k -ésima sinapse entre o e j ; η é a taxa de aprendizado; j é um neurônio da camada de entrada J .

De forma análoga à retropropagação para redes neurais convencionais, o equacionamento da variação do peso sináptico é expresso de formas diferentes para neurônios da camada de saída $j \in J$ e das demais camadas. Tais equacionamentos serão detalhados a seguir. Uma observação, contudo, deve ser feita: dada a natureza das redes pulsadas, o erro do sistema somente pode ser medido após o disparo do neurônio da camada de saída, o que ocorre no instante $t = t_j$.

Neurônios da camada de saída. Uma vez que $E = f(t_j, \tilde{t}_j)$ com $\tilde{t}_j$ constante (vide equação 3.5), a derivada parcial na equação 3.6 pode ser expandida como o indicado em 3.7. Por sua vez, $t_j = f(\mu_j(t), \vartheta_j)$ com ϑ_j constante (vide equação 2.11). Desta forma, obtém-se a expressão mostrada na equação 3.8.

$$\frac{\partial E}{\partial w_{oj}^k} = \frac{\partial E}{\partial t_j}(t_j) \cdot \frac{\partial t_j}{\partial w_{oj}^k}(t_j), \quad (3.7)$$

$$\frac{\partial E}{\partial w_{oj}^k} = \underbrace{\frac{\partial E}{\partial t_j}(t_j)}_{(1)} \cdot \underbrace{\frac{\partial t_j}{\partial \mu_j(t)}(t_j)}_{(2)} \cdot \underbrace{\frac{\partial \mu_j(t)}{\partial w_{oj}^k}(t_j)}_{(3)}, \quad (3.8)$$

onde: E é o erro do sistema; w_{oj}^k é o peso sináptico da k -ésima conexão entre os neurônios o e j ; t_j é o tempo de disparo atual do neurônio j ; $\mu_j(t)$ é o potencial do neurônio j .

Uma vez que se trata de um neurônio da camada de saída, o termo em (1) da equação 3.8 pode ser desenvolvido como:

$$\frac{\partial E}{\partial t_j}(t_j) = \frac{\partial}{\partial t_j} \left\{ \frac{1}{2} \cdot (t_j - \tilde{t}_j)^2 \right\} = t_j - \tilde{t}_j, \quad (3.9)$$

onde: E é o erro do sistema; t_j é o tempo de disparo atual do neurônio j ; $\tilde{t}_j$ é o tempo de disparo desejado do neurônio j .

O termo (2) da equação 3.8 requer uma análise mais detalhada, pois não é uma tarefa trivial determinar t_j como uma função de $\mu_j(t)$, já que $\mu_j(t)$ é uma função bastante complexa (vide equação 2.15).

Uma alternativa para o desenvolvimento deste termo é assumir que $\mu_j(t)$ varia de forma aproximadamente linear em uma pequena faixa de tempo próxima ao disparo, ou seja, $\mu_j(t) = \theta \cdot t$ para $t \rightarrow t_j$.

Nesse ponto, como o ilustrado na figura 3.2, a variação do tempo de disparo (Δt_j) relaciona-se com a variação no potencial do neurônio ($\Delta \mu_j(t)$) segundo a expressão: $\Delta t_j = -\Delta \mu_j(t)/\theta$. Como $\theta = \partial \mu_j(t)/\partial t$, e como a variação do potencial com relação ao tempo é conhecida (vide equação 2.15), a equação pode ser reescrita como:

$$\frac{\partial t_j}{\partial \mu_j(t)}(t_j) \approx \frac{-1}{\theta} = \frac{-1}{\frac{\partial \mu_j(t)}{\partial t}(t_j)} = \frac{-1}{\sum_{o \in \Gamma_j} \sum_k w_{oj}^k \cdot \frac{\partial \epsilon_{oj}^k(t)}{\partial t}(t_j)}, \quad (3.10)$$

onde: o é um neurônio pré-sináptico a j ; t_j é o tempo de disparo atual do neurônio j ; $\mu_j(t)$ é o potencial de disparo do neurônio j ; w_{oj}^k é o k -ésimo peso sináptico entre os neurônios o e j ; ϵ_{oj}^k é a função de resposta do neurônio j à k -ésima conexão com o neurônio o ; θ é a relação entre a variação do tempo de disparo do neurônio e a variação em seu potencial, segundo a aproximação linear.

Adaptado de (BOHTE, 2003).

Figura 3.2: Aproximação linear do potencial do neurônio no momento de seu disparo.

Finalmente, o termo (3) da equação 3.8 pode ser expresso por:

$$\frac{\partial \mu_j(t)}{\partial w_{oj}^k} = \frac{\partial}{\partial w_{oj}^k} \left\{ \sum_{o \in \Gamma_j} \sum_{k=1}^d w_{oj}^k \cdot \epsilon_{oj}^k(t) \right\} = \epsilon_{oj}^k(t_j), \quad (3.11)$$

onde: $\mu_j(t)$ é o potencial do neurônio j ; w_{oj}^k é o k -ésimo peso sináptico entre os neurônios o e j ; ϵ_{oj}^k é a função de resposta do neurônio j à sua k -ésima conexão com o neurônio o .

Desta forma, a equação 3.6 pode ser reescrita com os valores obtidos em 3.9, 3.10 e 3.11 da forma:

$$\Delta w_{oj}^k = -\eta \cdot \epsilon_{oj}^k(t_j) \cdot \delta_j, \text{ com } \delta_j = \frac{(\tilde{t}_j - t_j)}{\sum_{o \in \Gamma_j} \sum_{k=1}^d w_{oj}^k \cdot \left(\frac{\partial \epsilon_{oj}^k(t)}{\partial t} (t_j) \right)}, \quad (3.12)$$

onde: Δw_{oj}^k é a variação do k -ésimo peso sináptico entre os neurônios o e j ; η é a taxa de aprendizado; ϵ_{oj}^k é a k -ésima função de resposta do neurônio j ao neurônio o ; $\tilde{t}_j$ é o tempo desejado de disparo para o neurônio j ; t_j é o tempo atual de disparo para o neurônio j .

A equação 3.12 será referenciada no decorrer deste texto como equação da retropropagação para os neurônios da camada de saída.

Neurônios das camadas ocultas. De forma análoga à equação 3.8, a retropropagação do erro para neurônios da camada $\mathcal{O}$ definidos por $o \in \mathcal{O}$ pode

ser expressa por:

$$\frac{\partial E}{\partial w_{io}^k} = \underbrace{\frac{\partial E}{\partial t}(t_o)}_{(1)} \cdot \underbrace{\frac{\partial t}{\partial \mu_o(t)}(t_o)}_{(2)} \cdot \underbrace{\frac{\partial \mu_o(t)}{\partial w_{io}^k}(t_o)}_{(3)}, \quad (3.13)$$

onde: E é o erro do sistema; w_{io}^k é o peso sináptico da k -ésima conexão entre os neurônios i e o ; t_o é o tempo de disparo atual do neurônio o ; $\mu_o(t)$ é o potencial do neurônio o .

O termo (1) da equação 3.13 pode ser expandido relacionando o erro da saída da rede com o neurônio o , como o mostrado na equação:

$$\frac{\partial E}{\partial t_o} = \sum_{j \in \Gamma_o} \underbrace{\frac{\partial E}{\partial t_j} \cdot \frac{\partial t_j}{\partial \mu_j(t)}}_{\delta_j} \cdot \frac{\partial \mu_j(t)}{\partial t_o} = \sum_{j \in \Gamma_o} \delta_j \cdot \frac{\partial \mu_j(t)}{\partial t_o} \quad (3.14)$$

$$= \sum_{j \in \Gamma_o} \delta_j \cdot \left\{ \frac{\partial}{\partial t_o} \left\{ \sum_{o \in \mathcal{O}} \sum_{k=1}^d w_{oj}^k \cdot \epsilon_{oj}^k(t_j) \right\} \right\} = \sum_{j \in \Gamma_o} \delta_j \cdot \sum_k w_{oj}^k \cdot \frac{\partial}{\partial t_o} \{ \epsilon_{oj}^k(t_j) \}, \quad (3.15)$$

onde: E é o erro do sistema; t_o é o tempo de disparo atual do neurônio o ; Γ_o é o conjunto dos neurônios pós-sinápticos ao neurônio o ; j é um neurônio pós-sináptico a o ; $\mu_j(t)$ é o potencial do neurônio j ; ϵ_{oj}^k é a função de resposta do neurônio j à sua k -ésima conexão com o neurônio o ; δ_j é o termo calculado na equação 3.12.

Os termos (2) e (3) da equação 3.13 podem ser conseguidos através de analogia com a equação 3.12. Finalmente, a equação 3.6 para neurônios de outras camadas que não a de saída pode ser reescrita como:

$$\Delta w_{io}^k = -\eta \cdot \epsilon_{io}^k(t_o) \cdot \delta_o, \text{ com } \delta_o = \frac{\sum_j \left\{ \delta_j \cdot \sum_k w_{oj}^k \left(\frac{\partial \epsilon_{oj}^k(t)}{\partial t_o} \right) \right\}}{\sum_{i \in \Gamma_o} \sum_{k=1}^d w_{io}^k \left(\frac{\partial \epsilon_{io}^k(t_o)}{\partial t_o} \right)}, \quad (3.16)$$

onde: w_{io}^k é o k -ésimo peso sináptico entre os neurônios i e o ; η é a taxa de aprendizado; ϵ_{io}^k é a função de resposta do neurônio i associada à sua k -ésima conexão com o neurônio o ; i é um neurônio pré-sináptico ao neurônio o ; j é um neurônio pós-sináptico ao neurônio o ; t_o é o tempo de disparo atual do neurônio o .

A equação 3.16 é a expressão da retropropagação do erro para neurônios das

camadas de entrada e oculta.

O desenvolvimento do *SpikeProp* é extremamente recente, o que dificulta uma avaliação mais ampla sobre suas implicações. Os primeiros resultados da aplicação deste algoritmo são apresentados em (BOHTE, 2003).

3.2 Aprendizado não supervisionado

Um modelo de aprendizado específico para um neurônio SRM implementando função de base radial foi proposto por Gerstner (1996) e, posteriormente, um modelo não supervisionado foi estendido para uma rede desses neurônios (NATSCHLÄGER; RUF, 1998). Recentemente, avanços significativos foram incorporados nessa sistemática de aprendizado (BOHTE; KOK; la POUTRÉ, 2000) (BOHTE, 2003), ampliando a capacidade e a precisão dessas redes, e tornando-as propícias a tratar dados reais.

O objetivo do aprendizado é fazer com que cada neurônio RBF armazene o centro de um grupo, ou, em outros termos, fazer com que dispare o neurônio da camada de saída cujo centro mais se aproximar do padrão de pulsos fornecido pela camada de entrada. O processo proposto por Natschläger e Ruf (1998) para isto é basicamente o descrito a seguir.

Inicialmente os pesos sinápticos são aleatoriamente adotados. Após a apresentação de um vetor de entrada à rede, o neurônio da camada de saída que disparar primeiro será considerado o neurônio vencedor do processo competitivo. Tipicamente, apenas os pesos sinápticos do neurônio RBF vencedor são atualizados utilizando uma função de aprendizado Hebbiana $L(\Delta t)$.

A idéia básica para a utilização do princípio Hebbiano é a seguinte: se um neurônio recebeu um pulso de uma sinapse imediatamente antes de disparar, então o peso relativo a esta sinapse deve ser incrementado. Sinapses que receberam estímulos muito antes ou muito depois do disparo do neurônio não tiveram contribuição em seu disparo, e, portanto, devem ser decrementadas. Uma possível expressão para a atualização dos pesos sinápticos é (NATSCHLÄGER; RUF, 1998):

$$\Delta w_{ij}^k = \eta \cdot L(\Delta t) = \eta \cdot \left((1 - b) \cdot e^{-\frac{(\Delta t - c)^2}{\beta^2}} + b \right), \quad (3.17)$$

onde: Δw_{ij}^k é a taxa de variação do peso da k -ésima conexão entre o neurônio i e o neurônio j ; η é a taxa de aprendizado; $L(\cdot)$ é uma função de aprendizado sobre o argumento; b determina o valor do decremento dos pesos sinápticos; Δt é a

diferença de tempo entre o recebimento de um pulso em uma sinapse e o disparo do neurônio em questão; c determina o centro do pico da curva de aprendizado; η posiciona o comprimento da parte positiva da função de aprendizado. Tipicamente, espera-se que o centro do grupo em questão mova-se na direção do vetor de entrada corrente. A expressão 3.17 é ilustrada graficamente na figura 3.3.

Extraído de (BOHTE, 2003), com alterações.

Figura 3.3: Gráfico da função de aprendizado gaussiana.

As sucessivas aplicações desta regra de aprendizado geralmente levam os pesos sinápticos à saturação em w_{max} ou w_{min} , selecionando os atrasos de propagação relevantes para a determinação do centro dos grupos.

A figura 3.4 ilustra o processo de adaptação dos pesos sinápticos e potencial de um neurônio pulsado através da regra de aprendizado apresentada na equação 3.17. A aplicação do aprendizado *não-supervisionado* aqui descrito apresenta algumas características que merecem destaque quando aplicadas a uma rede com SRM-RBFs: *i)* a antecipação do disparo; *ii)* o deslocamento do vetor de pesos; *iii)* os padrões presentes no vetor de pesos sinápticos, *iv)* a sensibilidade ao peso máximo e *v)* a auto-inibição.

3.2.1 Antecipação do disparo

A idéia básica do treinamento é aumentar a sensibilidade do neurônio a alguns estímulos para que disparem sempre que os receber. Quando isso ocorre, no entanto, os neurônios RBF passam a disparar mais cedo com as sucessivas rerepresentações dos padrões de treinamento (NATSCHLÄGER; RUF, 1998). Tendo em vista que o tempo de disparo é uma medida da distância entre a instância apresentada à rede e o centro do grupo implementado pelo neurônio, esta anteci-

Esq. RNP com $n.g$ neurônios de entrada e um neurônio de saída. **Dir.** Potencial do neurônio j . a) A rede – por simplificação – é iniciada com todos os pesos sinápticos iguais e o neurônio j dispara em t_j . O algoritmo de aprendizado identifica os pulsos que antecederam o disparo, incrementa os pesos sinápticos correspondentes a eles e decrementa os demais; b) Os pesos foram atualizados e o neurônio j passa a disparar mais cedo quando a mesma instância de entrada é apresentada à rede. O processo se repete, mas a regra de aprendizado atualiza um outro subconjunto de W , ocasionando um deslocamento no vetor de pesos sinápticos; c) O processo se repete até que o tempo de disparo t_j encontre-se próximo do tempo de disparo do último neurônio de entrada, selecionando os atrasos δ .

Figura 3.4: Adaptação dos pesos sinápticos e potencial através do algoritmo de aprendizado.

pação, por si, não pode ser encarada como um problema, mas uma característica inerente a este modelo.

3.2.2 Deslocamento do vetor de pesos

O processo de treinamento proposto por Natschläger e Ruf (1998) baseia-se na idéia de incrementar o peso da entrada que é estimulada imediatamente antes do disparo do neurônio. Considere, por exemplo, o caso em que o estímulo em um neurônio faça-o disparar no instante t_1 . A aplicação da função de aprendizado estimula o aumento de um certo peso sináptico w_s associado à sinapse s , estimulada imediatamente antes de t_1 . Considere agora que o neurônio é novamente excitado com estímulo similar ao primeiro. Dada sua nova configuração de pesos (que o tornou mais sensível ao estímulo) ele agora dispara no instante $t_2 < t_1$. A função de aprendizado atualizará o peso sináptico cuja sinapse foi estimulada imediatamente antes de t_2 , que, pela disposição seqüencial dos tempos de atraso de propagação tende a ser a sinapse $s-1$. Em outros termos, o vetor de pesos sinápticos pode sofrer deslocamentos (*shift*) nas sinapses com a evolução do treinamento. De forma similar à antecipação do tempo de disparo, considera-se esta também uma característica inerente à rede.

3.2.3 Padrões presentes no vetor de pesos sinápticos

Diversos padrões podem ser observados no vetor de pesos sinápticos de um neurônio pulsado após a fase de treinamento. O número de pesos que atingiu a saturação em w_{max} e o espalhamento – entenda-se o número de sinapses subseqüentes em que o incremento dos pesos sinápticos é estimulado – são alguns deles. Alguns dos fatores que encontram-se diretamente relacionados com os padrões presentes no vetor de pesos após o processo de treinamento são:

- A generalização β : quanto maior a taxa de generalização, maior será o espalhamento dos pesos;
- A variância das instâncias do grupo para o qual o neurônio converge: quanto mais variam as instâncias do grupo, mais espalhados serão os pesos sinápticos do neurônio associado ao grupo; e
- A qualidade da convergência do neurônio: o espalhamento dos pesos tende a ser grande se um neurônio convergir para o centro de diversos grupos.

A figura 3.5 ilustra alguns desses casos.

Padrões presentes em um vetor de pesos sinápticos W de um neurônio pulsado j com 3 predecessores imediatos (i_1 , i_2 e i_3) e $d = 5$ conexões entre eles. a) padrão típico presente em W após o treinamento com uma taxa de generalização β pequena e com amostras com pequena variância. Os pesos sinápticos atingiram w_{max} apenas em uma conexão entre cada neurônio $i \in I$ e j ; b) padrão típico presente em W após o treinamento com uma taxa de generalização β mais alta. O incremento de um peso sináptico provocou o incremento nos pesos vizinhos, o que terá reflexos na generalização da rede; c) padrão típico presente em W após o treinamento com uma taxa de generalização β pequena e com amostras com variância acentuada nos tempos de emissão dos pulsos em i_1 . Apenas as conexões entre i_1 e j apresentaram espalhamento na distribuição dos pesos sinápticos; d) padrão típico presente em W após um processo de aprendizado incorreto. Dois pesos distintos foram estimulados nas conexões entre i_1 e j .

Figura 3.5: Padrão presente no vetor de pesos sinápticos de um neurônio após o aprendizado.

3.2.4 Sensibilidade ao peso máximo

Um dos pontos críticos para a convergência das RNP-RBFs é a escolha do valor máximo de peso sináptico w_{max} . Considere que os pesos sinápticos de um neurônio foram atualizados pelo algoritmo de treinamento após o recebimento de um padrão de entrada. Naturalmente, o potencial $\mu(t)$ será maior na próxima apresentação deste padrão à rede. Contudo, se um desses pesos for excessivamente alto, o neurônio pode vir a disparar sem que todos os pulsos dos i neurônios da camada de entrada tenham sido recebidos, o que traz complicações irreversíveis à convergência da rede. De forma análoga, valores muito pequenos de w_{max} podem, em casos extremos, impedir que o neurônio dispare.

Uma relação sugerida em Natschläger e Ruf (1998) para estimar o valor de

saturação dos pesos sinápticos é:

$$w_{max} = \frac{\vartheta}{n.g.\lambda}, \quad (3.18)$$

onde: w_{max} é o maior valor que pode ser assumido pelo peso sináptico; ϑ é o limiar de disparo do neurônio; $n.g$ é o número de neurônios de entrada da rede; λ é o comprimento do intervalo no qual a função de aprendizado $L(\Delta t)$ é maior que zero. Contudo, nenhuma metodologia foi apresentada para a determinação do parâmetro λ .

3.2.5 Auto-inibição

auto-
inibição

Dado que tipicamente no processo competitivo apenas o neurônio vencedor tem seus pesos atualizados pela regra de aprendizado, é oportuno adotar alguma sistemática para garantir que um único neurônio não vença demasiadas vezes o processo competitivo. Uma metodologia proposta por Natschläger e Ruf (1998) nesse sentido é a auto-inibição com relação ao aprendizado. A idéia básica é bastante simples: sempre que um neurônio vence muitas vezes um processo competitivo, ele deve se auto-inibir, ou seja, retirar-se do processo competitivo por algumas iterações de treinamento, possibilitando que outros neurônios sejam atualizados pela regra de aprendizado.

3.3 Aprendizado por reforço

Tradicionalmente, o aprendizado por reforço (WATKINS; DAYAN, 1992) (SUTTON; BARTO, 1998) se preocupa com a questão de “*como um agente autônomo que percebe e age em seu ambiente pode aprender a escolher ações ótimas para atingir seus objetivos*” (MITCHELL, 1997, p. 367). Em outros termos, deseja-se obter uma política de controle $\pi : E \rightarrow \mathcal{A}$, que escolhe uma ação a do conjunto de ações $\mathcal{A}$, dado o estado atual e do conjunto de estados E . Uma particularidade desta abordagem é que exemplos (e, a) não são fornecidos, e a única informação disponível é a recompensa ou punição.

Uma primeira abordagem – ainda introdutória – para a aplicação do aprendizado por reforço no treinamento de redes neurais pulsadas foi apresentado em (TAKITA; HAGIWARA, 2005) baseado em trabalhos prévios dos autores (TAKITA; HAGIWARA, 2002) (TAKITA; OSANA; HAGIWARA, 2000). Este trabalho – descrito nesta seção – apresenta uma rede capaz de processar entradas descritas em séries

temporais modelada para realizar um processo de decisão Markoviano parcialmente observável. O modelo demonstrou desempenho superior ao do bem conhecido algoritmo *Q-learning* (WATKINS; DAYAN, 1992) para o problema do pêndulo invertido (TAKITA; HAGIWARA, 2005).

3.3.1 Modelo de neurônio

Neste modelo, o neurônio incorpora algumas características do neurônio biológico já apresentadas em seções anteriores como a refração e o decremento do potencial. A figura 3.6 apresenta o comportamento desse neurônio.

Adaptado de (TAKITA; HAGIWARA, 2002).

Figura 3.6: Diagrama esquemático do neurônio pulsado para implementação do aprendizado por reforço.

O estado interno do neurônio pulsado é alterado por alguns fatores. Cada vez que um pulso de entrada proveniente de um neurônio pré-sináptico é recebido, o estado interno do neurônio é aumentado do valor do peso da conexão entre esses neurônios. Assume-se ainda o decaimento exponencial do valor do potencial com o tempo. Uma vez que o valor do estado interno ultrapassa um certo limiar, o neurônio dispara. Este disparo ocasiona três efeitos: *i*) pulsos de saída são enviados aos neurônios nas próximas camadas. Estas saídas são afetadas por um atraso no tempo; *ii*) o estado interno do neurônio que disparou é levado a zero; e *iii*) o efeito de refração é aplicado ao neurônio disparado.

3.3.2 Modelo de rede

A rede proposta consiste de quatro camadas: camada de entrada, primeira camada oculta ($\mathcal{O}1$), segunda camada oculta ($\mathcal{O}2$) e camada de saída. Não há conexões entre neurônios de um mesmo nível e nenhum neurônio existe nas ca-

madras ocultas inicialmente: eles serão gerados pelo processo de aprendizado. A camada $\mathcal{O}1$ é composta por neurônios com uma alta taxa de decremento do estado interno, enquanto que a camada $\mathcal{O}2$ é composta por neurônios com uma pequena constante de decremento do estado interno. A figura 3.7 apresenta o modelo da rede e as etapas do processo de aprendizado, descritas a seguir.

Adaptado de (TAKITA; HAGIWARA, 2002).

Figura 3.7: Aprendizado por reforço em uma rede neural pulgada.

Os neurônios pulsados formado as duas camadas ocultas – ou mais precisamente seus potenciais – fornecem uma pseudo-representação do **estado** do ambiente. Segundo Takita e Hagiwara (2005), este mecanismo evita o crescimento explosivo na capacidade de memória necessária, o que é um problema para modelos baseados em histórico temporal. Devido à configuração dos decrementos no estado interno, os neurônios em $\mathcal{O}1$ reconhecem o conjunto de entradas que dispararam simultaneamente como um estado. Os neurônios da camada $\mathcal{O}2$ reconhecem o conjunto de séries temporais de entrada como um estado.

3.3.3 Aprendizado

O sinal de **reforço** aplicado à rede é representado por um escalar. Um sinal positivo é dado quando o estado externo é o desejado, e um sinal negativo é fornecido caso contrário. O processo de aprendizado proposto para este modelo é dividido em cinco passos: 1) processo simplificado de geração da rede; 2) processo complexo de geração da rede; 3) processo de correção das probabilidades de saída; 4) processo correção de conexões inibitórias; e 5) processo interno de aprendizado do sinal de reforço.

O **processo simplificado de geração da rede** é o responsável por esta-

belecer uma conexão entre um conjunto de neurônios que dispararam em um dado instante na camada de entrada – com uma pequena margem de diferença de tempo – aos neurônios da camada de saída. Isto é feito através da criação de neurônios nas camadas ocultas $\mathcal{O}1$ e $\mathcal{O}2$. Em outros termos, este processo tem por objetivo formar um circuito capaz de gerar alguma saída para a entrada corrente. Neste processo, um neurônio de $\mathcal{O}2$ tem apenas uma conexão com $\mathcal{O}1$, agindo apenas como um retransmissor.

No **processo complexo de geração**, as conexões são construídas não apenas de forma a gerar uma saída para o conjunto de entradas correntes, mas também levando em consideração o efeito das entradas inicialmente desconsideradas.

Após a apresentação de algumas entradas, o neurônio de $\mathcal{O}2$ que contribuiu menos para o reforço positivo – denominado $n^{\mathcal{O}2}$ – é selecionado. É bastante provável que o neurônio que contribuiu menos com o sinal de reforço positivo (isto é, o neurônio cujo aprendizado não está evoluindo satisfatoriamente) represente um estado com percepção incompleta no problema parcialmente observável. Baseado nesta idéia, um novo elemento é construído para propagar a saída levando em conta informações não consideradas.

Desta forma, após um disparo de $n^{\mathcal{O}2}$, o neurônio em $\mathcal{O}1$ que disparou por último sem uma conexão com $n^{\mathcal{O}2}$ – denominado $n^{\mathcal{O}1}$ – é determinado. Caso não exista em $\mathcal{O}2$ algum neurônio cujas conexões sejam as mesmas de $n^{\mathcal{O}2}$ acrescidas da conexão com $n^{\mathcal{O}1}$, então um novo neurônio com essas características é criado em $\mathcal{O}2$, e seus pesos sinápticos atribuídos de forma que seu disparo ocorra somente neste caso.

Neurônios de $\mathcal{O}2$ são conectados a todos os neurônios da camada de saída. Contudo, um pulso é enviado a apenas um desses neurônios de saída quando um neurônio de $\mathcal{O}2$ dispara. O peso da conexão $w_{\mathcal{O}2-J}$ entre um neurônio de $\mathcal{O}2$ e um neurônio da camada de saída controla não a força do sinal transmitido, mas a probabilidade que o sinal seja transmitido sobre aquela conexão. No **processo de correção da probabilidade de saída**, $w_{\mathcal{O}2-J}$ é corrigido. Neste processo, quando um pulso é enviado de um neurônio de $\mathcal{O}2$ para um neurônio da camada de saída, o peso desta conexão é diminuído. Quando um sinal de reforço é aplicado após um certo período, o peso é aumentado se o sinal é positivo e é reduzido se o sinal é negativo. Por este processo, saídas que recebem um sinal de reforço positivo mais freqüentemente são aumentadas.

Na abordagem descrita até aqui, os neurônios em $\mathcal{O}1$ reconhecem entradas que disparam simultaneamente enquanto neurônios em $\mathcal{O}2$ reconhecem conjuntos

de neurônios da camada $\mathcal{O}1$ que disparam em uma série temporal. Em nenhum dos casos é desejável que neurônios de uma mesma camada disparem simultaneamente. O **processo de correção de conexões inibitórias** destina-se a detectar esses disparos indesejados, construir ou reforçar conexões inibitórias de forma que esses disparos sejam reprimidos.

Cada neurônio da camada $\mathcal{O}2$, quando dispara, gera um sinal de reforço interno que decai com o tempo. No **processo de correção do sinal de reforço interno**, a força deste sinal é aprendida. Quando esse sinal de reforço é gerado, os neurônios da camada $\mathcal{O}2$ que dispararam em um certo período passado são afetados, e a magnitude do sinal de reforço interno é aumentada. Quando o neurônio corrigido dispara novamente, o neurônio de $\mathcal{O}2$ que disparou anteriormente é afetado. Desta forma, o sinal de reforço propaga-se entre os neurônios da camada $\mathcal{O}2$. Estes elementos geram um sinal de reforço cujos resultados no aprendizado são similares ao do esquema de reforço convencional baseado na função de avaliação do estado (TAKITA; HAGIWARA, 2005).

3.4 Resumo

Este capítulo apresentou o estado da arte no que diz respeito à transposição dos três diferentes paradigmas de aprendizado (supervisionado, não-supervisionado e por reforço) para o universo das RNPs. O panorama geral apresentado neste capítulo deixa claro que a adaptação desses mecanismos de treinamento para o domínio temporal é extremamente recente, e que investigações mais aprofundadas são necessárias em todos os paradigmas e modelos de rede.

É importante salientar que ainda que as técnicas de aprendizado sejam em sua maioria dependentes dos modelos de rede e neurônio adotados, esforços foram realizados neste trabalho no sentido de abordá-las conjuntamente. Muito embora uma teoria de aprendizado de RNPs pareça ainda distante, busca-se, em suma, contribuir no sentido de que esta ótica unificada seja estimulada em futuros trabalhos.

Neste capítulo, particular atenção foi reservada ao aprendizado não supervisionado proposto por Natschläger e Ruf (1998), o primeiro algoritmo utilizado com sucesso no aprendizado em RNP-SRM-RBFs. A RNP-SRM-RBF utilizando campos receptivos gaussianos – apta para tratar dados reais – e com treinamento não-supervisionado será adotada no decorrer deste trabalho.

4 Aprendizado Baseado em Instâncias

“Um programa de computador é dito aprender a partir da experiência ϵ com respeito a algumas classes de tarefas $\mathcal{T}$ e medida de desempenho $\mathcal{D}$, se seu desempenho em tarefas de $\mathcal{T}$, como o medido por $\mathcal{D}$, melhora com a experiência ϵ .” (MITCHELL, 1997, p. 2).

AM O Aprendizado de Máquinas (AM) pode ser definido como a subárea da IA preocupada com programas que aprendem a partir de experiência (RUSSEL; NORVIG, 1995). Este campo de estudo tem procurado colecionar e organizar modelos de alguma forma associados a este tipo de aprendizado.

ABI Os métodos de aprendizado baseado em instâncias (ABI) não têm, em princípio, a preocupação de construir uma descrição geral e explícita da função-alvo. Eles preocupam-se inicialmente com a armazenagem dos exemplos de treinamento, e a generalização sobre esses exemplos é postergada até que uma nova instância precise ser classificada. Cada vez que uma nova instância é encontrada, sua relação com os exemplos previamente armazenados é examinada de forma a atribuir um valor da função-alvo à nova instância (MITCHELL, 1997).

Ripley (1996) classifica essa coleção de métodos como não-paramétricos, dado que os modelos usados são totalmente flexíveis, em contrapartida aos paramétricos, onde um pequeno número de parâmetros modela a função-alvo. Muito embora MLPs e redes RBF caracterizem-se por parametrizar a função-alvo, alguns autores os abordam neste contexto, dado que essas redes utilizam modelos com larga flexibilidade garantida por um grande número de parâmetros (RIPLEY, 1996).

Este capítulo procura reunir metodologias dessa natureza que possuem algum grau de semelhança com o processamento realizado pela RNP. Em outros termos, apresenta-se um conjunto de metodologias com as seguintes características:

Tarefa:	aprender através de instâncias
Funções-alvo:	$f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ou $f : \mathbb{R}^n \rightarrow v$
Experiência de treinamento:	instâncias x ou pares $(x, f(x))$
Representação da função-alvo:	qualquer
Algoritmo para aproximação:	qualquer

É preciso destacar que alguns dos métodos apresentados neste capítulo possuem raízes em uma ou mais teorias distintas (inteligência artificial, estatística, classificação de padrões, teoria de controle, etc.), e, portanto, certos conceitos aqui abordados apresentam pequenas variações na literatura dos diferentes campos.

A vertente não-supervisionada dos algoritmos do ABI, por exemplo, mantém estreita relação com metodologias da classificação de padrões. Neste contexto padrão correlato, um padrão pode ser definido como um par $\mathcal{P} = (\mathcal{B}, \omega)$, onde $\mathcal{B}$ é uma coleção de observações e ω é um conceito por detrás dessas observações. Nesses termos, a classificação de padrões diz respeito ao processo de inferir ω a partir de $\mathcal{B}$, ou, em outras palavras, é o processo de atribuir nomes ω para observações $\mathcal{B}$ (SCHÜRMANN, 1996).

Um problema associado ao da classificação de padrões é do agrupamento, que pode ser definido como o processo de determinar subclasses automaticamente a partir do conjunto de aprendizado $\{\mathcal{B}, k\}$ para um único número de classes k (SCHÜRMANN, 1996). Ripley (1996) descreve métodos dessa natureza como aqueles usados para dividir um conjunto de n observações em k grupos, de forma que membros de um mesmo grupo sejam mais parecidos que membros de grupos diferentes.

Considerando a taxonomia desses métodos, Halkidi, Batistakis e Varzirgiannis (2001) propõem dividir os algoritmos utilizados na tarefa do agrupamento segundo a estratégia de análise das instâncias em *i*) baseados em particionamento: objetivam decompor diretamente o conjunto de dados em um conjunto de grupos disjuntos; *ii*) hierárquicos: realizam sucessivamente o processo de agrupar pequenos grupos e dividir grupos grandes; *iii*) baseados em densidade: objetivam agrupar dados vizinhos em um grupo de dados baseados nas suas condições, como a variância desses dados; *iv*) baseados em grade: quantificam o espaço em um número finito de células e realizam operações sobre elas; *v*) baseados em abordagem nebulosa: algoritmos onde um dado mantém certo grau de pertinência a diversos grupos.

Este capítulo encontra-se organizado da seguinte maneira: a seção 4.1 descreve o algoritmo k-vizinho mais próximo *k-nearest neighbour* (KNN), que fornece

as bases para muitas das metodologias descritas a seguir. A seção 4.2 apresenta o *k-means*. A seção 4.3 destaca o aprendizado da quantificação vetorial, uma variação adaptativa do *k-means*. A seção 4.4 apresenta outra variação do *k-means* com organização topológica denominada mapas auto-organizados. Na seção 4.5 apresenta-se as redes RBF. A seção 4.6 destaca algoritmos para classificação de padrões com abordagem conhecida como hierárquica. A seção 4.7 fornece uma visão geral das abordagens utilizadas na avaliação de agrupamentos de dados, e que podem ser utilizadas como medida de desempenho $\mathcal{D}$.

4.1 k-vizinho mais próximo

Segundo Mitchell (1997), o K-vizinho mais próximo – *K-nearest neighbor* (KNN) (FIX; HODGES, 1951) é o mais simples algoritmo de aprendizado baseado em instâncias. Assume-se que todas as instâncias correspondem a pontos no espaço $\mathbb{R}^n$. A distância entre as instâncias é usualmente a distância Euclidiana, que para duas instâncias xi e xj é dada por:

$$d(xi, xj) = \sqrt{\sum_{r=1}^n (xi_r - xj_r)^2} \quad (4.1)$$

onde: d é a distância Euclidiana; xi_r é o r -ésimo atributo da i -ésima instância de entrada $x \in X$; n é o número de dimensões de entrada.

O algoritmo de treinamento para uma rede que adote este método de forma supervisionada consiste simplesmente em armazenar todos os exemplos de treinamento $(x, f(x))$ em uma lista **exemplos**. Quando uma nova instância de entrada xq é apresentada para classificação, o KNN retorna como $\hat{f}(xq)$ (valor estimado de $f(xq)$) o valor mais comum de f dentre os k exemplos de treinamento mais próximos de xq . A classificação segundo o KNN é apresentada no algoritmo 4.1.

Algoritmo 4.1 Classificação segundo o *k-vizinho mais próximo* básico

- 1: **repetir:**
 - 2: **para todo** instância de entrada xq **fazer:**
 - 3: Sejam $x1, \dots, xk$ as k instâncias em **exemplos** que são mais próximas de xq segundo o critério de distância d , e com $f(x) \rightarrow \mathcal{V}$, e onde $\mathcal{V} = \{v_1, \dots, v_m\}$, $m \leq k$
 - 4: **retornar**

$$\hat{f}(xq) \leftarrow \underset{v \in \mathcal{V}}{\operatorname{argmax}} \sum_{i=1}^k \varphi(v, f(xi))$$
 onde $\varphi(a, b) = 1$ para $a=b$, e $\varphi(a, b) = 0$ caso contrário.
 - 5: **fim**
 - 6: **até** fim das instâncias
-

Um exemplo da superfície de decisão em $\mathbb{R}^2$ tipicamente implementada por algoritmos dessa natureza é mostrado na figura 4.1. Os polígonos envolvendo cada centro indicam as regiões do espaço onde as instâncias atendem ao critério de menor distância para aquele centro. Para $k=1$, esta superfície de decisão tipicamente é um poliedro convexo circundando cada um dos exemplos de treinamento (MITCHELL, 1997). Esse tipo de diagrama também é conhecido como diagrama de Voronoi.

Extraído de (MITCHELL, 1997).

Figura 4.1: A superfície de decisão típica do algoritmo KNN.

Trata-se então de uma metodologia que efetua um mapeamento $f : \mathbb{R}^n \rightarrow \mathcal{V}$, com $\mathcal{V} = \{v_1, \dots, v_m\}$, mas que também pode ser adaptada para aproximar funções contínuas $f : \mathbb{R}^n \rightarrow \mathbb{R}$ retornando a média dos valores dos k vizinhos mais próximos ao invés do valor mais comum (MITCHELL, 1997).

KNN ponderado. Um refinamento do algoritmo KNN básico é pesar as contribuições de cada um dos k vizinhos de acordo com a sua distância para com a instância xq . Uma visão geral dessa técnica é apresentada, por exemplo, por Batista (2003).

4.2 *k-means*

Um algoritmo que possui estreitas relações com o KNN, porém com raízes na classificação de padrões, é o *k-means* (FORGY, 1965) (JANCEY, 1966) (MACQUEEN, 1967). Este algoritmo também é conhecido como *c-means*, apesar de alguns autores considerarem tratar-se de termos usados para definir algoritmos específicos (RIPLEY, 1996).

Objetiva-se particionar um espaço $\mathbb{R}^n$ em k grupos atribuindo-se um centro w_k a cada grupo, ou, em outros termos, objetiva-se escolher o centro dos grupos

de forma a minimizar o erro:

$$E = \sum_{x| x \in X \text{ e } x \in S_i} ||X - w_i||^2, \quad (4.2)$$

onde: X é uma coleção de pontos em um espaço n -dimensional $\mathbb{R}^n$; k é o número de grupos nos quais os pontos devem ser particionados; S_i é o conjunto dos dados pertencentes ao grupo i ; w_i é o centro dos elementos do grupo i .

Este algoritmo básico pode ser encontrado com uma enorme gama de variações na literatura das áreas de classificação de padrões, inteligência artificial e aprendizado de máquinas.

Segundo Ripley (1996), todos os algoritmos dessa classe iniciam com alguma metodologia para dividir os exemplos em k grupos ou em um conjunto de k centros de grupos. Forgy (1965) propõe uma sistemática onde todos os exemplos são atribuídos simultaneamente ao seu centro de grupo mais próximo, e, então, cada centro de grupo move-se para a média do grupo e este processo é repetido. O algoritmo proposto por MacQueen (1967) propõe que cada exemplo seja considerado em uma iteração, e que os centros de grupo sejam atualizados quando um exemplo é atribuído ao grupo. Hall e Ball (1965) apresentam um algoritmo similar, onde os grupos dividem-se e agrupam-se (k varia dinamicamente).

Saphiro (2005) propõe uma sistemática para o algoritmo *k-means* que será tomada como referência neste trabalho. Ela é apresentada no algoritmo 4.2.

Algoritmo 4.2 *k-means* básico

- 1: Inicializar conjunto de centros. Para cada grupo $i=1, \dots, k$ atribuir um vetor centro w_i com o mesmo número de dimensões que os dados de treinamento. Tipicamente os centros iniciais são observações aleatoriamente escolhidas dos dados de treinamento.
 - 2: **repetir:**
 - 3: **para todo** dado x **fazer:**
 - 4: Associar x com o centro mais próximo, de forma que:
 $S_i = \{x | x \in X \text{ e } ||x - w_i||^2 < ||x - w_j||^2 \text{ para todo } j \neq i\}$
 - 5: Computar a média dos dados do grupo, e adotar como centro do grupo o vetor médio, fazendo:
 $w_i \leftarrow \frac{1}{|S_i|} \cdot \sum_{x \in S_i} x$ para todo i
 - 6: **fim**
 - 7: **até** estabilização dos centros
-

Algumas das variações mais comuns desse algoritmo serão brevemente descritas a seguir.

k-medoids. Partindo-se de k centros, tipicamente considera-se a possibilidade de trocar um centro com uma instância, e seleciona-se a troca mais vantajosa (RIPLEY, 1996). Repete-se este processo até a convergência. Nesta variação do *k-means* – também denominada *k-median* (VINOD, 1969) – insiste-se então que o centro dos grupos sejam exemplos de treinamento (*medoids*). Este algoritmo apresenta maior aplicabilidade quando o espaço de atributos não é contínuo, e assegura-se que ele encontre o mínimo global apenas para $k=2$. Em geral encontra-se um mínimo local (RIPLEY, 1996).

k-means adaptativo. Segundo Ripley (1996), evoluções do algoritmo básico para torná-lo adaptativo levaram ao desenvolvimento de uma larga família de algoritmos usualmente classificados como teoria de ressonância adaptativa (*Adaptive Resonance Theory - ART*) (CARPENTER; GROSSBERG, 1987a) (CARPENTER; GROSSBERG, 1987b) (CARPENTER; GROSSBERG, 1990).

Dentre as características apresentadas pelo *k-means* destacam-se:

- Alto custo para classificar novas instâncias, dado que quase toda a computação é realizada em tempo de classificação (MITCHELL, 1997);
- Sensibilidade à escolha inicial dos centros e à ordem de apresentação dos exemplos de treinamento;
- A estabilização dos centros é assegurada (SAPHIRO, 2005), mas de uma forma geral métodos de agrupamento não têm convergência assegurada para um ótimo global (RIPLEY, 1996); e
- O número de classes k precisa estar adequadamente estabelecido.

De forma a exemplificar alguns desses problemas, considere uma possível escolha de valores iniciais para o centro de 4 grupos em $\mathbb{R}^2$ como o mostrado na figura 4.2a. A figura 4.2b mostra uma evolução bem sucedida dos centros dos grupos para um ponto de mínimo global a partir dos valores iniciais mostrados na figura 4.2a. A figura 4.2c apresenta outra possível escolha inicial de valores para os centros dos 4 grupos. A figura 4.2d apresenta uma classificação sub-ótima, isto é, a estabilização dos centros em um ponto de mínimo local da superfície de erro.

A figura 4.3 ilustra a convergência dos centros para valores sub-ótimos devido à escolha inadequada do número de grupos k . A figura 4.3a apresenta um exemplo de convergência com um número excessivo de classes, e a figura 4.3b apresenta a convergência com um número insuficiente de classes.

Extraído de (SAPHIRO, 2005).

Figura 4.2: Evolução típica dos centros dos grupos para o algoritmo *k-means*.

Extraído de (SAPHIRO, 2005).

Figura 4.3: Evolução típica dos centros dos grupos para o algoritmo *k-means* com um número inadequado de classes.

4.3 Aprendizado da Quantificação Vetorial

A quantificação vetorial é um método clássico do processamento de sinais utilizado na aproximação da distribuição de uma única classe em um *codebook*, isto é, um vetor típico da distribuição. Cada instância é mapeada ao vetor típico mais

próximo, e a proximidade é avaliada utilizando-se alguma medida de distância entre a instância e o *codebook*. Tomando-se a distância quadrática do sinal do vetor ao *codebook* mais próximo teremos o algoritmo *k-means* clássico (RIPLEY, 1996).

Uma versão adaptativa para este algoritmo (KOHONEN, 1986) (KOHONEN, 1988) (KOHONEN; BARNA; CHRISLEY, 1988) – usualmente denominada Aprendizado da Quantificação Vetorial – *Learning Vector Quantization* (LVQ) apresenta uma sistemática onde o *codebook* w_k mais próximo da instância x é atualizado de acordo com a distância d para a instância x apresentada na forma:

$$w_k \leftarrow \begin{cases} w_k + \alpha(t)[x - w_k] & \text{se } x \text{ foi corretamente classificado;} \\ w_k - \alpha(t)[x - w_k] & \text{caso contrário.} \end{cases} \quad (4.3)$$

onde: d é a distância entre a instância e o vetor típico, com $d(x, w_k) < d(x, w_i)$ para qualquer classe i ; $\alpha(t)$ é um parâmetro inicialmente escolhido menor do que 0.1 e é reduzido até zero durante o número escolhido de iterações. Todos os demais *codebooks* w_i não sofrem alteração.

O conjunto inicial de *codebooks* é adotado através de análise do conjunto de treinamento: é largamente desejável que os vetores iniciais estejam posicionados dentro dos limites de suas classes (RIPLEY, 1996).

Segundo Cabral (1999), o LVQ difere dos métodos baseados em vizinho mais próximo na maneira como os representantes *codebooks* das classes são determinados. Enquanto que no *k-nearest-neighbour* estes são determinados estatisticamente, no LVQ estes são determinados pelo processo de aprendizado da rede.

4.4 SOM

O SOM – *Self-Organized Maps* – Mapas Auto-Organizados (KOHONEN, 1982b) (KOHONEN, 1982a) é um algoritmo similar ao *k-means* onde tenta-se atribuir alguma estrutura aos vetores representativos das classes. Este processo é conhecido como organização topológica. Existem três processos essenciais envolvidos na formação dos mapas auto-organizados:

- Competição: para cada padrão de entrada, os neurônios da rede computam seus respectivos valores de uma função discriminante, e o neurônio com o maior valor desta função é declarado o vencedor da competição;

- Cooperação: o neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados;
- Adaptação: habilita os neurônios excitados a atualizarem seus valores individuais da função discriminante em relação à entrada através de ajustes em seus pesos sinápticos. Os ajustes são feitos de forma que a resposta do vencedor neurônio à aplicação subsequente de uma entrada similar seja aumentada.

O vetor de pesos sinápticos w dos neurônios tem o mesmo número de dimensões que os dados de entrada. Uma forma usual de comparar o vetor de entrada x com o vetor de pesos w é computar $w_j^T \cdot x$ para todos os neurônios j , e adotar o maior. Em outros termos, o vencedor é dado pela relação (HAYKIN, 1999):

$$v(x) = \arg \min_j \|x - w_j\| \quad (4.4)$$

onde: v é o neurônio vencedor do processo competitivo; x é o vetor de entrada; w_j é o peso sináptico do j -ésimo neurônio. A vizinhança topológica é usualmente adotada na forma de uma gaussiana expressa por (HAYKIN, 1999):

$$h_{j,v(x)} = \exp\left(-\frac{d_{j,v}^2}{2\sigma^2}\right) \quad (4.5)$$

onde: h é a função que define a vizinhança topológica; d é a distância entre os neurônios; e σ é uma função que tipicamente decresce com o tempo. Uma formulação usual para σ é (HAYKIN, 1999):

$$\sigma(k) = \sigma_0 \cdot \exp\left(-\frac{k}{\tau}\right) \quad (4.6)$$

onde: σ_0 é o valor de σ no início do processo de aprendizado; τ é uma constante de tempo; k é a iteração. Finalmente, uma expressão usual para a atualização dos pesos sinápticos é (KOHONEN, 1982b):

$$w_j(k+1) = w_j(k) + \eta(k) \cdot h_{j,v(x)}(k) \cdot (x - w_j(k)) \quad (4.7)$$

Desta forma, a aplicação sucessiva do algoritmo tipicamente conduz a uma organização topológica, dado que neurônios que são adjacentes em um reticulado tenderão a ter vetores de pesos sinápticos similares.

4.5 Redes de base radial

Até a seção anterior foram abordados métodos típicos de aprendizado baseado em instâncias e não-paramétricos, onde não existe um modelo definido para a função-alvo. Alguns modelos paramétricos (onde a função-alvo é descrita através de parâmetros) são também usualmente abordados dentro do contexto dos métodos de aprendizado baseado em instâncias (MITCHELL, 1997). Dada sua similaridade para com a RNP-SRM-RBF, as redes de base radial serão brevemente descritas nesta seção.

Teorema
de Cover

A idéia básica para a classificação de padrões utilizando RBFs é a de transformar o problema para um espaço de muitas dimensões de uma forma não-linear (HAYKIN, 1999). A justificativa para tal é conhecida como Teorema de Cover. Em síntese, um problema complexo de classificação de padrões tem uma maior probabilidade de ser linearmente separável quando em um espaço de muitas dimensões (COVER, 1965).

A arquitetura de uma rede com RBF é apresentada na figura 4.4, onde a primeira camada computa o valor de várias funções *kernel* e a camada de saída realiza a combinação linear dos valores da primeira camada.

Figura 4.4: Arquitetura de uma rede com função de base radial.

Dado um conjunto de exemplos de treinamento da função-alvo, as redes RBF são tipicamente treinadas em dois estágios. Inicialmente, o número k de unidades é determinado e cada unidade i é definida escolhendo-se os valores de x_i e σ_i^2 que definem o *kernel* da função. Na segunda etapa, as funções *kernel* são mantidas fixas, e os pesos w_i são treinados de forma a maximizar a cobertura dos dados de entrada pela rede, utilizando um critério de erro global (MITCHELL, 1997) (KOVÁCS, 1996). É comum escolher o *kernel* como uma gaussiana centrada no ponto c_i e com uma variância σ_i^2 .

Segundo Mitchell (1997), duas abordagens são usuais para a escolha do nú-

mero de gaussianas da rede e seus valores iniciais. Uma primeira abordagem é centrar uma função *kernel* (e, portanto, uma gaussiana) em cada exemplo de treinamento x_i e fazer todas as gaussianas com a mesma variância σ^2 . Nesta abordagem a rede RBF cobre os dados de treinamento exatamente, mas trata-se de uma metodologia pouco eficiente para grandes conjuntos de treinamento. Uma segunda abordagem é escolher um conjunto de *kernels* menor do que o número de exemplos de treinamento. Neste caso, as condições iniciais da rede dependem de análises prévias do conjunto de treinamento, e algumas das principais alternativas são: *i*) distribuir os *kernels* com centros espaçados uniformemente através do espaço de instâncias X ; *ii*) distribuir os centros de forma não uniforme, de acordo com a incidência de instâncias (por exemplo adotando centros escolhidos aleatoriamente em subconjuntos das instâncias de treinamento); e *iii*) identificar candidatos a grupos e adicionar uma função *kernel* centrada em cada grupo. Já Ripley (1996) cogita a utilização do algoritmo *k-means* para a seleção dos centros das gaussianas.

Nesta abordagem, a função-alvo é parametrizada na forma:

$$\hat{f}(x) = w_0 + \sum_{i=1}^k w_i \cdot \kappa_i(d(c_i, x)), \quad (4.8)$$

onde: k é o número de *kernels*; x é uma instância de X ; w_i é o i -ésimo peso sináptico associado ao i -ésimo *kernel*; κ_i é a função *kernel* definida de forma a decrescer com o aumento da distância $d(c_i, x)$; c_i é o centro do *kernel* i . A contribuição de cada um dos termos $\kappa_i(d(c_i, x))$ é restrita a uma região próxima ao ponto c_i . A função $\kappa_i(d(c_i, x))$ é usualmente dada por:

$$\kappa_i(d(c_i, x)) = e^{\frac{1}{2 \cdot \sigma_i^2} \cdot d^2(c_i, x)} \quad (4.9)$$

Uma rede de base radial com o *kernel* acima é sabidamente capaz de aproximar qualquer função arbitrariamente bem se o número de gaussianas k for suficientemente grande (MITCHELL, 1997).

4.6 Algoritmos hierárquicos

No agrupamento hierárquico, tipicamente um dado não é particionado em um grupo específico em uma única passagem. Ao contrário, as partições ocorrem sucessivamente uma a uma, gerando uma estrutura que registra hierarquicamente

os grupos encontrados. Duas abordagens tradicionalmente empregadas para realizar esta tarefa são: *i*) métodos aglomerativos, que visam basicamente realizar uma série de fusões dos n exemplos em grupos e *ii*) métodos divisivos, que objetivam separar n exemplos sucessivamente refinando seu grupo. Tipicamente esta metodologia faz uso de um diagrama bidimensional denominado *dendrograma* que ilustra as fusões ou divisões que devem ser realizadas no estágio de análise.

Uma metodologia hierárquica aglomerativa segundo Moore (2001) é apresentada no algoritmo 4.3. Uma metodologia hierárquica divisiva pode ser expressa pelo algoritmo apresentado em 4.4. Algoritmos representativos dessas classes podem ser respectivamente encontrados em (ZHANG; RAMAKRISHNMAN; LINVY, 1996) e (GUHA; RASTOGI; SHIM, 1998).

Algoritmo 4.3 Agrupamento hierárquico aglomerativo básico

- 1: Inicialização: tomar cada ponto do conjunto de dados de entrada como um novo grupo
 - 2: **repetir:**
 - 3: Utilizando um critério de distância, determinar os dois grupos mais próximos dentre todos os grupos
 - 4: Agrupar os dois grupos mais próximos em um único novo grupo
 - 5: **até** que só reste um grupo
-

Algoritmo 4.4 Agrupamento hierárquico divisivo básico

- 1: Inicialização: tomar todos os pontos como pertencentes a um único grupo
 - 2: **repetir:**
 - 3: Utilizando um critério de distância, determinar os dois pontos mais distantes em um grupo, dentre todos os grupos
 - 4: Dividir o grupo com os dois elementos mais distantes criando dois novos grupos
 - 5: **até** que todo ponto seja um grupo
-

4.7 Medidas de desempenho

Algoritmos que objetivam realizar uma escolha automática do número de classes tipicamente fazem necessária alguma função que avalie a qualidade do agrupamento realizado. De fato, uma importante questão na análise de agrupamentos é a avaliação do resultado deste agrupamento de forma a encontrar o particionamento mais adequado aos dados (HALKIDI; BATISTAKIS; VARZIRGIANIS, 2001). Este problema é também denominado validade do agrupamento.

Segundo Berry e Linoff (1996), existem dois critérios fundamentais para avaliar um agrupamento:

- i. **Espalhamento:** membros de cada grupo devem ser tão próximos uns dos

outros quanto possível. Uma medida comum nesse âmbito é a *variância* dos dados. A divergência de Kullback-Leibler (KULLBACK, 1959) e a informação mútua (CHURCH, 1991) são outras medidas importantes. O espalhamento tipicamente deve ser minimizado;

- separação ii. **Separação:** dois grupos devem ser tão distantes entre si quanto possível. Três medidas usuais nesse contexto são a *distância entre os membros mais próximos* do grupo, a medida de *distância entre os membros mais distantes* do grupo e a *distância entre os centros* dos grupos (HALKIDI; BATISTAKIS; VARZIRGIANNIS, 2001). A separação tipicamente deve ser maximizada.

Baseado nesses conceitos fundamentais, é possível utilizar ao menos três abordagens para investigar a validade de grupos (THEODORIDIS; KOUTROUBAS, 1999). Estas são apresentadas a seguir.

Matriz de confusão **Critério externo:** faz-se uso de informações fornecidas externamente sobre as classes para avaliar o desempenho das partições realizadas pelo algoritmo. Avalia-se então a similaridade (ou diferença) entre as classes. Um critério típico é a matriz de confusão (SWETS, 1988). Esta matriz apresenta as classes atribuídas pelo classificador a pontos de cada uma das classes segundo um gabarito. Desta forma, n_{12} representa o número de pontos da classe 1 (segundo o gabarito) classificados como classe 2 (pelo classificador), e assim sucessivamente.

Tabela 4.1: Matriz de confusão para várias classes.

Classes reais (gabarito)	Classes atribuídas pelo classificador			
	Classe 1	Classe 2	...	Classe N
Classe 1	n_{11}	n_{12}	...	n_{1N}
Classe 2	n_{21}	n_{22}	...	n_{2N}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Classe N	n_{N1}	n_{N2}	...	n_{NN}

A partir desta tabela, é possível definir os valores apresentados na equação 4.10, e, fazendo uso desses valores é possível definir para cada classe as contagens de instâncias apresentadas na equação 4.11, onde: TP_i é a taxa de verdadeiros positivos da classe i ; FP_i é a taxa de falsos positivos da classe i ; TN_i é a taxa de verdadeiros negativos da classe i ; FN_i é a taxa de falsos negativos da classe i .

$$A = \sum_{\substack{a=1..N \\ a \neq i}} \sum_{\substack{b=1..N \\ b \neq i}} n_{ab}, \quad B = \sum_{\substack{a=1..N \\ a \neq i}} n_{ai}, \quad C = \sum_{\substack{a=1..N \\ a \neq i}} n_{ia}, \quad D = n_{ii} \quad (4.10)$$

$$TP_i = \frac{D}{C + D}, \quad FP_i = \frac{B}{A + B}, \quad TN_i = \frac{A}{A + B}, \quad FN_i = \frac{C}{C + D} \quad (4.11)$$

Uma outra forma de expressar os resultados obtidos com a sistemática da matriz de confusão é através do Característica de Operação do Receptor – *Receiver ROC Operating Characteristic* (ROC) (SWETS, 1988), que consiste em apresentar os dados em termos do par especificidade e sensibilidade, estas definidas por:

$$sens = \frac{TP}{TP + FN}, \quad espec = \frac{TN}{TN + FP} \quad (4.12)$$

Uma vez que o classificador perfeito está no ponto (1,1), a distância euclidiana ao classificador perfeito (DCP) é dada por:

$$DCP = \sqrt{(1 - sens)^2 + (1 - espec)^2} \quad (4.13)$$

Critério interno: objetiva-se fazer a avaliação do desempenho do agrupamento usando apenas valores e atributos inerentes ao conjunto de dados. Existem dois casos nos quais aplica-se critérios internos de validade de agrupamentos dependendo da estrutura do agrupamento: para agrupamentos hierárquicos, por exemplo, a medida típica é a matriz cofenética (HALKIDI; BATISTAKIS; VARZIR-GIANNIS, 2001)), e para agrupamentos isolados – cujo índice usual é a estatística Γ de Hubert (ou estatística Γ normalizada). Neste caso, conforme o apresentado em Tseng e Kao (2004), sejam X e Y duas matrizes de dimensões $n \times m$, onde $X(i,j)$ indica a similaridade entre os dados i e j , e $Y(i,j)$ é definido como:

$$Y(i,j) = \begin{cases} 1 & \text{caso } i \text{ e } j \text{ estejam agrupados na mesma classe;} \\ 0 & \text{caso contrário.} \end{cases} \quad (4.14)$$

A estatística de Hubert representa a correlação serial entre as matrizes X e Y , e é definida como:

$$\Gamma = \frac{1}{M} \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(\frac{X(i,j) - \bar{X}}{\sigma_x} \right) \cdot \left(\frac{Y(i,j) - \bar{Y}}{\sigma_y} \right), \quad (4.15)$$

onde: Γ é a estatística Γ de Hubert, que varia no intervalo $[-1,1]$; $M = n \cdot (n-1)/2$; X é a matriz de similaridade; $\bar{X}$ é a média de X ; σ_x é o desvio padrão de X .

Critério relativo: uma outra alternativa é avaliar um agrupamento através da comparação com outros agrupamentos. Em uma abordagem não supervisionada tipicamente executa-se o algoritmo para uma larga variedade de parâmetros (por exemplo, diversas condições iniciais) e então observa-se a maior variação desses parâmetros para a qual o número de grupos permanece constante. Escolhe-se então como valores apropriados dos parâmetros aqueles que correspondem ao centro desta variação (HALKIDI; BATISTAKIS; VARZIRGIANNIS, 2001).

4.8 Resumo

Redes neurais pulsadas de base radial treinadas de forma não supervisionada como apresentado nos capítulos anteriores – ou mas especificamente seu processo competitivo – mantêm uma estreita relação com os algoritmos de aprendizado baseado em instâncias. Este capítulo espera ter apresentado de forma introdutória algumas das metodologias para agrupamento que mais se assemelham à natureza do agrupamento realizado por essas redes.

Desta forma – após a revisão dos algoritmos fundamentais para o aprendizado – espera-se estabelecer relações que possam nortear os desenvolvimentos com as redes pulsadas e seus algoritmos de treinamento. Algumas propostas nesse sentido serão apresentadas a seguir.

Parte II

Aprendizado Proposto

5 Revisitando a RNP-SRM-RBF

“A lealdade a uma opinião petrificada nunca quebrou uma corrente ou libertou uma alma humana” (Mark Tawain).

Nos capítulos anteriores, RNPs foram apresentadas. Elas são uma nova abstração para neurônios biológicos emergente da *neurociência* que tem como característica principal a codificação temporal da informação. Muitos de seus algoritmos de aprendizado possuem estreita relação com os algoritmos clássicos para o aprendizado baseado em instâncias, abordados no capítulo 4.

Este capítulo busca, em um primeiro momento, estabelecer as principais similaridades entre a RNP-SRM-RBF e os modelos clássicos de aprendizado através de estudos preliminares conceituais e práticos com a rede. À luz deste novo prisma, busca-se formas para tratar algumas das perguntas que permanecem abertas nesta abordagem, tais como: os modelos de rede de Natschläger e Ruf (1998) e Hopfield (1995) são equivalentes no que diz respeito à forma de processar os dados de entrada? A inserção dos campos gaussianos no modelo de Natschläger e Ruf (1998) tem alguma influência sobre a forma de processamento da RNP? As RNPs – particularmente a RNP-SRM-RBF com campos receptivos – apresentam alguma nova perspectiva no processo de convergência?

Através da discussão de possíveis caminhos para responder a estas perguntas, busca-se identificar os possíveis novos horizontes abertos por esta abordagem, determinando direções para o aprimoramento e extensão do modelo de rede e seu aprendizado.

O capítulo encontra-se organizado da seguinte forma: a seção 5.1 busca posicionar a RNP no contexto do aprendizado baseado em instâncias. A seção 5.2 revisita a codificação temporal proposta por Bohte (2003) e estabelece algumas de suas implicações. A seção 5.3 traça um comparativo entre as redes propostas por Hopfield (1995) e Natschläger e Ruf (1998), enfatizando suas similaridades e diferenças. A seção 5.4 apresenta as vantagens e desvantagens do paradigma pulsado no processo de convergência no aprendizado baseado em instâncias. Finalmente,

a seção 5.5 apresenta um dos principais diferenciais da rede adotada no processo de convergência: a possibilidade de armazenar diversos centros simultaneamente.

Durante a leitura deste capítulo recomenda-se a freqüente consulta à figura 2.16, que apresenta a nomenclatura completa para o modelo da RNP adotada.

5.1 Posicionamento entre as técnicas

O processamento básico da RNP-SRM-RBF de Natschläger e Ruf (1998) e Bohte (2003) utilizando o aprendizado não-supervisionado pode ser resumido da seguinte forma: uma instância composta por n variáveis – contínuas ou discretas – é apresentada à rede. A instância é transformada para um novo conjunto de $n.g$ variáveis – contínuas ou discretas. Cada uma das $n.g$ variáveis é tomada como o tempo de disparo de um neurônio da camada de entrada. Os neurônios da camada de entrada disparam, e o neurônio da camada de saída que disparar primeiro tem seus parâmetros livres atualizados pelo algoritmo de aprendizado para selecionar atrasos de forma que os pulsos dos neurônios da camada de entrada sejam computados simultaneamente no potencial do neurônio da camada de saída.

Como já destacado em seções anteriores, a rede tem forte semelhança com redes RBF clássicas, e o aprendizado competitivo entre os neurônios da camada de saída para sintonia dos centros mantém uma estreita relação com o algoritmo *k-means*. Estabelecendo uma analogia com abordagens clássicas, tem-se:

- i. Os dados são transformados para uma série de pulsos temporais através de um **pré-processamento** proposto por Bohte (2003) cujas implicações no aprendizado não são claras;
- ii. O critério adotado para avaliar a separação entre a instância e um referencial difere fortemente dos critérios usuais como a *distância Euclidiana* ou a *distância de Mahalanobis* (DUDA; HART, 1973). Baseado em fundamentos biológicos, Natschläger e Ruf (1998) introduziram em seu modelo o uso do *tempo de disparo* do neurônio de saída t_j como **medida de distância**. Em síntese, quanto mais rápido um neurônio dispara a um padrão de entrada, mais certo é que o padrão pertence à classe em questão. Na RNP, então, cada neurônio responde pelo próprio critério de distância, que é dado por uma complexa dinâmica no tempo com $t_j = f(T_i, W, \epsilon, \mu, t)$, onde: T_i é o conjunto de tempos de disparo do neurônio pré-sináptico i ; W é o conjunto de pesos sinápticos do neurônio j ; ϵ é a função que modela a influência dos

pulsos recebidos pela k -ésima sinapse entre os neurônios i e j no potencial do neurônio j ; μ é o potencial do neurônio; t é o tempo.

- iii. O **aprendizado** destina-se a selecionar o centro dos neurônios RBF e faz uso da metodologia *winner-takes-all* (o vencedor leva tudo), também empregada em métodos como o *k-means* e LVQ. Esse processo traz à rede ao menos um problema de convergência tradicionalmente associado a esses métodos: a convergência em mínimos locais.

Estes processos serão discutidos em maior profundidade nas seções a seguir.

5.2 Codificação temporal de Bohte

Os campos receptivos gaussianos (BOHTE, 2003) que codificam um valor de entrada em um conjunto de funções impulso disparados em diferentes tempos, são uma contribuição bastante nova às RNPs, e, por isso, suas implicações no processamento da rede não se encontram bem estabelecidas na literatura. No contexto da RNP-SRM-RBF, é possível identificar ao menos três características primordiais dos campos receptivos:

- i. Garantem uma transformação do tipo $f : \mathbb{R}^n \rightarrow \mathcal{V}$, com $\mathcal{V} = \{v_1, \dots, v_{n.g}\}$, onde $v \in \mathbb{R}_+$. Como adota-se $T_i = \mathcal{V}$, esta codificação implica que cada neurônio de entrada i disparará uma única vez (uma condição necessária para o trabalho com a RNP-SRM-RBFs);
- ii. Influenciam na separação dos dados de entrada, já que o agrupamento desses dados sofre alteração após a sua transformação para um novo espaço;
- iii. Codificam os dados de forma que a somatória das latências dos pulsos gerados pela codificação em cada uma das n dimensões de entrada $v_1 + v_2 + \dots + v_g$, tenha ponto de mínimo e de máximo.

Separação dos dados. A transformação é evidentemente dependente da posição dos centros c_k e do desvio padrão σ_k das gaussianas, mas em geral exibe uma tendência de espaçar pontos que estejam próximos do cruzamento de duas gaussianas subseqüentes, tornando-os melhor separáveis para o classificador. Exemplos em $\mathbb{R}^3$ que fornecem uma idéia da separação dos pontos nesse novo espaço são apresentados na figura 5.1.

Esq. Três gaussianas g_1, g_2, g_3 (em azul) de um campo receptivo com centros gerados segundo a equação 2.20. Os valores de entrada equidistantes 0-8 (em vermelho) foram apresentados aos campos receptivos para transformação e os valores de latência na geração dos pulsos foram anotados. **Dir.** O sistema transformado, onde cada eixo representa a latência em uma das gaussianas g_1, g_2 e g_3 . Os pontos em preto representam as projeções dos pontos nos planos laterais. As gaussianas têm desvio padrão: a) 2; b) 1; c) 0.5; d) 0.3. A transformação exibe uma tendência a espaçar valores próximos do cruzamento entre duas gaussianas subsequentes, e apresenta maior homogeneidade (menor distorção) entre os pontos quando as gaussianas têm inclinação aproximada de 45° .

Figura 5.1: Influência dos campos receptivos na separação dos pontos.

Pontos de máximo e mínimo. Seja uma RNP-SRM-RBF formada por $n.g$ neurônios de entrada (associados às $n.g$ gaussianas dos campos receptivos) e um único neurônio de saída j_1 . Seja ainda uma dimensão de entrada x_1 submetida às g primeiras gaussianas para transformação em uma sequência de pulsos temporais que serão enviados aos neurônios $i_1 - i_g$ da camada de entrada.

Os gráficos apresentados na figura 5.2 mostram o comportamento da somatória $t_{i_1} + t_{i_2} + \dots + t_{i_g}$ das latências dos disparos dos g neurônios da camada de entrada para diversos valores de x_1 , com $0 \leq x_1 \leq 1$. Claramente o conjunto de pulsos será gerado mais rapidamente para valores de entrada x_1 próximos da metade entre seu valor máximo e mínimo. Pulsos codificando as entradas com valores próximas aos limites (máximo ou mínimo) tendem a ser mais tardios.

O eixo horizontal das figuras apresenta uma dimensão de entrada $[0,1]$ e o eixo vertical apresenta as latências geradas pelas gaussianas do campo receptivo, representadas em azul. Tomando-se, por exemplo, $x_1 \cong 0.73$, a somatória das latências dos três pulsos gerados (1, 14 e 35) é aproximadamente 50. As curvas em vermelho representam a somatória das latências dos pulsos para todos os possíveis valores de entrada $[0,1]$, com: a) 3 gaussianas; b) 4 gaussianas.

Figura 5.2: Somatória dos valores das latências na geração dos pulsos para campos receptivos gaussianos.

O fato da somatória de latências dos pulsos gerados em cada dimensão de entrada possuir pontos de máximo e mínimo tem implicações no modo como a rede processa os dados de entrada de duas formas distintas: *i*) estabelece o centro do grupo representado pelo neurônio de saída; e *ii*) institui uma forma preferencial para a generalização da rede.

Seja a situação onde todos os pesos sinápticos da rede ($w_{i_1 j_1}^1$ a $w_{i_{ng} j_1}^k$) são iniciados igualmente. Nesses termos, o ponto em $\mathcal{R}^n$ que terá o disparo mais rápido do neurônio j_1 será aquele cujos pulsos chegarem mais rapidamente a este neurônio. Como a somatória das latências é menor na posição central de $\mathcal{R}^n$, a codificação utilizada estabelece quatro consequências no que diz respeito às

condições iniciais dos neurônios da RNP-SRM-RBF:

- Proposição $\mathcal{P}1$
 - Os neurônios de saída têm como centro dos respectivos grupos o ponto central das n dimensões de entrada;
- Proposição $\mathcal{P}2$
 - A rede apresenta generalização característica em forma decrescente radial a partir do centro do espaço $\mathbb{R}^n$;
- Proposição $\mathcal{P}3$
 - A posição inicial de excitação máxima dos neurônios pode ser alterada mediante deslocamentos nos campos receptivos gaussianos;
- Proposição $\mathcal{P}4$
 - A discretização dos tempos de disparo do neurônio pode ser alterada mediante alteração na altura h das gaussianas.

As proposições $\mathcal{P}1 - \mathcal{P}3$ são experimentalmente comprovadas através da figura 5.3. Cada gráfico desta figura apresenta o tempo necessário para o disparo de um neurônio (codificado pela cor cuja legenda encontra-se na barra lateral) para diversas instâncias de entrada distribuídas pelo espaço $\mathbb{R}^3 : [0, 255]$. Cada ponto presente em um dos cinco planos representa uma instância de entrada. O neurônio dispara mais rápido para os pontos assinalados em vermelho, e dispara mais lentamente para os pontos assinalados em azul. A figura 5.3 apresenta a variação do ponto de disparo mais rápido de um neurônio pulsado para três diferentes condições iniciais. Muito embora Natschläger e Ruf (1998) tenham proposto a inicialização da rede com pesos aleatoriamente distribuídos, pode-se observar que a condição inicial da rede não é significativamente alterada para uma larga faixa de variação dos pesos iniciais.

A figura 5.4 comprova experimentalmente a proposição $\mathcal{P}4$, apresentando as implicações da escolha do parâmetro h dos campos receptivos gaussianos na generalização em $\mathbb{R}^3$. Nesta figura, entradas codificadas com pequena amplitude na somatória dos atrasos dos neurônios (curva em vermelho) implicam em neurônios que possuem pequena discretização em seu tempo de disparo (número de níveis indicados na barra lateral). O aumento da constante h nos campos gaussianos implica em um aumento na amplitude da somatória das latências, o que provoca um aumento no número de níveis discretos para o tempo de disparo do neurônio.

Evidentemente, durante o processo de aprendizado espera-se obter variação nos tempos de disparo dos neurônios pulsados – e no seu ponto de disparo mais rápido – via alterações no vetor W . A rede é capaz de apresentar generalizações bastante complexas e fortemente não-lineares (ver exemplos na figura 5.5). Contudo, o algoritmo de aprendizado de Natschläger e Ruf (1998) com a codificação

a) Generalização de uma RNP com pesos sinápticos aleatoriamente escolhidos com variação de $\pm 25\%$; b) Generalização de uma RNP com pesos sinápticos aleatoriamente escolhidos com variação de $\pm 100\%$; c) Três campos receptivos (em azul) com seus valores de centro deslocados. A somatória das latências na geração dos pulsos (em vermelho) desloca-se para a direita; d) O centro correspondente do neurônio pulsado considerando que as três dimensões de entrada são codificadas como o apresentado em 'c'. As cores – identificadas na legenda lateral – representam o número de iterações competitivas para o disparo do neurônio. O símbolo '+' representa o ponto com o disparo mais rápido (centro do grupo).

Figura 5.3: Condições iniciais da RNP-SRM-RBF.

de Bohte (2003) em condições adequadas de aplicação – discutidas no decorrer do capítulo – não deve conduzir a esse tipo de generalização, mas sim àquela onde os neurônios assemelham-se a neurônios estáticos de base radial.

Em linhas gerais, os pontos abordados nesta seção permitem dizer que os campos receptivos gaussianos com desvio padrão dentro de certos limites parecem manter homogêneas as instâncias de entrada após a transformação para o domínio do tempo. Contudo, quanto mais espaçados no tempo estiverem os pulsos recebidos dos neurônios da camada de entrada, melhor separáveis eles serão. Por este motivo, é importante que o parâmetro h não seja adotado demasiadamente pequeno. Porém, utilizar valores altos de h estabelece uma tendência de generalização para o NP que pode alterar sua forma natural de processamento.

Esquerda: somatória das latências nos disparos (em vermelho) para diferentes configurações de uma rede com 3 campos receptivos gaussianos (em azul). **Direita:** generalização da rede no espaço $\mathbb{R}^3$ com as três dimensões de entrada codificadas com os campos gaussianos apresentados. a) Gaussianas com $h = 10$, e os tempos de disparo apresentando 1 único nível de discretização; b) Gaussianas com $h = 50$ e tempos de disparo com 2 níveis de discretização; c) Gaussianas com $h = 100$ e tempos de disparo com 5 níveis de discretização; d) Gaussianas com $h = 119$ e generalização com 6 níveis de discretização.

Figura 5.4: Influências do parâmetro h na discretização dos tempos de disparo do NP.

a) Dois exemplos distintos de generalização da rede com pesos sinápticos altamente aleatórios;
 b) generalização (à direita) obtida com desvio padrão das gaussianas excessivamente baixo (à esquerda).

Figura 5.5: Generalizações adversas de uma RNP.

5.3 Modelo de Natschläger

Natschläger e Ruf (1998) propuseram uma implementação baseada no modelo do neurônio pulsado RBF proposto por Hopfield (1995) (discutido na seção 2.5.2). Uma das questões abertas neste contexto é qual é a relação que existe entre esses dois modelos. Esta seção procura discutir essa relação e apresentar algumas considerações adicionais do modelo de Natschläger e Ruf (1998).

Comparativo com o modelo de Hopfield. Em um primeiro momento, o modelo de Natschläger e Ruf (1998) parece propor uma simples discretização dos valores de atraso de propagação axonal δ_{ij} do modelo de Hopfield (1995). Tipicamente a saturação de um dos pesos das conexões entre cada neurônio em w_{max} e dos demais $k - 1$ pesos em $w_{min} = 0$ seleciona o δ_{ij} desejado, tarefa que não pode ser realizada diretamente no modelo de Hopfield (1995) devido à falta de um algoritmo de aprendizado para tal.

Consideremos, contudo, o início do processo de aprendizado nos dois modelos. A utilização do tempo de disparo como critério de distância na rede de Hopfield

(1995) apresenta uma seria restriao: como cada δ_{ij} precisa ser inicializado com algum valor – qualquer que seja –, cada neuronio tera uma regiao de influencia bem definida em $\Re^n$. Se for apresentada a rede uma instancia relativamente distante de todos os m centros das classes em $\Re^n$, e possivel que nenhum neuronio dispare. Embora este seja justamente o comportamento desejado para a rede, utilizar a medida de distancia com fundamento biologico neste modelo estabelece uma grande dificuldade para iniciar o seu processo de aprendizado.

Sob este prisma, o modelo de Natschlager e Ruf (1998) trouxe uma importante contribuiao que vai muito alem da discretizaao: no inicio do processo de aprendizado, o NP responde a instancias em todo o espaco de $\Re^n$, uma caracteristica sem paralelo na abordagem original. Este fato altera drasticamente a dinamica do potencial μ_j , que agora recebe um numero muito maior de contribuioes e assegura o disparo de todo neuronio $j \in J$ para qualquer conjunto de entradas.

Sob uma otica mais ampla, trata-se entao de uma metodologia capaz de garantir que a medida de distancia dada pelo tempo de disparo do neuronio aplique-se a todas as instancias durante o processo de treinamento. Sob esta otica os modelos de Hopfield (1995) e Natschlager e Ruf (1998) nao sao equivalentes.

Proposiao
P5

Aprendizado. O processo de aprendizado da rede de Natschlager e Ruf (1998) – apresentado na figura 3.4 – permite prever a existencia de uma antecipaao do tempo de disparo e tambem de um deslocamento dos pesos sinapticos durante a seleao dos atrasos axonais. Uma simulaao deste mesmo processo e apresentada na figura 5.6, onde pode-se observar esses eventos. A aplicaao sucessiva da regra de aprendizado em condioes normais de aplicaao – discutidas adiante – altera de forma distinta o tempo de disparo para a instancia de entrada e tempo de disparo para as demais instancias (com uma pequena generalizaao).

Refinamento da sintonia da RNP. Existem diversos conjuntos de pesos sinapticos que fazem um neuronio j disparar em um dado instante de tempo t_j . O menor desses vetores e aquele onde um de seus membros δ_{ij}^k aproxima-se da menor unidade de tempo, ja que nao ocasionara um atraso desnecessario no disparo de j . Em condioes normais de processamento, este e o vetor selecionado pelo algoritmo de aprendizado. Em outros termos, dentro de um conjunto $\mathcal{H}_\Delta$ de hipoteses para o conjunto de atrasos Δ , o processo de deslocamento dos pesos sinapticos causados pela constante c na equaao 3.17 e responsavel por selecionar aquela que implica no disparo mais rapido do neuronio.

Simulao da influncia do aprendizado no potencial de um neurnio de sada de uma RNP com $n = 3$, $m = 1$, $g = 6$, $\sigma = 1$, $d = 250$, $h = 100$, $w_{inicial} = 20$, $\tau_m = 10.000$, $\tau_s = 0.1$. A rede foi treinada com um nico exemplo de treinamento. a) disparo dos $n.g$ neurnios da camada de entrada com a codificao temporal para o exemplo de treinamento (0.234202, 0.232397, 0.226718); b) situao do potencial do neurnio de sada antes do aprendizado; c) iterao 25; d) iterao 50; e) iterao 100; f) iterao 1500. Observando a evoluo do treinamento  possvel notar a antecipao do disparo do neurnio treinado. O disparo de um neurnio $i \in I$ d-se no mximo no tempo $t = h$.

Figura 5.6: Simulao da adaptao dos pesos sinpticos e potencial atravs do algoritmo de aprendizado.

critério
de parada

Muito embora um critério de parada não tenha sido formalmente estabelecido por autores prévios, é óbvio que o treinamento deve ser suspenso antes que $t_j = h$, a partir de quando o neurônio poderá alterar drasticamente seu processamento.

5.4 RNP-SRM-RBF com campos receptivos gaussianos

Como o apresentado nas seções anteriores, a codificação utilizando campos receptivos gaussianos tem implicações nas condições iniciais e na distribuição do tempo de disparo dos neurônios, sobretudo para valores altos da amplitude dos campos gaussianos h . O uso desta codificação na rede de Natschläger e Ruf (1998) tem implicações no processo competitivo entre os neurônios da camada de saída. Esta seção apresenta algumas das características relevantes para a compreensão dessas últimas implicações.

estabilidade

Importância da estabilidade na soma dos pesos sinápticos Considere

um caso onde o módulo do parâmetro de aprendizado b (valor de decremento dos pesos sinápticos, apresentado na eq. 3.17) é escolhido demasiadamente pequeno. Quando a função de aprendizado for aplicada sobre os pesos sinápticos de um neurônio, esta fará um grande acréscimo em um dos pesos sinápticos, um acréscimo menor nos pesos vizinhos deste (dependendo do valor da variável β), e um pequeno decréscimo nos demais, ampliando significativamente o peso sináptico médio do neurônio w_{med} . Considerando a equação 2.21, o neurônio tende a disparar mais cedo pelo aumento de θ (inclinação do incremento no potencial da membrana). Este processo, portanto, fará com que o tempo de disparo do neurônio para todas as instâncias de entrada seja reduzido devido à escala dos pesos sinápticos, e não devido aos picos na distribuição dos pesos (NATSCHLÄGER; RUF, 1998), alterando completamente a dinâmica do aprendizado.

escala
dos pesos
sinápticos

Da mesma forma, quando o módulo de b é adotado demasiadamente grande, a função de aprendizado levará à redução de w_{medio} , que levará à redução de θ , que fará com que o tempo de disparo do neurônio aumente para as instâncias de entrada. A figura 5.7 exemplifica a implicação da escolha de b no tempo de disparo de uma RNP.

Uma vez que o tempo de disparo dos neurônios – o critério de distância da abordagem – é atualizado de formas distintas para as três diferentes possibilidades de evolução dos pesos sinápticos (crescente, decrescente ou estável), o processo

Uma RNP treinada com a função gaussiana durante 500 iterações de aprendizado com um único exemplo de treinamento e com $n = 3$, $g = 3$, $m = 3$, $c = -2.85$, $d = 450$, $h = 200$, $\beta = 1.67$, $\vartheta = 3600$, $w_{inicial} = 20$. a) Evolução da somatória dos pesos sinápticos dos neurônios da camada de saída com a evolução do treinamento; b) Evolução do tempo de disparo dos neurônios vencedores do processo competitivo com a evolução do treinamento. As redes treinadas com forte redução na somatória dos pesos sinápticos têm substancial aumento no tempo de disparo. Quando a rede não altera sensivelmente a somatória dos pesos sinápticos por causa do reescalonamento do vetor de pesos, o tempo de disparo tende a evoluir apenas devido ao processo competitivo e à regra de aprendizado. Quando a rede é treinada com aumento no valor médio dos pesos, o tempo de disparo é reduzido. É importante observar que pequenas variações da ordem de 0 a -0.3 em b implicam em dinâmicas completamente distintas no processo de aprendizado.

Figura 5.7: Simulação da influência do parâmetro b na somatória dos pesos sinápticos e no tempo de disparo dos neurônios.

competitivo entre os neurônios da camada de saída tem sua dinâmica alterada de três formas distintas:

- i. Em uma rede com *soma dos pesos sinápticos crescente*, o neurônio vencedor do primeiro processo competitivo tende a ter seu tempo de disparo reduzido para todas as possíveis instâncias de entrada e, portanto, tenderá a vencer todos os processos competitivos subsequentes. Esta condição é crítica e inviabiliza o aprendizado;
- ii. Em uma rede com *soma dos pesos sinápticos decrescente*, o neurônio vencedor do processo competitivo tende a ter o tempo de disparo para a instância corrente ligeiramente incrementado e os tempos de disparo para as demais instâncias incrementados com mais veemência. Desta forma, o neurônio vencedor tende a perder futuras competições, até que outros neurônios sejam atualizados segundo a mesma regra. Trata-se de uma situação que estimula o processo competitivo, já que outros neurônios (que não o vencedor) terão preferência a disparar. A existência do processo competitivo é,

portanto, assegurada;

- iii. Em uma rede com *soma dos pesos sinápticos constante*, o neurônio vencedor do processo competitivo tende a ter o tempo de disparo decrementado para todas as instâncias de entrada e incrementado para a instância corrente. Trata-se da situação ideal de trabalho segundo Natschläger e Ruf (1998).

A influência de b no processo competitivo dos neurônios é apresentado de forma simplificada na figura 5.8, que apresenta resultados experimentais do tempo de disparo de três neurônios da camada de saída de uma RNP treinada com diferentes valores de b .

É importante salientar que a inserção da codificação utilizando campos receptivos gaussianos impõe alterações no processamento da rede originalmente concebida por Natschläger e Ruf (1998). Considere uma situação onde uma RNP tem seu parâmetro b calibrado de forma a manter a soma dos pesos sinápticos constante. Como o mencionado nas seções anteriores, a codificação tende a impor uma generalização centralizada no centro de $\mathfrak{R}^n$. Assim sendo, se um neurônio tiver seu centro deslocado pela aplicação sucessiva do processo de treinamento, seu processo de convergência pode ser perturbado pela forma como os tempos de disparo distribuem-se devido à codificação via campos receptivos gaussianos. A figura 5.9 exemplifica esta situação.

A situação ilustrada pela figura 5.9 é bastante comum no trato com a RNP utilizando os campos receptivos gaussianos. Ainda que exista o deslocamento do centro na direção da instância de entrada, no treinamento o neurônio é conduzido a um estado onde ele mantém um tempo de disparo mais rápido que os demais neurônios da rede para a quase totalidade das possíveis instâncias, inviabilizando o processo competitivo, muito embora w_{medio} seja constante.

Hipótese $\mathcal{H}_1$ Baseado nas considerações e estudos realizados, estabelece-se a hipótese de que uma alteração na sistemática de aprendizado proposta por Natschläger e Ruf (1998) pode melhorar a convergência da RNP-SRM-RBF quando se utiliza campos receptivos gaussianos com grande número de atrasos axonais d . Neste paradigma, o trabalho inicial com a rede na região de decremento dos pesos sinápticos pode torná-la mais robusta no que diz respeito à competição nas primeiras iterações. A figura 5.10 apresenta uma rede treinada sem sucesso na região de estabilidade e a figura 5.11 apresenta a mesma rede com a convergência comprovada na região de decremento de w_{medio} .

A rede da figura 5.7 com três neurônios de saída treinada com um único exemplo por 50 iterações e com $b = -0.1$ (esq.), $b = -0.07$ (centro) e $b = 0$ (dir.); a) neurônios vencedores dos processos competitivos; b) tempos de disparo do neurônio de saída 1 para cada valor de b após as 50 iterações; c) neurônio 2; d) neurônio 3. $b = 0$: w_{medio} crescente faz com que o neurônio 1 vença o primeiro processo competitivo, tenha seu tempo de disparo reduzido para todas as possíveis instâncias e passe a vencer todos os processos competitivos subsequentes. Os tempos de disparo do neurônio 3 – que nunca venceu o processo competitivo – podem ser tomados como situação inicial de todos os neurônios da rede, com disparo mais rápido em 256 iterações e o mais lento em 264. Após o treinamento o neurônio vencedor dispara em 251–261. $b = -0.007$: o neurônio tem w_{medio} aproximadamente estável, e a evolução do processo de aprendizado tende a depender exclusivamente da dinâmica da competição entre os neurônios. Neste caso, como existe um único exemplo de treinamento, um único neurônio vence todos os processos competitivos. Seu centro desloca-se em direção ao exemplo e seus tempos de disparo são reduzidos no centro e aumentados nas demais regiões (255–265). $b = -0.1$: o neurônio vence o processo competitivo e, como w_{medio} é decrescente, seu tempo de disparo é aumentado para todas as instâncias de entrada, fazendo com que ele perca o processo competitivo seguinte para os outros neurônios. Neste caso, o centro de todos os neurônios é deslocado na direção do exemplo.

Figura 5.8: Implicações de b no processo competitivo dos neurônios.

a) Situação inicial de um neurônio de saída de uma RNP com três entradas x_1, x_2 e x_3 que dispara entre 171 e 180 iterações competitivas. O neurônio é treinado com o exemplo $P1=(36, 153, 208)$ por uma única iteração. Antes do treinamento o disparo com P1 ocorre após 173 iterações competitivas e P2 em 180. w_{medio} é aproximadamente constante, com sutil tendência ao decremento. b) Situação do neurônio de saída após 1 iteração de treinamento com $\eta = 100$. O centro passa a envolver a posição do exemplo, e os pontos P1–P2 disparam em 171–181 iterações; c) Situação do neurônio de saída após 1 iteração de treinamento com $\eta = 500$. Disparos de P1–P2 em 170–182 iterações; d) Situação do neurônio de saída após 1 iteração de treinamento com $\eta = 1000$. P1–P2 disparam em 170–184 iterações. A rede exibe uma tendência de reduzir o tempo de disparo para a instância de entrada (ponto P1), mas a generalização da rede não se desloca da mesma forma, mantendo em grande parte a tendência de generalização inicial ao redor ponto central de $\mathbb{R}^n$ devido à codificação imposta pelas gaussianas. É importante observar que se este neurônio recebesse após o treinamento uma instância em P3, sensivelmente distante do exemplo P1, ele dispararia muito mais rápido do que na situação original (172 x 175 iterações). Em uma rede com mais neurônios, esta generalização imposta pela codificação ao redor do centro inviabilizaria o processo competitivo.

Figura 5.9: Implicação de uma soma constante dos pesos sinápticos no processo competitivo.

RNP com $n = 3$, $g = 3$, $m = 3$, $d = 450$, $b = -0.00725$, $\beta = 1.67$, $w_{inicial} = 20$ e treinada com 3 exemplos pertencentes a classes distintas. a) Exemplo utilizado em cada iteração de treinamento (0 a 2) e o neurônio vencedor de cada processo competitivo (apenas o neurônio 1 vence todas as competições); b) Somatória dos pesos sinápticos ligeiramente decrescente; c) Tempo de disparo dos neurônios vencedores; d) Exemplos utilizados no treinamento; e) Tempos de disparo dos neurônios 1, 2 e 3 na iteração $t = 0$; f) tempo de disparo dos neurônios na iteração $t = 500$. O processo competitivo não é bem sucedido porque o tempo de disparo do neurônio 1 é sempre decrementado para todas as possíveis instâncias de entrada (256–264 em $t = 0$ para 248–262 em $t = 500$).

Figura 5.10: Falha em processo competitivo de uma RNP com somatória dos pesos sinápticos constante.

RNP com $n = 3$, $g = 3$, $m = 3$, $d = 450$, $b = -0,0145$, $\beta = 1.67$, $w_{inicial} = 20$ e treinada com 3 exemplos de classes distintas. a) Exemplo utilizado em cada iteração de treinamento e os neurônios vencedores de cada processo competitivo; b) Somatória dos pesos sinápticos fortemente decrescente; c) Tempo de disparo dos neurônios vencedores; d) Tempos de disparo dos neurônios 1, 2 e 3 na iteração $t = 0$; e) Tempo de disparo dos neurônios na iteração $t = 100$; f) tempo de disparo dos neurônios na iteração $t = 500$. O processo competitivo é bem sucedido com os tempos de disparo ligeiramente decrescentes (256–264 em $t = 0$ para 258–276 em $t = 500$).

Figura 5.11: Sucesso em processo competitivo de uma RNP com somatória decrescente nos pesos sinápticos.

O ponto de estabilidade dos pesos. Para manter uma soma constante dos pesos sinápticos, Natschläger e Ruf (1998) propuseram a escolha de uma função de aprendizado tal que sua integral ao longo do intervalo de tempo Δt seja nula.

$$\int_{-\infty}^{\infty} L(\Delta t).dt \approx 0 \quad (5.1)$$

Segundo esta sistemática, um dos parâmetros da função de aprendizado poderia ser isolado, de forma a manter a integral aproximadamente nula. Utilizando a função de aprendizado proposta pelos autores (eq. 3.17), isso implicaria em obter uma relação do tipo $b = f(c, \beta, \Delta t, d)$. Contudo, esta abordagem esbarra no problema de que a função de aprendizado apresentada pelos autores utiliza uma curva gaussiana e^{x^2} , cuja integral não tem solução analítica.

Como forma de contornar este problema, Natschläger e Ruf (1998) propõem realizar uma somatória de valores discretos da função de aprendizado para validar possíveis valores de b , como o ilustrado pela figura 5.12.

Os círculos denotam como os pesos sinápticos são alterados quando da aplicação da função de aprendizado, e podem ser utilizados para realizar uma estimativa do valor adequado de b para a estabilização da soma dos pesos sinápticos. Extraído de Natschläger e Ruf (1998).

Figura 5.12: Valores discretizados da função de aprendizado.

Pelos motivos apresentados nas seções anteriores, é de extrema importância estimar o parâmetro b que leva à condição de estabilidade na somatória de pesos sinápticos, sobretudo na rede com campos gaussianos. A metodologia de Natschläger e Ruf (1998) fornece uma boa ferramenta para tal, mas uma função de aprendizado integrável que permita o equacionamento imediato do parâmetro b seria de grande relevância, sobretudo para grandes valores de d .

5.5 Possíveis novos horizontes

Na quase totalidade das abordagens do aprendizado baseado em instância (*k-means*, LVQ, SOM, entre outros), o algoritmo de aprendizado fundamenta-se na idéia de armazenar todas as informações sobre as instâncias previamente apresentadas ao neurônio em um único vetor que representa o membro típico daquela classe (centro da classe ou *codebook*).

Quando a regra de aprendizado da RNP é aplicada para o conjunto de pesos sinápticos de um neurônio, o incremento dos pesos sinápticos (dado pelo pico da curva de aprendizado) utiliza um fator de correção muito maior do que o decremento para os demais pesos (dado pelo fator b da curva de aprendizado). Se duas instâncias de classes distintas forem apresentadas a um neurônio para adaptação por algumas iterações, não há impecílio a que este neurônio apresente convergência para ambos os centros. Estabelece-se, portanto, a proposição de que um neurônio pulsado pode apresentar mais de um centro.

Proposição
 $\mathcal{P}_6$

Uma comprovação experimental de que um neurônio pulsado pode apresentar pontos de resposta mais rápida (centros) para mais de um ponto em $\mathbb{R}^n$ é apresentada nas figuras 5.13 e 5.14, que apresentam respectivamente uma RNP e seu vetor de pesos sinápticos. Apenas através da observação dos pesos sinápticos é possível identificar que em um dos casos o neurônio convergiu para o centro de duas classes, já que mais de um grupo de sinapses foi selecionado em cada uma das 12 conexões $i - j$. É importante observar que no caso da figura 5.14a, também múltiplos δ foram selecionados em cada conexão $i-j$. Contudo, isso deve-se à taxa de generalização β da rede, como o apresentado na figura 6.7.

Hipótese
 $\mathcal{H}_3$

Desta forma, estabelece-se por hipótese que o vetor de pesos sinápticos W pode ser utilizado para avaliar a qualidade da convergência do neurônio pulsado.

5.6 Resumo

Este capítulo procurou reavaliar a rede neural pulsada de base radial utilizando o modelo de neurônio SRM e com aprendizado não supervisionado segundo o prisma do aprendizado baseado em instâncias.

Campos receptivos gaussianos com inclinações próximas a 45° impõem uma transformação sobre os dados de entrada que não tem grande influência na separação dos pontos, mas que: *i*) estimulam uma generalização em forma radial; *ii*) atribuem os centros das classes no início do processo de treinamento.

a) Os dois pontos indicados com o sinal '+' são exemplos utilizados no processo de treinamento de uma rede com 2 neurônios de saída, ou seja, duas classes. Os pontos indicados com o sinal '.' são as projeções dos exemplos nos planos laterais e inferior. b) Os pontos indicados com o sinal '+' representam pontos no espaço de entrada $\mathbb{R}^3$ para os quais um neurônio treinado com sucesso tem sua resposta mais rápida. O neurônio convergiu para uma das duas classes; c) Os pontos indicados com o sinal '+' representam pontos no espaço de entrada $\mathbb{R}^3$ para os quais a resposta de um neurônio treinado sem sucesso é mais rápida. O neurônio tem sua resposta mais rápida para pontos que não estão relacionados com nenhum dos exemplos de treinamento, em um processo muito similar ao apresentado na figura 4.2. Este neurônio apresenta mais de um centro, já que os pontos de resposta máxima não estão espacialmente agrupados.

Figura 5.13: Tempos de disparo de um neurônio da camada de saída para treinamentos com e sem sucesso.

Um modelo de rede e algoritmo de aprendizado para **sintonia de Δ pela seleção de sinapses** vem preencher importantes lacunas deixadas pelo modelo básico de rede pulsada RBF. Dentre suas características, destacam-se: *i*) torna possível utilizar o tempo de disparo do neurônio como critério de distância aplicável a todos as instâncias de entrada; *ii*) estimula futuras soluções moldando o critério de distância, que é função de uma complexa dinâmica no domínio do tempo; *iii*) oferece uma importante informação sobre o ponto de convergência do neurônio no espaço de entrada que é desprezada pelos algoritmos de aprendizado,

Vetor de pesos sinápticos W com 5400 pesos de uma RNP com $g=4$, $i=12$, $d=450$ após o processo de treinamento. a) neurônio da figura 5.13b, treinado com sucesso; b) neurônio da figura 5.13c, treinado sem sucesso (neurônio convergiu para o centro de duas classes).

Figura 5.14: Padrão de pesos sinápticos de um neurônio após o treinamento.

e pode ter importante contribuição na dinâmica da convergência.

O **aprendizado** da rede apresenta como características principais: *i*) o algoritmo tem a capacidade de selecionar o vetor de atrasos de propagação que proporciona o disparo mais rápido do neurônio para a instância de entrada; *ii*) o processo competitivo da rede tem grande proximidade com o *k-means*, e caracteriza-se pela estabilização em mínimos locais da superfície de erro; e *iii*) a taxa de decremento da função de aprendizado é de vital importância para garantir a existência do processo competitivo entre os neurônios da camada de saída.

A tabela 5.1 apresenta uma revisão do modelo de rede e aprendizado sob a ótica do aprendizado de máquinas.

Tabela 5.1: Quadro resumo da RNP-SRM-RBF com campos receptivos gaussianos e aprendizado não-supervisionado sob a ótica do AM.

Conceito	RNP-SRM-RBFs
Pré-processamento (campos gaussianos)	$f : \mathfrak{R}^n \rightarrow \mathcal{V}$, com $\mathcal{V} = \{v_1, v_2, \dots, v_{n.g}\}$ e $v \in \mathfrak{R}_+$
Entrada	$\{v_1, v_2, \dots, v_{n.g}\}$ tomados como tempos de disparo $\{t_{i_1}, t_{i_2}, \dots, t_{i_n}\}$
Saída	Código 1-de- m com os tempos de disparo $\{t_{j_1}, t_{j_2}, \dots, t_{j_m}\}$
Mapeamento	$\{t_{i_1}, t_{i_2}, \dots, t_{i_n}\} \rightarrow \{t_{j_1}, t_{j_2}, \dots, t_{j_m}\}$
Medida de distância	Tempo de disparo do neurônio, dado por uma complexa dinâmica temporal com $t_j = f(T_i, W, \epsilon, \mu, t)$
Inicialização dos centros das classes	Centros das classes no centro de $\mathfrak{R}^n$ com pequena variação aleatória
Função de aprendizado	$(1 - b) \cdot e^{-\frac{(\Delta t - c)^2}{\beta^2}} + b$
Generalização	β estimula a generalização nas proximidades do ponto de resposta máxima, e a codificação dos campos gaussianos estimula generalização sempre no ponto de partida
Metodologias para calibração de parâmetros	Não definido
Critério de parada	Não definido para a rede com campos gaussianos
Robustez na convergência	Via auto-inibição baseada no número de iterações vencidas pelo neurônio
Principais problemas	<i>i)</i> dificuldade para calibrar os muitos parâmetros da rede; <i>ii)</i> excessiva sensibilidade ao parâmetro b ; e <i>iii)</i> convergência inviabilizada em muitos casos práticos devido à codificação imposta pelos campos gaussianos
Novas possibilidades	<i>i)</i> medida de distância é um critério particular do neurônio; e <i>ii)</i> neurônio tem informação para identificar quando está convergindo de forma incorreta. Estimula-se alteração no critério de distância do neurônio utilizando informação puramente local

6 Concepção do Aprendizado

“A curiosidade é sempre o primeiro passo em um problema que vai ser resolvido” (Galileu Galilei).

Durante a revisão das RNP-SRM-RBFs utilizando campos receptivos gaussianos, três hipóteses foram levantadas para aprimorar seu processo de convergência:

- **Hipótese $\mathcal{H}_1$:** A sistemática de aprendizado de Natschläger e Ruf (1998) – que trabalha na região de estabilidade dos pesos sinápticos – não é a sistemática ideal de trabalho quando se utiliza a codificação via campos receptivos gaussianos. Neste contexto, o reescalonamento sináptico – isto é, o trabalho na região de decremento dos pesos sinápticos – pode compensar a tendência de generalização central imposta pela codificação e garantir robustez ao processo de convergência;
- **Hipótese $\mathcal{H}_2$:** Existe uma função de aprendizado propícia à utilização na RNP que é integrável e que permite a determinação automática da constante de decremento b através de dedução algébrica;
- **Hipótese $\mathcal{H}_3$:** O neurônio pulsado admite o uso de heurísticas que auxiliem seu processo de convergência baseadas em informação puramente local (via observação da situação dos pesos sinápticos do neurônio) e atuação puramente local (atuação sobre o critério de distância do neurônio, atrasando seu disparo em alguns casos).

Com base nestas hipóteses, este trabalho propõe a reestruturação do aprendizado apresentado na tabela 5.1 fazendo uso de: *i)* uma nova função de aprendizado; *ii)* reescalonamento sináptico para reduzir a influência dos campos receptivos gaussianos na generalização da rede; *iii)* metodologias específicas para calibração de ϑ e w_{max} ; *iv)* um critério de parada, e *v)* uma função heurística para avaliação da convergência da rede. Esta reestruturação é resumida na tabela 6.1.

Tabela 6.1: Sistemática proposta para o aprendizado na RNP-SRM-RBF

Função de aprendizado	Dada pela equação 6.1
Generalização	β estimula a generalização nas proximidades do ponto de resposta máxima, e o reescalonamento sináptico reduz a influência da codificação dos campos gaussianos na generalização
Metodologias para calibração de parâmetros	Metodologias específicas para a calibração dos parâmetros b (eq. 6.2), ϑ (eq. 6.4) e w_{max} (eq. 6.5)
Critério de parada	O treinamento é suspenso quando um dos pesos sinápticos atinge seu valor máximo ou mínimo
Avaliação da convergência	Utilizando função heurística com base em W

Este capítulo dedica-se ao detalhamento dessas propostas, e encontra-se organizado da seguinte forma: a seção 6.1 apresenta nova função de aprendizado $L(\Delta t)$ proposta para a rede. A seção 6.2 apresenta a sistemática de calibração automática do parâmetro b possível com o uso da nova função de aprendizado. A seção 6.3 apresenta sistemática de trabalho proposta para rede utilizando campos receptivos gaussianos. A seguir, a seção 6.4 apresenta a metodologia proposta para a calibração do limiar de disparo. A seção 6.5 destaca os procedimentos adotados para calibração do parâmetro λ , e, por conseguinte, do peso máximo w_{max} . A seção 6.6 apresenta uma função heurística propícia para utilização na rede pulsada.

6.1 Função de aprendizado *tanh*

Para facilitar a inicialização de alguns dos parâmetros da rede, propõe-se a substituição da função de aprendizado (eq. 3.17) pela apresentada na equação 6.1:

$$\Delta w_{ij}^k = \eta \cdot L(\Delta t) = \eta \cdot \left[(b - 1) \cdot \tanh^2 \left(\frac{\Delta t - c}{\xi} \right) + 1 \right], \quad (6.1)$$

onde: Δw_{ij}^k é a variação no peso sináptico da k -ésima sinapse entre os neurônios

i e j ; η é a taxa de aprendizado; $L(\cdot)$ é a função de aprendizado; t é o tempo de competição entre os neurônios, ocorrida a cada iteração de aprendizado; b é o parâmetro de decremento dos pesos sinápticos; c é o recuo no tempo da função de aprendizado; ξ é a taxa de generalização no aprendizado; $\tanh$ é a função tangente hiperbólica.

A figura 6.1 apresenta esquematicamente um gráfico da equação 6.1, que é bastante similar à função gaussiana proposta por Natschläger e Ruf (1998). As constantes utilizadas na nova função são também idênticas às constantes utilizadas na função gaussiana, exceção à nova constante de incremento a e ao parâmetro de generalização ξ , que é análogo (porém não idêntico) ao parâmetro β . A comparação entre as duas funções é apresentada na figura 6.2, onde pode ser observada a pequena diferença entre elas.

Figura 6.1: Função de aprendizado proposta para a rede neural pulsada.

a) Funções com $a=1$, $b=-0.3$, $c=-2.85$, $\beta = \xi = 1.67$; b) Funções com $a=1$, $b=-0.41$, $c=-4.85$, $\beta = \xi = 2.78$.

Figura 6.2: Comparativo entre as funções de aprendizado gaussiana e $\tanh$.

6.2 Calibração do ponto de estabilidade

A função de aprendizado proposta neste trabalho é integrável. Desta forma, é possível obter algebricamente a expressão de b para estimar o ponto de estabilidade na somatória dos pesos sinápticos.

Seja a função:

$$L(\Delta_t) = (b - a) \cdot \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) + a$$

aplicada sobre os pesos sinápticos de uma conexão entre os neurônios i e j com d sinapses. Considerando que a parte incremental da função não se encontre nas extremidades dos pesos, as áreas acima e abaixo do eixo horizontal podem ser igualadas forçando que a integral da função de $-d$ até 0 seja nula.

Figura 6.3: Integral sobre a função de aprendizado tanh.

Matematicamente:

$$\begin{aligned} \int_{-d}^0 \left[(b - a) \cdot \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) + a \right] d\Delta_t &= 0 \\ \int_{-d}^0 (b - a) \cdot \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) \cdot d\Delta_t + \int_{-d}^0 a \cdot d\Delta_t &= 0 \\ (b - a) \cdot \int_{-d}^0 \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) \cdot d\Delta_t + a \cdot \int_{-d}^0 1 \cdot d\Delta_t &= 0 \\ (b - a) \cdot \int_{-d}^0 \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) \cdot d\Delta_t + a \cdot \left[t \right]_{-d}^0 &= 0 \\ (b - a) \cdot \int_{-d}^0 \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) \cdot d\Delta_t + a \cdot \left[0 + d \right] &= 0 \\ (b - a) \cdot \int_{-d}^0 \tanh^2\left(\frac{\Delta_t - c}{\xi}\right) \cdot d\Delta_t + a \cdot d &= 0 \end{aligned}$$

Fazendo uma mudança de variável de Δ_t para u , teremos:

$$\begin{cases} u = \frac{\Delta_t - c}{\xi} = \frac{\Delta_t}{\xi} - \frac{c}{\xi} \\ du = \frac{1}{\xi} d\Delta_t \\ \Delta_t \rightarrow -d, u \rightarrow -\frac{d}{\xi} - \frac{c}{\xi} \\ \Delta_t \rightarrow 0, u \rightarrow -\frac{c}{\xi} \end{cases}$$

$$(b-a) \cdot \left[\int_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} \xi \cdot \tanh^2(u) \cdot du \right] + a \cdot d = 0$$

$$\xi \cdot (b-a) \cdot \left[\int_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} \tanh^2(u) \cdot du \right] + a \cdot d = 0$$

$$\xi \cdot (b-a) \cdot \left[\int_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} 1 - \operatorname{sech}^2(u) \cdot du \right] + a \cdot d = 0$$

$$\xi \cdot (b-a) \cdot \left[\int_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} 1 \cdot du - \int_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} \operatorname{sech}^2(u) \cdot du \right] + a \cdot d = 0$$

$$\xi \cdot (b-a) \cdot \left[[u]_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} - [\tanh(u)]_{-\frac{d}{\xi} - \frac{c}{\xi}}^{-\frac{c}{\xi}} \right] + a \cdot d = 0$$

$$\xi \cdot (b-a) \cdot \left[\left(\underbrace{-\frac{c}{\xi}} + \frac{d}{\xi} + \underbrace{\frac{c}{\xi}} \right) - \left(\tanh\left(-\frac{c}{\xi}\right) - \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right) \right] + a \cdot d = 0$$

$$\xi \cdot (b-a) \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] + a \cdot d = 0$$

$$(b-a) \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] + \frac{a \cdot d}{\xi} = 0$$

$$b \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] - a \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] + \frac{a \cdot d}{\xi} = 0$$

$$b \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] - a \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) - \frac{d}{\xi} \right] = 0$$

$$b \cdot \left[\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] - a \cdot \left[-\tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right) \right] = 0$$

Fazendo $b = f(a)$, teremos:

$$b = a \cdot \frac{-\tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right)}{\frac{d}{\xi} - \tanh\left(-\frac{c}{\xi}\right) + \tanh\left(-\frac{d}{\xi} - \frac{c}{\xi}\right)}$$

Dado que $d \gg 1$, temos que $(-\frac{d}{\xi} - \frac{c}{\xi}) \ll -1$. Desta forma:

$$b \cong \kappa_e \cdot a \cdot \frac{-\tanh(-\frac{c}{\xi}) - 1}{\frac{d}{\xi} - \tanh(-\frac{c}{\xi}) - 1} \quad (6.2)$$

A equação 6.2 é a expressão proposta para a estimação de b , onde: b é o valor utilizado para o decremento dos pesos sinápticos; κ_e é o coeficiente de estabilidade, que torna a integral aproximadamente nula para $\kappa_e = 1$, decrementa-a para $\kappa > 1$ e incrementa-a para $\kappa < 1$; c é o recuo no tempo da função de aprendizado; ξ é o parâmetro de generalização da rede; d é o número de conexões entre os neurônios i e j ; $\tanh$ é a função tangente hiperbólica de aprendizado. A figura 6.4 exibe a curva desta função e sua integral.

a) A função de aprendizado proposta; b) Curva dos valores de b em função do número de sinapses d entre cada conexão $i - j$ que tornam aproximadamente nula a integral da função de aprendizado.

Figura 6.4: Curva exibindo os valores de b que tornam nula a integral da função de aprendizado.

Pela natureza contínua, a precisão da função aumenta com o aumento do número de sinapses.

6.3 Generalização via escalonamento sináptico

Conforme abordado no capítulo anterior, a gradativa redução no peso sináptico médio dos neurônios estimula a competição entre os mesmos e reduz sua generalização no ponto central de $\mathcal{R}^n$, influência da codificação via campos

receptivos gaussianos. Para tanto, este trabalho adota – particularmente nos experimentos realizados nas seções adiante – a inicialização do parâmetro b com um valor superior ao que torna nula a integral da função de aprendizado, determinado automaticamente utilizando a equação 6.2.

6.4 Calibração do limiar ϑ

Para valores de limiar muito pequenos, o neurônio poderá disparar antes de receber todos os pulsos dos neurônios da camada de entrada. Para valores muito altos, o neurônio poderá não disparar. Ambas as situações são indesejadas. Obter valores para o limiar empírica ou experimentalmente é uma tarefa custosa. Esta seção propõe uma metodologia que pode ser utilizada para calibrar o limiar de disparo ϑ .

Pela natureza do modelo RNP, a rede não deve disparar antes de receber pulsos de todos os neurônios da camada de entrada, o que certamente ocorre após $t = h$. O tempo de disparo máximo do neurônio não pode ultrapassar d , já que o neurônio pode sofrer incremento em seu potencial até que existam sinapses para oferecê-lo. Uma metodologia, então, é prever que o disparo ocorra entre esses dois instantes. Esta situação é apresentada na figura 6.5.

Figura 6.5: Valores máximo e mínimo para a estimativa de ϑ .

Desta forma, utilizando o valor estimado de θ apresentado na equação 2.21, tem-se que:

$$\frac{\partial \mu_j}{\partial t} = \frac{\vartheta_{min}}{0,5.h} ; \quad \frac{\partial \mu_j}{\partial t} = \frac{\vartheta_{max}}{d - 0,5.h}$$

o que estabelece a seguinte relação para estimar valores para ϑ :

$$\therefore 0,5.h.n.g.w_{medio} < \vartheta < (d - 0,5.h).n.g.w_{medio} \quad (6.3)$$

onde: ϑ é o limiar de disparo do neurônio; h é a altura dos campos receptivos gaussianos; n é o número de dimensões de entrada; g é o número de gaussianas; w_{medio} é o peso inicial médio dos neurônios da rede.

Em particular, uma boa metodologia é fazer com que o neurônio dispare logo depois de $t = h$, evitando atrasos desnecessários. Uma relação empírica, particularmente útil é, então:

$$\hat{\vartheta} = 1,1.h.n.g.w_{medio} \quad (6.4)$$

onde: ϑ é o limiar de disparo do neurônio; h é a altura dos campos receptivos gaussianos; n é o número de dimensões de entrada; g é o número de gaussianas; w_{medio} é o peso inicial médio dos neurônios da rede.

6.5 Estimação do peso máximo

Para estimar o peso máximo w_{max} , Natschläger e Ruf (1998) propuseram a utilização da equação 3.18. Contudo, nenhuma relação foi proposta para estimar λ . Contudo, este parâmetro pode ser deduzido calculando-se a distância entre as duas raízes da função de aprendizado.

Seja a função:

$$L(\Delta_t) = (b - a).tanh^2\left(\frac{\Delta_t - c}{\xi}\right) + a$$

Esta função intercepta o eixo Δt sempre em dois pontos, que podem ser obtidos fazendo:

$$(b - a).tanh^2\left(\frac{\Delta_t - c}{\xi}\right) + a = 0$$

Desta forma, teremos:

$$\begin{aligned} tanh^2\left(\frac{\Delta_t - c}{\xi}\right) &= -\frac{a}{b - a} \\ tanh\left(\frac{\Delta_t - c}{\xi}\right) &= \pm\sqrt{-\frac{a}{b - a}} \end{aligned}$$

$$\frac{\Delta t - c}{\xi} = \operatorname{arctgh}\left(\pm \sqrt{-\frac{a}{b-a}}\right)$$

$$\Delta t - c = \xi \cdot \operatorname{arctgh}\left(\pm \sqrt{-\frac{a}{b-a}}\right)$$

$$\Delta t = \xi \cdot \operatorname{arctgh}\left(\pm \sqrt{-\frac{a}{b-a}}\right) + c$$

As duas raízes reais distintas serão:

$$\Delta t_1 = \xi \cdot \operatorname{arctgh}\left(+ \sqrt{-\frac{a}{b-a}}\right) + c$$

$$\Delta t_2 = \xi \cdot \operatorname{arctgh}\left(- \sqrt{-\frac{a}{b-a}}\right) + c$$

Logo, λ será dado por:

$$\lambda = |\Delta t_2 - \Delta t_1|$$

$$\lambda = \left| \xi \cdot \operatorname{arctgh}\left(+ \sqrt{\frac{a}{a-b}}\right) - \xi \cdot \operatorname{arctgh}\left(- \sqrt{\frac{a}{a-b}}\right) + 2 \cdot c \right| \quad (6.5)$$

A relação apresentada na equação 6.5 é a relação proposta para o cálculo de λ , onde: λ é o intervalo de tempo no qual a função de aprendizado é maior do que zero; ξ é a taxa de generalização; arctgh é a função arco tangente hiperbólico; a é o incremento da função de aprendizado, usualmente 1; b é a taxa de decremento da função de aprendizado; c é o deslocamento no tempo do pico da função de aprendizado.

6.6 Uso de função heurística

A técnica denominada *heurística* preocupa-se com o desenvolvimento de funções que apontam para direções geralmente interessantes na solução de um problema (RICH; KNIGHT, 1993). Na área específica dos algoritmos de busca, heurística refere-se a uma função que fornece uma estimativa de custo da solução. Segundo Russel e Norvig (1995), no entanto, hoje a palavra heurística é utilizada como adjetivo, referindo-se a qualquer técnica que melhore o desempenho médio de casos em uma tarefa de solução de problema (não necessariamente melhorando o desempenho no pior caso). Trata-se, em suma, de uma função que de alguma forma altera ou direciona a dinâmica de um problema de forma a conduzi-lo mais satisfatoriamente a uma solução.

Uma informação rica e disponível no NP – já mencionada no capítulo anterior – é a configuração de seus pesos sinápticos W . O resultado da aplicação de uma função de avaliação em W pode ser utilizado de pelo menos duas formas distintas: *i)* pelo neurônio, para alterar sua forma de convergência, buscando evitar a convergência para mais de uma classe; ou *ii)* simplesmente como uma medida de avaliação da qualidade da convergência da rede. Existem ao menos duas formas distintas para extrair estas informações relevantes de W .

Primeira possibilidade. A configuração dos pesos sinápticos de um neurônio tem reflexos em seu potencial. A figura 6.6 apresenta o potencial de um neurônio pulsado quando este encontra-se em um processo de convergência para o centro de duas classes. Desta forma, os ângulos θ e ϕ podem ser empregados em uma função heurística.

a) Potencial típico de um neurônio que dispara apenas quando todos os pulsos chegam simultaneamente; b) Potencial típico de um neurônio que encontra-se em processo de convergência para pontos excessivamente distantes (centro de duas ou mais classes). Neste caso, pesos sinápticos que convergiram de maneira inadequada estimulam significativamente (ângulo ϕ) o potencial do neurônio bem antes de seu disparo em $t = t_j$.

Figura 6.6: Comparação do potencial de um neurônio com convergência inadequada durante o processo de aprendizado.

Segunda possibilidade. Uma função heurística pode ser aplicada diretamente sobre o vetor de pesos sinápticos. Neste caso, uma possível função binária para avaliar a qualidade da convergência seria a apresentada na equação 6.6. A

figura 6.7 ilustra alguns casos típicos para a aplicação desta função.

$$\mathcal{H} = \begin{cases} 0, & \text{se o número de picos} > n.g \text{ ou se o sinal da derivada} \\ & \text{dos pesos sinápticos alterar-se mais de uma vez nos} \\ & \text{pesos referentes ao mesmo neurônio predecessor;} \\ 1, & \text{caso contrário.} \end{cases} \quad (6.6)$$

onde: n é o número de dimensões de entrada; g é o número de gaussianas.

Vetor dos pesos sinápticos de um neurônio da camada de saída que tem 5 predecessores imediatos e 10 conexões com cada um deles. a) Para cada um dos 5 neurônios predecessores, apenas um peso sináptico é estimulado (convergência adequada); b) Para cada um dos 5 neurônios predecessores, um peso sináptico é estimulado e sua vizinhança é estimulada pela generalização da rede (convergência adequada); c) Mais de um peso sináptico é estimulado em cada um dos 5 neurônios predecessores (convergência inadequada); d) Mais de um peso sináptico é estimulado em cada um dos 5 neurônios predecessores, e em particular nos pesos referentes ao neurônio 1 existe descontinuidade nos pesos estimulados (convergência inadequada).

Figura 6.7: Diferentes configurações de W para avaliação do processo de convergência.

6.7 Critério de parada

Uma vez de posse da sistemática proposta de decremento do peso sináptico médio, ao aplicá-la o neurônio tenderá a aumentar seu tempo de disparo com as sucessivas iterações de treinamento. O deslocamento de pesos sinápticos mencionado na seção 5.3 não ocorrerá, e, desta forma, o critério de parada em $t_j = h$, discutido na mesma seção, não terá validade.

Um critério de parada alternativo nesta rede é finalizar o treinamento quando todos os pesos sinápticos dos neurônios atingirem w_{max} ou w_{min} . Contudo, quando um neurônio atinge este limiar antes dos demais, alterações no processo competitivo são possíveis em decorrência do reescalonamento sináptico. De forma a evitar esta condição, propõe-se a parada do algoritmo na iteração na qual o primeiro peso sináptico atinja seu limiar.

Parte III

Resultados Experimentais

7 Materiais e Métodos

Este capítulo tem por objetivo expor a forma de condução dos experimentos realizados. Os resultados obtidos no processo de investigação do modelo da RNP-SRM-RBF utilizando a metodologia aqui descrita serão apresentados nos próximos capítulos. Inicialmente, a seção 7.1 aborda a forma de implementação da rede. A seção 7.2 apresenta um panorama geral da metodologia utilizada nos testes efetuados. A seção 7.3 destaca a forma de documentar o trabalho realizado.

7.1 Implementação

Uma rede de neurônios pulsados de base radial foi implementada utilizando linguagem C++ em ambiente Dev-C++ 4.9.9.2 (BLOODSHED SOFTWARE, 2005), executado no sistema operacional Windows®. No modelo implementado, as entradas foram codificadas utilizando campos receptivos gaussianos como o discutido na seção 2.5.3. Para o aprendizado da rede foi adotada basicamente a sistemática proposta no capítulo 6, com a função de aprendizado *tanh*. Os parâmetros da rede com metodologias específicas de calibração – discutidas também no capítulo 6 – foram calibrados conforme as sistemáticas propostas. O parâmetro b , particularmente, foi adotado de forma conduzir a rede intencionalmente ao ponto de decremento dos pesos sinápticos, estimulando a competição entre os neurônios da camada de saída. Após a coleta dos dados dos experimentos, gráficos foram gerados utilizando o software Matlab® 6.5 (THE MATHWORKS INC., 2002) em ambiente Windows®.

7.2 Metodologia de testes

Tradicionalmente, a avaliação de algoritmos é feita utilizando problemas padronizados, denominados *benchmarks*. Prechelt (1994) considera o *XOR* como o mais famoso desses problemas, decorrência do fato deste ter sido um dos grandes limitadores à expansão das RNAs antes do desenvolvimento do algoritmo da re-

tropropagação do erro. A análise do desempenho da RNP-SRM-RBF ao problema *XOR* foi primeiramente realizada por Bohte (2003). Desta forma, estimula-se a avaliação deste modelo em tarefas mais complexas utilizando novos *benchmarks*.

No intuito de avaliar o desempenho da rede em novas tarefas e de contribuir com o estabelecimento de algumas de suas limitações, o modelo foi avaliado em duas tarefas distintas: *i*) agrupamento de dados artificialmente gerados em $\mathbb{R}^3$; e *ii*) classificação da cor de pixels no domínio da visão computacional.

Grupos gerados artificialmente. Algumas coleções de pontos foram deliberadamente arranjadas nestes problemas no sentido de representar alguns dos principais tipos de nuvens. As nuvens de instâncias foram também dispostas de forma a permitir um bom acompanhamento da evolução do aprendizado da rede. Quando dados gerados artificialmente são utilizados, no entanto, as distribuições das classes são conhecidas *a priori*, o que permite o uso destas informações no sentido de ajustar os parâmetros da rede de forma a obter um desempenho adequado do algoritmo. Apesar da disponibilidade destas informações, nenhum estudo especial foi realizado para a calibração da rede nos testes efetuados com esses grupos. De forma a permitir futuras comparações com esses resultados, os dados artificiais utilizados neste trabalho podem ser obtidos no endereço: <http://www.sorocaba.unesp.br/professor/assimoes/clustering>.

Quando se avalia a evolução de um aprendizado utilizando dados artificialmente gerados, no entanto, não é possível saber quanto os resultados obtidos nesses problemas podem nos dizer a respeito da aplicabilidade do sistema em tarefas do mundo real (PRECHELT, 1994). Desta forma, este trabalho procura utilizar mais um tipo de avaliação para esta rede, visando garantir que os resultados obtidos sejam relevantes para ao menos um domínio real, nominalmente aquele sendo testado.

Classificação de cores. O conceito da palavra cor, embora aparentemente imediato, envolve muitas noções de ordem física e psicológica, o que dificulta muito sua modelagem.

Embora os comprimentos de onda da faixa de luz visível sejam grandezas contínuas ao longo de toda a faixa do espectro, quando um ser humano observa tais comprimentos de onda, estes são imediatamente agrupados em classes, fazendo uso de noções psicológicas e exemplos prévios de cada indivíduo na separação entre cada uma das cores visíveis, ou, na forma como é usualmente colocado, na

classificação de cores.

As cores são usualmente representadas – sobretudo no universo computacional – utilizando o paradigma tricromático. Nesse sistema, tipicamente adota-se três ondas primárias (cores) como três eixos ortogonais de um sistema de coordenadas, e as demais cores são representadas como uma tripla em função das cores dos eixos. O sistema RGB – cujos eixos são respectivamente vermelho, verde e azul – é o mais usual sistema de representação de cores, e por isso é adotado como padrão para a representação de cores neste trabalho.

A tarefa de classificar a cor dos pixels em uma imagem pode então ser revista como a tarefa de particionar adequadamente um espaço (usualmente o RGB em $\mathbb{R}^3$). Para uma descrição mais completa desta tarefa veja por exemplo Simões (2000). Classificadores aptos a realizar esta tarefa – sobretudo em ambientes não supervisionados – são de grande importância no âmbito da VC e do AM.

Nesse sentido, como segunda plataforma de estudos, o desempenho do classificador foi avaliado na tarefa da classificação de pixels utilizando o atributo cor. Este trabalho utiliza, entre outras, imagens selecionadas da base de dados para segmentação da Universidade de Berkeley (*Berkeley Segmentation Dataset*) (Berkeley Computer Vision Group, 2001) (MARTIN, 2001).

7.3 Documentação

Um dos grandes problemas tradicionalmente associados à pesquisa utilizando RNAs é a dificuldade de reproduzir os resultados de pesquisas anteriores. Este problema é particularmente amplificado em se tratando de RNPs, pois além do grande número de parâmetros, existe uma forte interdependência entre eles, tornando praticamente impossível reproduzir qualquer simulação sem o conhecimento da totalidade valores dos parâmetros utilizados. De fato, alguns dos trabalhos utilizando esta abordagem relatam problemas dessa natureza (BOHTE, 2003) (BOOIJ, 2004).

Visando contribuir no sentido de uma pesquisa científica aprofundada neste novo paradigma, os experimentos realizados neste trabalho foram conduzidos em acordo com a metodologia **PROBEN1** (PRECHELT, 1994). Trata-se de uma sistemática que define um conjunto de regras para aplicação de *benchmarks* a redes neurais artificiais. Este conjunto de regras diz respeito principalmente à forma de condução dos experimentos e à sua documentação. Busca-se, em princípio, garantir: *i*) que os experimentos sejam válidos; *ii*) que os experimentos possam ser

reproduzidos; *iii*) que os experimentos possam ser comparados. Apesar de tratar-se de uma metodologia primordialmente destinada a algoritmos de treinamento supervisionado, algumas de suas principais diretrizes são aqui aplicadas.

Para cada um dos problemas abordados, uma lista de pontos considerados relevantes para a reprodução dos resultados será apresentada, ainda que esta sistemática possa parecer repetitiva ao longo dos vários testes. Os pontos selecionados são apresentados como um *framework* na tabela 7.1.

Tabela 7.1: *Framework* utilizado nos experimentos realizados.

Ponto	Ítems documentados
Problema	Nome, tipo de dado, dimensões, descrição, endereço para obtenção
Conjuntos	Escala de entrada, escala de saída, conjunto de treinamento, conjunto de validação, conjunto de teste
Arquitetura da rede	Tipo de rede, constantes: n , m , d
Codificação temporal	Tipo de código, tipo de codificação, constantes: g , h , σ
Inicialização	Descrição, $w_{inicial}$;
Parâmetros do algoritmo	Constantes: a , b , c , w_{max} , η , ϑ , τ_m , τ_s , ξ , λ , particularidades
Término	Critério de parada, número de épocas.

8 Estudo de casos: agrupamentos 3D

Este capítulo tem por objetivo apresentar o desempenho do algoritmo proposto em alguns casos típicos de agrupamento em $\mathbb{R}^3$. Foram realizados no total quatro experimentos em situações distintas, apresentadas a seguir. Para cada um dos experimentos, os seguintes dados foram observados:

- i. **Convergência:** deslocamento do ponto de resposta mais rápida dos neurônios da camada de saída em função da iteração de aprendizado;
- ii. **Aprendizado dos neurônios:** tempo de disparo dos neurônios pulsados para diversos pontos em $\mathbb{R}^3$;
- iii. **Competição:** diagrama do neurônio vencedor de cada um dos processos competitivos realizados durante o aprendizado;
- iv. **Escalonamento:** curva quadrática da somatória dos pesos sinápticos x iteração de aprendizado;
- v. **Critério de distância:** curva tempo de disparo x iteração de aprendizado;
- vi. **Partição:** neurônios vencedores do processo competitivo para diversos pontos do espaço $\mathbb{R}^3$.

Os experimentos realizados foram:

- EXP-01: Agrupamento de instâncias organizadas em 3 grupos linearmente separáveis de tamanhos iguais;
- EXP-02: Agrupamento de instâncias organizadas em 3 grupos linearmente separáveis de tamanhos distintos;
- EXP-03: Agrupamento de instâncias organizadas em 5 grupos linearmente separáveis de tamanhos iguais;

- EXP-04: Agrupamento de instâncias organizadas em 2 grupos entrelaçados linearmente inseparáveis de tamanhos iguais;

8.1 Experimento EXP-01

A RNP foi treinada conforme especificado na tabela 8.1. A figura 8.1a apresenta os exemplos utilizados no treinamento da rede. A figura 8.1b ilustra o processo de convergência da rede. A figura 8.2 apresenta o tempo de disparo dos três neurônios da camada de saída para diversas instâncias de entrada em $\mathbb{R}^3$.

A ilustração do processo de convergência (deste experimento e dos experimentos subsequentes) foi gerada realizando-se estimativas da posição do centro dos neurônios RBF a partir das imagens dos tempos de disparo dos neurônios em diferentes momentos do treinamento. A seguinte metodologia foi adotada: para cada neurônio da camada de saída, os três menores tempos de disparo (tempos de disparo associados às três primeiras cores da parte superior da barra lateral de cores) foram observados. Todos os pontos em $\mathbb{R}^3$ – dentre os observados – para os quais o neurônio disparou em um desses tempos foram adicionados a uma lista `lista_centros`. O centro do neurônio pulsado foi estimado calculando-se a média dos pontos presentes em `lista_centros`. Este processo foi repetido para cada neurônio a cada vinte iterações de treinamento até o critério de parada ser atingido.

A figura 8.3a apresenta uma a uma as 430 iterações de treinamento, e, em cada caso, apresenta o exemplo de treinamento sorteado e o neurônio vencedor do processo competitivo. Neste gráfico os exemplos 0-19 referem-se a uma das classes, os exemplos 20-39 referem-se à segunda classe e os exemplos 40-49 referem-se à terceira classe. É possível observar que, neste experimento, desde as primeiras iterações de treinamento cada um dos três neurônios convergiu em direção ao centro de um grupo.

A figura 8.3b apresenta o gráfico da somatória quadrática dos pesos sinápticos em função da iteração de treinamento. A rede opera claramente na região de decremento dos pesos sinápticos, condição imposta pela configuração inicial dos parâmetros. Esta constatação é confirmada pelo gráfico crescente do tempo de disparo dos neurônios em função da iteração de aprendizado, apresentado na figura 8.3c.

Finalizado o treinamento, diversas instâncias de $\mathbb{R}^3$ foram apresentadas à rede para classificação, e a classe atribuída em cada caso pode ser vista na figura 8.3d,

que fornece uma idéia da partição do espaço $\mathbb{R}^3$ realizada pela RNP.

O desempenho da RNP utilizando a sistemática de aprendizado proposta foi comparado com o desempenho do algoritmo *k-means* clássico. A figura 8.4 apresenta os melhores centros obtidos com a abordagem clássica e a tabela 8.2 apresenta numericamente o valor do Erro Médio Quadrático – *Mean Square Error* (MSE) – definido na equação 4.2 – para diferentes configurações do algoritmo. Os dados apresentados para o *k-means* (neste experimento e nos experimentos subseqüentes) dizem respeito a uma média de 20 treinamentos bem sucedidos. Em cada teste efetuado com o *k-means*, o MSE foi observado a toda iteração de treinamento. O treinamento foi suspenso quando a taxa de variação desse erro assumiu valor negativo. Caracterizou-se uma *falha na convergência* quando algum dos centros não foi atualizado do início até a suspensão do processo de treinamento.

No primeiro caso, o *k-means* foi iniciado com o centro de todas as classes no centro de $\mathbb{R}^3$ e em nenhuma das 20 observações o algoritmo apresentou convergência para os três centros. No segundo caso, iniciando os centros relativamente próximos ao centro de $\mathbb{R}^3$, o algoritmo apresentou convergência em 55% dos casos, e, nesses casos, um MSE médio de 0,0021. No terceiro caso, o *k-means* foi inicializado tomando-se como centros iniciais posições aleatoriamente escolhidas de $\mathbb{R}^3$. O erro obtido equipara-se ao caso anterior, porém com um número maior de convergências. No último caso, o algoritmo foi inicializado tomando-se algumas das instâncias de entrada como centros. O algoritmo apresentou convergência em todos os casos, mas encontrou um número maior de mínimos locais, aumentando o erro médio.

Para o cálculo do MSE da RNP, os centros dos neurônios RBF – estimados segundo a metodologia descrita anteriormente – foram adotados. No caso relatado, a rede apresentou um erro médio de 0,062. É válido observar que: *i*) adequadamente configurada, a rede sempre convergiu; e *ii*) nenhuma sistemática especial de inicialização ou mesmo refinamento do passo de aprendizado foi adotada. Ainda assim, a RNP apresentou um erro médio bastante pequeno, mesmo considerando que existe um erro referente à aproximação dos centros e à discretização do espaço para a estimativa dos centros.

Tabela 8.1: Documentação do experimento EXP01.

Problema	<p>Nome: Grupos globulares 1 Tipo de dado: artificial Dimensões: 3 Classes: 3 Número de exemplos: 60 (20/20/20) Descrição: Três grupos em um espaço $\mathbb{R}^3$: $[0,1]$ com centros em posições aleatórias. Cada grupo possui 20 instâncias aleatoriamente distribuídas, com distribuição uniforme a uma distância máxima do centro de 0,1. Endereço: http://www.sorocaba.unesp.br/professor/assimoes/clustering</p>
Conjuntos	<p>Escala de entrada: $[0,1]$ Escala de saída: - Conjunto de treinamento: todos os pontos Conjunto de validação: - Conjunto de teste: todos os pontos</p>
Arquitetura da rede	<p>Tipo de rede: RNP-SRM-RBF Constantes: $n = 3$; $m = 3$; $d = 250$</p>
Codificação temporal	<p>Tipo de código: latência Metodologia de codificação: campos receptivos gaussianos Constantes: $g = 9$; $h = 100$; $\sigma = 1$</p>
Inicialização	<p>Descrição: centros das classes no centro de $\mathbb{R}^n$, com pequena variação aleatória Constantes: $w_{inicial} = 40 \pm 10\%$</p>
Parâmetros do algoritmo	<p>Constantes: $a = 1$; $b = -0,0232555$; $c = -2$; $w_{max} = 1.497$; $\eta = 10$; $\vartheta = 45.000$; $\tau_m = 1.000$; $\tau_s = 0,1$; $\xi = 1,67$; $\lambda = 4$ Particularidades: sem auto-inibição; λ calculado segundo a equação 6.5; b levado intencionalmente acima do valor calculado para estabilidade segundo equação 6.2</p>
Término	<p>Critério de parada: um dos pesos sinápticos atinge w_{max} ou w_{min} Número de épocas: 430</p>

a) Pontos utilizados no processo de treinamento. Os pontos em azul representam os exemplos de treinamento, e os pontos em preto representam as projeções dos exemplos nos planos laterais; b) Posição média das regiões amostradas com menor tempo de disparo ao longo das iterações de aprendizado. O ponto 'o' representa a posição inicial de cada centro. O ponto 'x' representa o ponto final de cada centro. O número ao longo do deslocamento dos centros representa a iteração de treinamento.

Figura 8.1: Processo de convergência para o EXP-01.

Tempo de disparo dos neurônios da camada de saída para diversas instâncias em $\mathcal{R}^3$ após o processo de treinamento. a) neurônio 1; b) neurônio 2; c) neurônio 3.

Figura 8.2: Tempos de disparo dos neurônios da rede no EXP-01.

a) Exemplos (0-59) sorteados em cada iteração de treinamento (1-430), e o neurônio (1-3) vencedor de cada processo competitivo. Os exemplos dos três grupos encontram-se numerados seqüencialmente (exemplos 0-19, 20-39 e 40-59); b) Somatória quadrática dos pesos sinápticos em função da iteração de treinamento; c) Tempo de disparo em função da iteração de treinamento; d) Partição do espaço $\mathbb{R}^3$.

Figura 8.3: Resultados do EXP-01.

Figura 8.4: Melhores centros obtidos com o algoritmo *k-means* no EXP-01.

Tabela 8.2: Quadro comparativo entre a RNP e o *k-means* no EXP-01.

Algoritmo	MSE	Falha de convergência
k-means com centros no centro de $\mathcal{R}^3$	-	100%
k-means com centros no centro de $\mathcal{R}^3 \pm 10\%$	0,0021	55%
k-means com centros aleatoriamente adotados em $\mathcal{R}^3$	0,0021	50%
k-means tomando observações como centros	0,03638	-
RNP	0,0062	-

8.2 Experimento EXP-02

Para o segundo experimento, a RNP foi treinada como o apresentado na tabela 8.3. A distribuição das instâncias foi feita de forma a gerar nuvens com tamanhos distintos. Os exemplos utilizados no processo de treinamento da rede são apresentados na figura 8.5a. A figura 8.5b ilustra o processo de convergência obtido para os três neurônios – sensivelmente mais sinuoso que o anterior – enquanto a figura 8.6 apresenta os tempos de disparo dos neurônios de saída para diversas instâncias em $\mathbb{R}^3$.

A figura 8.7a exibe os exemplos de treinamento sorteados a cada iteração de aprendizado e o neurônio vencedor do processo competitivo em cada caso. Os exemplos 0-54 referem-se à primeira classe, os exemplos 55-109 referem-se à segunda classe e os exemplos 110-164 referem-se à terceira classe. Ao contrário do caso anterior, é possível notar que a fase de ordenação dos neurônios estendeu-se por quase todo o processo de treinamento, o que explica a acentuada variação no trajeto desses centros.

De maneira análoga ao experimento anterior, as figuras 8.7b, c e d apresentam respectivamente a somatória quadrática dos pesos sinápticos, o tempo de disparo dos neurônios e a partição do espaço $\mathbb{R}^3$.

A figura 8.8 apresenta os melhores centros encontrados utilizando o algoritmo *k-means* clássico. A tabela 8.4 apresenta o erro médio quadrático e o percentual de falha no processo de convergência. Neste experimento, mais complexo que o anterior, a convergência obtida pela RNP – na configuração apresentada – supera a abordagem clássica.

Tabela 8.3: Documentação do experimento EXP02.

Problema	<p>Nome: Grupos globulares 2 Tipo de dado: artificial Dimensões: 3 Classes: 3 Número de exemplos: 165 (55/55/55) Descrição: Três grupos em um espaço $\mathbb{R}^3$: $[0,1]$ com centros em posições aleatórias. Cada grupo possui 55 instâncias uniformemente distribuídas nas proximidades do centro com distância máxima do centro para os grupos 1, 2 e 3 respectivamente de: 0,05 , 0,1 e 0,2 Endereço: http://www.sorocaba.unesp.br/professor/assimoes/clustering</p>
Conjuntos	<p>Escala de entrada: $[0,1]$ Escala de saída: - Conjunto de treinamento: todos os pontos Conjunto de validação: - Conjunto de teste: todos os pontos</p>
Arquitetura da rede	<p>Tipo de rede: RNP-SRM-RBF Constantes: $n = 3$; $m = 3$; $d = 250$</p>
Codificação temporal	<p>Tipo de código: latência Metodologia de codificação: campos receptivos gaussianos Constantes: $g = 9$; $h = 100$; $\sigma = 1$</p>
Inicialização	<p>Descrição: centros das classes no centro de $\mathbb{R}^n$, com pequena variação aleatória Constantes: $w_{inicial} = 40 \pm 10\%$</p>
Parâmetros do algoritmo	<p>Constantes: $a = 1$; $b = -0,0258395$; $c = -2$; $w_{max} = 1.100$; $\eta = 10$; $\vartheta = 39.600$; $\tau_m = 1.000$; $\tau_s = 0,1$; $\xi = 1,67$; $\lambda = 4$ Particularidades: sem auto-inibição; λ calculado segundo a equação 6.5; b levado intencionalmente acima do valor calculado para estabilidade segundo equação 6.2</p>
Término	<p>Critério de parada: um dos pesos sinápticos atinge w_{max} ou w_{min} Número de épocas: 399</p>

a) Pontos utilizados no processo de treinamento. Os pontos em azul representam os exemplos de treinamento, e os pontos em preto representam as projeções dos exemplos nos planos laterais; b) Posição média das regiões amostradas com menor tempo de disparo ao longo das iterações de aprendizado. O ponto 'o' representa a posição inicial de cada centro. O ponto 'x' representa o ponto final de cada centro. O número ao longo do deslocamento dos centros representa a iteração de treinamento.

Figura 8.5: Processo de convergência para o EXP-02.

Tempo de disparo dos neurônios da camada de saída para diversas instâncias em $\mathbb{R}^3$ após o processo de treinamento. a) neurônio 1; b) neurônio 2; c) neurônio 3.

Figura 8.6: Tempos de disparo dos neurônios da rede no EXP-02.

(a)

(b)

(c)

(d)

a) Exemplos (0-164) sorteados em cada iteração de treinamento (1-399), e o neurônio (1-3) vencedor de cada processo competitivo. Os exemplos dos três grupos encontram-se numerados seqüencialmente (exemplos 0-54, 55-109 e 110-164); b) Somatória quadrática dos pesos sinápticos em função da iteração de treinamento; c) Tempo de disparo em função da iteração de treinamento; d) Partição do espaço $\mathbb{R}^3$.

Figura 8.7: Resultados do EXP-02.

Figura 8.8: Melhores centros obtidos com o algoritmo *k-means* para o EXP-02.

Tabela 8.4: Quadro comparativo entre a RNP e o *k-means* no EXP-02.

Algoritmo	MSE	Falha de convergência
k-means com centros no centro de $\mathfrak{R}^3$	-	100%
k-means com centros no centro de $\mathfrak{R}^3 \pm 10\%$	0,0137	35%
k-means com centros aleatoriamente adotados em $\mathfrak{R}^3$	0,0192	40%
k-means tomando observações como centros	0,02312	-
RNP	0,0056	-

8.3 Experimento EXP-03

A documentação do experimento EXP-03 pode ser vista na tabela 8.5. Os cinco grupos utilizados possuem instâncias uniformemente distribuídas em uma região de raio 0,1 ao redor de seus centros, como representado na figura 8.9a. A convergência estimada para os centros dos cinco neurônios pulsados é exibida na figura 8.9b. A figura 8.10 apresenta os tempos de disparo para os cinco neurônios da camada de saída após o treinamento.

A figura 8.11a apresenta os exemplos de treinamento sorteados nas 923 iterações, assim como o neurônio vencedor do processo competitivo em cada caso. Estabelecendo uma analogia como um processo competitivo clássico, pode-se observar que, neste experimento, um processo de ordenação ocorreu aproximadamente até a iteração 200, a partir de onde iniciou-se uma fase de convergência de cada neurônio a um dos grupos.

As figuras 8.11b e 8.11c apresentam respectivamente a somatória quadrática dos pesos sinápticos e o tempo de disparo dos neurônios vencedores dos processos competitivos, demonstrando que a rede realiza o decremento do peso sináptico médio. A figura 8.11d fornece uma estimativa da partição do espaço $\mathbb{R}^3$ realizada pela abordagem.

A figura 8.12 apresenta os melhores centros dos grupos obtidos utilizando o algoritmo clássico *k-means*, e a tabela 8.6 apresenta o quadro comparativo entre as abordagens. Claramente, neste caso, o algoritmo clássico teve maior dificuldade para convergir se comparado aos experimentos anteriores. Isso é evidenciado pelos altos índices de falha na convergência. O valor de erro quadrático médio da RNP foi também inferior à melhor convergência do *k-means*.

Tabela 8.5: Documentação do experimento EXP03.

Problema	<p>Nome: Grupos globulares 3 Tipo de dado: artificial Dimensões: 3 Classes: 5 Número de exemplos: 100 (20/20/20/20/20) Descrição: Cinco grupos em um espaço $\mathbb{R}^3$: $[0, 1]$ com centros em posições aleatórias. Cada grupo possui 20 instâncias aleatoriamente distribuídas, com distância máxima do centro de 0,1 Endereço: http://www.sorocaba.unesp.br/professor/assimoes/clustering</p>
Conjuntos	<p>Escala de entrada: $[0, 1]$ Escala de saída: - Conjunto de treinamento: todos os pontos Conjunto de validação: - Conjunto de teste: todos os pontos</p>
Arquitetura da rede	<p>Tipo de rede: RNP-SRM-RBF Constantes: $n = 3$; $m = 3$; $d = 250$</p>
Codificação temporal	<p>Tipo de código: latência Metodologia de codificação: campos receptivos gaussianos Constantes: $g = 9$; $h = 100$; $\sigma = 1$</p>
Inicialização	<p>Descrição: centros das classes no centro de $\mathbb{R}^n$, com pequena variação aleatória Constantes: $w_{inicial} = 58 \pm 10\%$</p>
Parâmetros do algoritmo	<p>Constantes: $a = 1$; $b = -0,0258395$; $c = -2$; $w_{max} = 1.581$; $\eta = 10$; $\vartheta = 56.925$; $\tau_m = 1.000$; $\tau_s = 0,1$; $\xi = 1,67$; $\lambda = 4$ Particularidades: sem auto-inibição; λ calculado segundo a equação 6.5; b levado intencionalmente acima do valor calculado para estabilidade segundo equação 6.2</p>
Término	<p>Critério de parada: um dos pesos sinápticos atinge w_{max} ou w_{min} Número de épocas: 923</p>

(a)

(b)

a) Pontos utilizados no processo de treinamento. Os pontos em azul representam os exemplos de treinamento, e os pontos em preto representam as projeções dos exemplos nos planos laterais; b) Posição média das regiões amostradas com menor tempo de disparo ao longo das iterações de aprendizado. O ponto 'o' representa a posição inicial de cada centro. O ponto 'x' representa o ponto final de cada centro. O número ao longo do deslocamento dos centros representa a iteração de treinamento.

Figura 8.9: Processo de convergência para o EXP-03.

Tempo de disparo dos neurônios da camada de saída para diversas instâncias em $\mathbb{R}^3$ após o processo de treinamento. a) neurônio 1; b) neurônio 2; c) neurônio 3; d) neurônio 4; e) neurônio 5.

Figura 8.10: Tempo de disparo dos neurônios da rede no EXP-03.

a) Exemplos (0-100) sorteados em cada iteração de treinamento (1-923), e o neurônio (1-5) vencedor de cada processo competitivo. Os exemplos dos cinco grupos encontram-se numerados sequencialmente (exemplos 0-19, 20-39, 40-59, 60-79 e 80-99); b) Somatória quadrática dos pesos sinápticos em função da iteração de treinamento; c) Tempo de disparo em função da iteração de treinamento; d) Partição do espaço $\mathbb{R}^3d$.

Figura 8.11: Resultados do EXP-03.

Figura 8.12: Melhores centros obtidos com o algoritmo *k-means* para o EXP-03.

Tabela 8.6: Quadro comparativo entre a RNP e o *k-means* no EXP03.

Algoritmo	MSE	Falha na convergência
k-means com centros no centro de $\mathfrak{R}^3$	-	100%
k-means com centros no centro de $\mathfrak{R}^3 \pm 10\%$	0,0024	90%
k-means com centros aleatoriamente adotados em $\mathfrak{R}^3$	-	100%
k-means tomando observações como centros	0,0135	5%
RNP	0,0078	-

8.4 Experimento EXP-04

A tabela 8.7 apresenta a documentação do experimento EXP-04. Neste caso, dois grupos entrelaçados não linearmente separáveis – que podem ser vistos na figura 8.13a – foram submetidos à RNP para classificação. O processo de convergência dos centros dos dois neurônios pulsados é apresentado na figura 8.13b, e os tempos de disparo dos dois neurônios para diversas entradas em $\mathcal{R}^3$ após o treinamento são apresentados na figura 8.14.

A figura 8.15a exibe os exemplos sorteados nas iterações de treinamento e o neurônio vencedor de cada processo competitivo. As figuras 8.15b e 8.15c apresentam respectivamente a somatória quadrática dos pesos sinápticos e o tempo de disparo do neurônio vencedor dos processos competitivos.

O gráfico 8.15a permite observar que uma fase de ordenação ocorre aproximadamente até a iteração 30, a partir de quando inicia-se o processo de convergência. Uma estimativa da partição do espaço $\mathcal{R}^3$ na iteração 100 é apresentada na figura 8.15d. É importante observar que neste momento do treinamento, a rede é capaz de classificar corretamente todas as instâncias de entrada. Contudo, prosseguindo com o treinamento, alguns exemplos passam a ser incorretamente classificados pela RNP (exemplos 0–5 vencidos pelo neurônio 1 por volta da iteração 230) antes que a rede atinja seu ponto de parada. A partição do espaço $\mathcal{R}^3$ após a suspensão do aprendizado na iteração 299 é apresentada na figura 8.15e.

A observação das partições em $t=100$ e $t=402$ permite concluir que houve uma simplificação excessiva na função-alvo, já que no segundo caso as classes são separadas quase linearmente. Esta constatação aponta na direção de que o critério de parada suspendeu tardiamente o treinamento da rede, e que, portanto, o critério de parada proposto não se aplica, ao menos a este domínio. É preciso destacar também que nem todos os 534 exemplos das classes foram utilizados nas 268 iterações de treinamento.

Evidentemente, a separação desses grupos não-linearmente separáveis foi possível uma vez que o centro de um grupo não se encontra dentro da nuvem dos dados pertencentes ao outro grupo. A separação de grupos nesta condição de sobreposição não parece ser possível utilizando a RNP apresentada.

Os melhores centros obtidos com da aplicação do algoritmo *k-means* são apresentados na figura 8.16, e a tabela 8.8 mostra os valores de erro médio quadrático para implementações do algoritmo clássico. Para este experimento, a RNP apresenta resultados praticamente equivalentes aos do *k-means*.

Tabela 8.7: Documentação do experimento EXP04.

Problema	<p>Nome: Grupos entrelaçados 1 Tipo de dado: artificial Dimensões: 3 Classes: 2 Número de exemplos: 534 (267/267) Descrição: Dois grupos em um espaço $\mathbb{R}^3 : [0, 1]$, onde as instâncias de cada grupo encontram-se dispostas equidistantes a um centro. A distribuição das instâncias dá-se a uma distância máxima de 0,045 do arco de circunferência Endereço: http://www.sorocaba.unesp.br/professor/assimoes/clustering</p>
Conjuntos	<p>Escala de entrada: $[0, 1]$ Escala de saída: - Conjunto de treinamento: todos os pontos Conjunto de validação: - Conjunto de teste: todos os pontos</p>
Arquitetura da rede	<p>Tipo de rede: RNP-SRM-RBF Constantes: $n = 3$; $m = 3$; $d = 250$</p>
Codificação temporal	<p>Tipo de código: latência Metodologia de codificação: campos receptivos gaussianos Constantes: $g = 9$; $h = 100$; $\sigma = 1$</p>
Inicialização	<p>Descrição: centros das classes no centro de $\mathbb{R}^n$, com pequena variação aleatória Constantes: $w_{inicial} = 58 \pm 10\%$</p>
Parâmetros do algoritmo	<p>Constantes: $a = 1$; $b = -0,0258395$; $c = -2$; $w_{max} = 1.581$; $\eta = 10$; $\vartheta = 56.925$; $\tau_m = 1.000$; $\tau_s = 0, 1$; $\xi = 1,67$; $\lambda = 4$ Particularidades: sem auto-inibição; λ calculado segundo a equação 6.5; b levado intencionalmente acima do valor calculado para estabilidade segundo equação 6.2</p>
Término	<p>Critério de parada: um dos pesos sinápticos atinge w_{max} ou w_{min} Número de épocas: 402</p>

(a)

(b)

a) Pontos utilizados no processo de treinamento. Os pontos em azul representam os exemplos de treinamento, e os pontos em preto representam as projeções dos exemplos nos planos laterais; b) Posição média das regiões amostradas com menor tempo de disparo ao longo das iterações de aprendizado. O ponto 'o' representa a posição inicial de cada centro. O ponto 'x' representa o ponto final de cada centro. O número ao longo do deslocamento dos centros representa a iteração de treinamento.

Figura 8.13: Processo de convergência para o EXP-04.

Tempo de disparo dos neurônios da camada de saída para diversas instâncias em $\mathbb{R}^3$ após o processo de treinamento. a) neurônio 1; b) neurônio 2.

Figura 8.14: Tempos de disparo dos neurônios da rede no EXP-04.

(a)

(b)

(c)

(d)

(e)

a) Exemplos (0-534) sorteados em cada iteração de treinamento (1-402), e o neurônio (1-2) vencedor de cada processo competitivo. Os exemplos dos dois grupos encontram-se numerados sequencialmente (exemplos 0-266 e 267-534); b) Somatória quadrática dos pesos sinápticos em função da iteração de treinamento; c) Tempo de disparo em função da iteração de treinamento; d) Partição do espaço $\mathbb{R}^3$ em $t=100$; e) Partição do espaço $\mathbb{R}^3$ em $t=402$.

Figura 8.15: Resultados do EXP-04.

Figura 8.16: Melhores centros obtidos com o algoritmo *k-means* para o EXP-04.

Tabela 8.8: Quadro comparativo entre a RNP e o *k-means* no EXP04.

Algoritmo	MSE	Falhas na convergência
k-means com centros no centro de $\mathfrak{R}^3$	-	100%
k-means com centros no centro de $\mathfrak{R}^3 \pm 10\%$	0,0595	-
k-means com centros aleatoriamente adotados em $\mathfrak{R}^3$	0,0595	20%
k-means tomando observações como centros	0,0595	-
RNP	0,0608	-

9 Estudo de casos: classificação de cores

Este capítulo apresenta um estudo da aplicabilidade da RNP na tarefa de classificar cores de pixels em imagens coloridas. Nesta investigação, dois experimentos – detalhados a seguir – foram realizados. Alguns novos dados foram observados, além daqueles já observados nos experimentos anteriores:

- i. **Espaço de cores da imagem original:** imagem original transformada para o espaço de cores RGB, isto é, os pixels dessa imagem, um a um, transportados para um espaço em $\mathbb{R}^3$ onde os eixos representam seus valores de R, G e B;
- ii. **Exemplos utilizados:** coleção de pontos utilizada no processo de treinamento da rede apresentada no espaço RGB. A comparação da distribuição dessa nuvem em relação à distribuição das cores no espaço de cores da imagem original fornece uma boa estimativa da representatividade dos pontos adotados como exemplos;
- iii. **Centros obtidos:** gráfico com a posição estimada dos centros obtidos após o treinamento da rede;
- iv. **Imagem classificada:** imagem original com seus pixels classificados – um a um – pela RNP. A classificação de cada um dos pixels da imagem é apresentada na forma de uma nova imagem, onde existem apenas m cores, onde m é o número de classes (grupos).

Os experimentos realizados foram:

- **IMG-01:** Classificação de cores em uma imagem real – onde a técnica da segmentação baseada no atributo cor dos pixels é aplicável – selecionada dentre os *benchmarks* para segmentação de imagens em Berkeley Computer Vision Group (2001);
- **IMG-02:** Classificação de cores na tradicional imagem 'Lena'.

9.1 Experimento IMG-01

A tabela 9.1 apresenta as características principais do experimento IMG-01. A imagem utilizada – imagem 238011 do banco de dados em Berkeley Computer Vision Group (2001) – pode ser vista na figura 9.1a. Ao ser apresentado a esta imagem, um observador humano tipicamente nela identifica três cores: branco (lua), azul (céu) e preto (árvores). Muito embora essas cores sejam facilmente classificadas por humanos, as cores desta imagem distribuem-se de forma bastante complexa pelo espaço de cores RGB, como o apresentado na figura 9.1c.

Para a classificação de cores neste primeiro experimento, alguns pixels foram aleatoriamente extraídos de cada uma dessas três classes e oferecidos à rede para treinamento. O conjunto dos exemplos utilizados é apresentado na figura 9.1d.

Após o treinamento por 325 iterações utilizando os 20 exemplos extraídos de cada classe, o algoritmo atingiu seu critério de parada. A estimativa da posição dos centros das três classes pode ser vista na figura 9.1e. Após este processo, a imagem original (figura 9.1a) foi apresentada à rede (pixel a pixel) e uma nova imagem foi gerada (figura 9.1b), onde cada pixel foi classificado como branco, azul ou preto. Os gráficos do neurônio vencedor de cada processo competitivo, da somatória quadrática dos pesos sinápticos, do tempo de disparo dos neurônios e a partição do espaço RGB podem ser vistos na figura 9.2.

De forma a avaliar quantitativamente o desempenho da rede nesta tarefa, uma imagem foi classificada pixel a pixel por um humano. Esta imagem será tratada como **gabarito** da imagem 238011 no decorrer do texto. O experimento foi então repetido com um número variável de exemplos (1–10) extraídos de cada classe. Uma vez treinada a rede para cada caso e gerada a imagem de resposta, a comparação entre a imagem classificada e o gabarito foi feita utilizando-se a matriz de confusão (seção 4.7) como medida de desempenho. As imagens de resposta e os resultados obtidos são apresentados respectivamente na figura 9.3 na tabela 9.2.

Embora trate-se de uma imagem com baixa complexidade para a classificação de pixels, o desempenho da RNP pode ser considerado satisfatório, mesmo com um número muito baixo de exemplos apresentados à rede.

Tabela 9.1: Documentação do experimento IMG-01.

Problema	<p>Nome: 238011 Tipo de dado: imagem real Dimensões: 3 (r,g,b) Classes: 3 (azul, preto, branco) Número de exemplos: 481x321 pixels Descrição: A imagem possui três classes bem definidas de cores, mas as nuvens de pontos não apresentam separação clara no espaço de cores RGB. O número de pixels não é homogeneamente distribuído entre as três classes. Endereço: http://www.cs.berkeley.edu/projects/vision/grouping/segbench/</p>
Conjuntos	<p>Escala de entrada: $[0, 255]^3$ Escala de saída: - Conjunto de treinamento: 20 pontos aleatoriamente escolhidos de cada classe Conjunto de validação: - Conjunto de teste: todos os pixels da imagem</p>
Arquitetura da rede	<p>Tipo de rede: RNP-SRM-RBF Constantes: $n = 3$; $m = 3$; $d = 250$</p>
Codificação temporal	<p>Tipo de código: latência Metodologia de codificação: campos receptivos gaussianos Constantes: $g = 3$; $h = 100$; $\sigma = 1$</p>
Inicialização	<p>Descrição: centros das classes no centro de $\mathbb{R}^n$, com pequena variação aleatória Constantes: $w_{inicial} = 58 \pm 25\%$</p>
Parâmetros do algoritmo	<p>Constantes: $a = 1$; $b = -0,0258395$; $c = -2$; $w_{max} = 1.812$; $\eta = 10$; $\vartheta = 65.250$; $\tau_m = 1.000$; $\tau_s = 0,1$; $\xi = 1,67$; $\lambda = 4$ Particularidades: sem auto-inibição</p>
Término	<p>Critério de parada: um dos pesos sinápticos atinge w_{max} ou w_{min} Número de épocas: 325</p>

a) Imagem 238011 extraída do banco de dados para segmentação de imagens em Berkeley Computer Vision Group (2001); b) Imagem com os pixels classificados pela RNP após o treinamento utilizando 20 exemplos de cada classe; c) Imagem em 'a' transformada para o espaço de cores RGB. Nesta imagem, o ponto de coordenadas (r, g, b) no espaço RGB é mostrado com a cor (r, g, b) ; d) Exemplos utilizados no processo de treinamento da rede; e) Posição estimada dos centros das classes após o treinamento da RNP.

Figura 9.1: Resultado da classificação de cores no experimento IMG-01.

a) Exemplos (0-59) sorteados nas iterações de treinamento e o neurônio (0-2) vencedor dos processos competitivos; b) Somatória quadrática dos pesos sinápticos; c) Tempo de disparo dos neurônios vencedores dos processos competitivos; d) Situação final do neurônio 1; e) Situação final do neurônio 2; f) Situação final do neurônio 3; g) Partição do espaço RGB.

Figura 9.2: Dados da simulação do experimento IMG-01.

a) Gabarito utilizado na avaliação da acuidade da classificação; Imagem original (figura 9.1) classificada pela RNP treinada com: b) 1 exemplo de cada classe; c) 2 exemplos de cada classe; d) 3 exemplos de cada classe; e) 4 exemplos de cada classe; f) 5 exemplos de cada classe; g) 6 exemplos de cada classe; h) 7 exemplos de cada classe; i) 8 exemplos de cada classe; j) 9 exemplos de cada classe; k) 10 exemplos de cada classe.

Figura 9.3: Imagens classificadas pela RNP no experimento IMG-01.

Tabela 9.2: Avaliação do desempenho da RNP no experimento IMG-01.

Num. ex.	Classe	Matriz confusão ideal			ROC		
		B	A	P	Sens	Espec	DCP
1	B	96.55	3.44	0	9.65e-001	9.99e-001	3.44e-002
	A	0.05	92.36	7.58	9.23e-001	9.98e-001	7.63e-002
	P	0	0.14	99.85	9.98e-001	9.24e-001	7.55e-002
2	B	95.76	4.23	0	9.57e-001	9.99e-001	4.23e-002
	A	0.05	94.19	5.75	9.41e-001	9.95e-001	5.82e-002
	P	0	0.43	99.56	9.95e-001	9.42e-001	5.74e-002
3	B	95.76	4.23	0	9.57e-001	9.99e-001	4.23e-002
	A	0.05	94.02	5.92	9.40e-001	9.96e-001	5.98e-002
	P	0	0.33	99.66	9.96e-001	9.41e-001	5.90e-002
4	B	95.76	4.23	0	9.57e-001	9.99e-001	4.23e-002
	A	0.05	93.44	6.50	9.34e-001	9.96e-001	6.56e-002
	P	0	0.26	99.73	9.97e-001	9.35e-001	6.48e-002
5	B	96.58	3.41	0	9.65e-001	9.99e-001	3.41e-002
	A	0.05	90.65	9.29	9.06e-001	9.98e-001	9.34e-002
	P	0	0.06	99.93	9.99e-001	9.07e-001	9.25e-002
6	B	95.76	3.38	0.84	9.57e-001	9.99e-001	4.23e-002
	A	0.05	93.23	6.71	9.32e-001	9.97e-001	6.77e-002
	P	0	0.24	99.75	9.97e-001	9.33e-001	6.69e-002
7	B	95.76	4.23	0	9.57e-001	9.99e-001	4.23e-002
	A	0.05	93.15	6.78	9.31e-001	9.97e-001	6.84e-002
	P	0	0.23	99.76	9.97e-001	9.32e-001	6.76e-002
8	B	96.58	3.41	0	9.65e-001	9.99e-001	3.41e-002
	A	0.05	92.48	7.46	9.24e-001	9.98e-001	7.51e-002
	P	0	0.15	99.84	9.98e-001	9.25e-001	7.43e-002
9	B	96.58	3.41	0	9.65e-001	9.99e-001	3.41e-002
	A	0.05	89.98	9.96	8.99e-001	9.99e-001	1.00e-001
	P	0	0.03	99.96	9.99e-00	9.00e-001	9.92e-002
10	B	96.58	3.41	0	9.65e-001	9.99e-001	3.41e-002
	A	0.05	92.40	7.54	9.24e-001	9.97e-001	7.60e-002
	P	0	0.16	99.83	9.98e-001	9.24e-001	7.51e-002

9.2 Experimento IMG-02

As constantes utilizadas no experimento **IMG-02** são apresentadas na tabela 9.3. A imagem utilizada neste caso – figura 9.4a – é a foto da modelo sueca Lena Soderberg, bastante difundida na literatura de processamento de imagens e largamente utilizada como *benchmark*. A complexa distribuição dos pixels da imagem no espaço RGB é apresentada na figura 9.4c.

Identificar classes de cores nesta imagem não é uma tarefa trivial, mesmo para humanos. O processo de classificação de cores, neste caso, possui uma estreita relação com a tarefa da quantificação de cores, onde um número reduzido de cores em uma paleta é especificado, e um mapeamento é realizado de forma que cada pixel da imagem seja atribuído a uma cor da paleta. A classificação das cores via RNP, então, pode ser entendida como um processo para a seleção automática das cores desta paleta.

Uma investigação da habilidade da RNP como classificadora de cores neste contexto foi realizada adotando-se $m = 16$ classes. A figura 9.4d apresenta os 576 exemplos utilizados no processo de treinamento da rede aleatoriamente escolhidos dentre os 256x256 pixels da imagem. A figura 9.4e apresenta a estimativa dos centros das classes determinados após a convergência do algoritmo.

Face à impossibilidade prática de avaliar a classificação utilizando uma matriz de confusão, a avaliação do desempenho do modelo foi realizada através do cálculo do erro médio quadrático. O erro foi comparado com o obtido com a abordagem clássica do *k-means* e com outras abordagens referenciadas na literatura nas mesmas condições sobre o mesmo *benchmark*, como: (YEO, 2005) e (SIRISATHITKUL; AUWATANAMONGKOL; UYYANONVARA, 2004). Embora o erro médio quadrático obtido neste caso seja alto, a abordagem clássica apresenta grande número de falhas no processo de convergência. Deve-se destacar, novamente, que esta avaliação é feita sobre a posição estimada do centro dos neurônios pulsados.

Tabela 9.3: Documentação do experimento IMG-02.

Problema	<p>Nome: Lena Tipo de dado: imagem real Dimensões: 3 (r,g,b) Classes: não definido. Utilizando 16 Número de exemplos: 576 pixels Descrição: Imagem da modelo sueca Lena Soderberg originalmente publicada em 1972. Possivelmente a figura mais utilizada para avaliações da área de processamento de imagens. Endereço: http://www.ee.cityu.edu.hk/lmpo/lenna/Lenna97.html</p>
Conjuntos	<p>Escala de entrada: $[0, 255]^3$ Escala de saída: - Conjunto de treinamento: 576 pontos aleatoriamente escolhidos da imagem Conjunto de validação: - Conjunto de teste: todos os pixels da imagem</p>
Arquitetura da rede	<p>Tipo de rede: RNP-SRM-RBF Constantes: $n = 3$; $m = 16$; $d = 250$</p>
Codificação temporal	<p>Tipo de código: latência Metodologia de codificação: campos receptivos gaussianos Constantes: $g = 3$; $h = 100$; $\sigma = 1$</p>
Inicialização	<p>Descrição: centros das classes no centro de $\mathbb{R}^n$, com pequena variação aleatória Constantes: $w_{inicial} = 40 \pm 10\%$</p>
Parâmetros do algoritmo	<p>Constantes: $a = 1$; $b = -0,0177591$; $c = -2$; $w_{max} = 1.100$; $\eta = 10$; $\vartheta = 39.600$; $\tau_m = 1.000$; $\tau_s = 0, 1$; $\xi = 1, 67$; $\lambda = 4$ Particularidades: sem auto-inibição</p>
Término	<p>Critério de parada: um dos pesos sinápticos atinge w_{max} ou w_{min} Número de épocas: 1219</p>

a) Imagem original de 'Lena'; b) Imagem em 'a' classificada pela RNP utilizando 16 classes. Nesta imagem, cada pixel foi impresso com a cor do centro mais próximo; c) Espaço de cores da imagem em 'a'; d) Exemplos aleatoriamente extraídos da imagem original utilizados no processo de treinamento da RNP; e) Posição estimada dos centros das classes após o treinamento. A cor (r, g, b) de cada centro foi impressa em acordo com a posição (r, g, b) dos mesmos no espaço RGB.

Figura 9.4: Resultado da classificação de cores no experimento EXP-02.

Tabela 9.4: Avaliação do desempenho da RNP no experimento IMG-02.

Algoritmo	MSE	Falha de convergência
k-means inicializado no centro	-	100%
k-means inicializado no centro $\pm 10\%$	-	100%
k-means inicializado com pontos aleatórios	-	100%
k-means inicializado com observações	277,48	-
SmART (YEO, 2005)	256,97	-
CQA (SIRISATHITKUL; AUWATANAMONGKOL; UYYANONVARA, 2004)	304,00	-
RNP	409,09	-

10 Conclusão

“No domínio dos problemas de cálculo, coordenadas cartesianas e polares representam dois diferentes paradigmas para ver e manipular o problema. Ambos produzem resposta correta, mas um exige menos trabalho para certos problemas.” (MURPHY, 2000, p. 5).

Este trabalho realizou em sua parte I uma revisão dos conceitos relacionados à coleção de novas abstrações para neurônios biológicos denominada *redes neurais pulsadas*, com ênfase no modelo de rede neural pulsada de base radial, particularmente quando os dados de entrada são codificados utilizando campos receptivos gaussianos. Em sua parte II, este trabalho propôs uma sistemática de calibração de alguns de seus parâmetros-chave (b , λ , ϑ) baseada em uma nova função para o aprendizado não-supervisionado dessa rede. Finalmente, na parte III a rede e a metodologia proposta foram avaliadas em domínios distintos.

Os estudos realizados – com a completa documentação do processo de convergência da rede – comprovam que a metodologia proposta é eficaz no processo de calibração da rede. Os resultados obtidos corroboram os primeiros resultados da aplicação deste modelo de rede obtidos em Natschläger e Ruf (1998) e Bohte (2003), e atestam – de forma documentada – que as RNP-RBFs constituem um paradigma viável para a classificação de padrões. A nova ótica de abordagem do modelo de rede e algoritmo de aprendizado não-supervisionado aponta para um sistema de aprendizado estreitamente relacionado com o bem conhecido método *k-means*, porém com novas possibilidades que podem vir a garantir maior robustez a mínimos locais, principal problema da abordagem clássica. Dentre essas novas possibilidades, destacam-se a existência de informação local que permite ao neurônio identificar convergência para mínimos locais e a utilização da codificação temporal da informação, cujos limites e benefícios não se encontram plenamente estabelecidos.

10.1 Sumário das contribuições

Inicialmente, por se tratar de uma temática emergente, esforços foram feitos na formulação do texto para abordar as RNPs de uma forma unificada, muito embora este conjunto de modelos apresente-se disperso na literatura. Em um segundo momento, esforços foram feitos no sentido de trazer à tona o funcionamento do modelo RNP-SRM-RBF em uma abordagem mais próxima do aprendizado de máquinas, relegando os aspectos de cunho biológico a um segundo plano. Finalmente, proposições no sentido de estender o aprendizado da rede foram formuladas.

As principais contribuições deste trabalho são, então, identificadas como:

- Abordagem da RNP-SRM-RBF no contexto do aprendizado de máquinas
 - i. Abordagem unificada dos modelos de neurônios pulsados (seção 2.3);
 - ii. Abordagem unificada dos processos de aprendizado para RNP-SRM-RBFs (capítulo 3);
 - iii. Proposta de uma taxonomia para os diversos modelos de neurônios para RNPs (figura 2.2);
 - iv. Proposta de posicionamento da RNP no contexto do aprendizado baseado em instâncias (tabela 5.1);
- Elucidação da sistemática de funcionamento da RNP-SRM-RBF
 - i. Estabelecimento da não-equivalência entre os modelos de Hopfield (1995) e Natschläger e Ruf (1998) (seção 5.3);
 - ii. Estudo sobre a influência dos campos receptivos gaussianos no processamento da rede (seção 5.2);
 - iii. Estabelecimento da influência dos campos receptivos gaussianos na separação dos dados de entrada (figura 5.1);
 - iv. Estabelecimento da influência dos campos receptivos gaussianos no centro inicial das classes dos neurônios (figura 5.3);
 - v. Estabelecimento da influência dos campos receptivos gaussianos na generalização do NP (figura 5.3);
 - vi. Proposta de alteração na posição inicial dos centros dos neurônios via alteração nas gaussianas de entrada (figura 5.3);
 - vii. Observação das relevantes implicações do parâmetro h na sistemática de funcionamento da rede (figura 5.4);

- Proposição de novos caminhos para o aprendizado da RNP-SRM-RBF
 - i. Maior investigação do potencial de aplicação do SRM no modelo de rede adotado;
 - ii. Proposição do reescalonamento sináptico como forma de estimular o processo competitivo da rede (figura 5.11 e seção 6.3);
 - iii. Proposição da função de aprendizado integrável $\tanh$ (seção 6.1);
 - iv. Proposição da equação para estimar b utilizando a função de aprendizado $\tanh$, reduzindo a necessidade de testes empíricos para sua calibração (equação 6.2);
 - v. Proposição do uso da informação de W para a melhoria no processo de convergência do neurônio (seção 6.6);
 - vi. Proposição de uma equação para estimar ϑ utilizando a função de aprendizado $\tanh$, reduzindo a necessidade de testes empíricos para sua calibração (equação 6.4);
 - vii. Proposição de uma equação para estimar λ utilizando a função de aprendizado $\tanh$, reduzindo a necessidade de testes empíricos para sua calibração (equação 6.5);

10.2 Direções futuras

Como trabalhos futuros inspirados por este, destacam-se:

- i. Investigar o poder de classificação de cores de pixels da RNP em outros espaços de cores, comparando o classificador com abordagens clássicas;
- ii. Avaliar a viabilidade da utilização simultânea de campos receptivos gaussianos com configurações distintas, como o proposto por Bohte, Kok e la Poutre (2000);
- iii. Estudar com maior profundidade a possibilidade de implementar esta classe de rede em *hardware*, dado que a forma pulsada de representar informação demanda menor consumo de energia;
- iv. Investigar novas funções heurísticas e a viabilidade de sua aplicação no auxílio do processo de convergência;
- v. Investigar algoritmos que possam explorar formas híbridas de trabalho com o SRM, já que este neurônio parece incorporar algumas das características de portas sigmoidais e neurônios RBF, entre outros;

- vi. Avaliar a robustez da RNP-RBF com a nova sistemática de calibração na tarefa da classificação de objetos em imagens.

De uma forma geral, parece haver um consenso de que a abordagem pulsada possui informações que não estão presentes em redes neurais estáticas. No entanto, ainda há muito o que explorar sobre as RNPs. Os estudos realizados aqui – no que tange à codificação, convergência e aprendizado da RNP – são ainda introdutórios, e existe uma clara demanda por estudos sistematizados que possam estabelecer seus domínios de aplicação mais adequados.

Referências Bibliográficas

- ABE, K. Spatio-temporal dynamics of large scale neural networks of localized connectivity. In: WANG, L. et al. (Ed.). *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02)*. Cingapura: IEEE Press, 2002. v. 4, p. 1689–1693.
- AMEMORI, K.; ISHII, S. Self-organization network learning of sub-millisecond temporal coded information. In: *Proceedings of the 5th International Conference on Neural Information Processing (ICONIP'98)*. Kitakyusyu, Japan: IOS Press, 1998. v. 3, p. 1285–1288.
- BATISTA, G. E. A. P. A. *Pré-processamento de dados em aprendizado de máquina supervisionado*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação. Universidade de São Paulo, São Carlos, 2003.
- BEAR, M. F.; CONNORS, B. W.; PARADISO, M. A. *Neuroscience: exploring the brain*. 3rd ed. USA: Lippincott Williams & Wilkins, 2006.
- Berkeley Computer Vision Group. *Berkeley Segmentation Dataset*. 2001. <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>. Acesso em 25/01/2006.
- BERRY, M. J. A.; LINOFF, G. *Data mining techniques for marketing, sales and costomer support*. USA: John Wiley & Songs Inc., 1996. Apud (HALKIDI; BATISTAKIS; VARZIRGIANNIS, 2001).
- BLOODSHED SOFTWARE. fevereiro 2005. <http://www.bloodshed.net>. Acesso em 25/01/2006.
- BOHTE, S. M. *Spiking Neural Networks*. Tese (Doutorado) — Institute for Programming Research and Algorithmics. Centre for Mathematics and Computer Science, Amsterdam, 2003.
- BOHTE, S. M.; KOK, J. N.; la POUTRÉ, H. Unsupervised classification of complex clusters in networks of spiking neurons. *IEEE Transactions on Neural Networks*, v. 13, n. 2, p. 426–435, march 2000.
- BOHTE, S. M.; KOK, J. N.; la POUTRÉ, H. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing. Elsevier*, n. 48, p. 17–37, 2002.
- BOHTE, S. M.; KOK, J. N.; la POUTRÉ, H. Implementing position-invariant detection of feature-conjunctions in a network of spiking neurons. *Neural Networks*, v. 2, p. 1097–1102, 2002.
- BOHTE, S. M.; la POUTRÉ, H.; KOK, J. N. Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer rbf networks. *IEEE Transactions on Neural Networks*, v. 13, n. 2, p. 426–435, march 2002.

- BOOIJ, O. *Temporal pattern classification using spiking neural networks*. Dissertação (Mestrado) — Informatics Institute. Faculty of Science. University of Amsterdam, 2004.
- BOSH, H.; LABBI, A.; MILANESE, R. Object segmentation by attention-induced oscillations. In: *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI '98)*. Anchorage, Alaska, USA: IEEE Press, 1998.
- CABRAL, E. F. *Redes Neurais Artificiais - Um curso prático para engenheiros e cientistas*. São Paulo: Edição dos autores, 1999.
- CAJAL, S. R. *Die Retine der Wirbeltiere*. Weisbaden: J. F. Bergman, 1894. Apud (KOVÁCS, 1996).
- CAMPBELL, S. R. *Synchrony and desynchrony in neural oscillators*. Tese (Doutorado) — School of the Ohio State University, 1997.
- CARPENDER, G. A.; GROSSBERG, S. Art2: stable self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, n. 26, p. 4919–4930, 1987b.
- CARPENTER, G. A.; GROSSBERG, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, n. 37, p. 54–115, 1987a.
- CARPENTER, G. A.; GROSSBERG, S. Art3: hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, n. 3, p. 129–152, 1990.
- CHOE, Y.; MIIKKULAINEN, R. Self-organization and segmentation with laterally connected spiking neurons. *Neurocomputing*, n. 21, p. 139–157, 1998.
- CHURCH, K. et al. Using statistics in lexical analysis. In: ZERNIK, U. (Ed.). *Lexical acquisition: Exploiting on-line resources to build a lexicon*. New Jersey: Lawrence Erlbaum, 1991. p. 115–164.
- COVER, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on electronic Computers*, EC-14, p. 326–334, 1965.
- DELORME, A.; THORPE, S. J. Face identification using one spike per neuron: resistance to image degradations. *Neural Networks*, n. 14, p. 795–803, 2001.
- DUDA, R. O.; HART, P. E. *Pattern classification and scene analysis*. New York: Wiley, 1973.
- ECKHORN, R. et al. Feature linking via synchronization among distributed assemblies: simulation of results from cat cortex. *Neural Computation*, v. 2, p. 293–307, 1990.
- EDELMAN, S.; DUVDEVANI-BAR, S. Similarity, connectionism, and the problem of representation in vision. *Neural Computation*, n. 9, p. 701–721, 1997.

FIX, E.; HODGES, J. L. Discriminatory analysis - nonparametric discrimination: consistency properties. Report n. 4, US Air force School of Aviation Medicine, Random Field, Texas. 1951.

FORGY, E. W. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, n. 21, p. 768–769, 1965.

GERSTNER, W. Associative memory in a network of biological neurons. *Advances in Neural Information Processing Systems*, v. 3, p. 84–90, 1991.

GERSTNER, W. Time structure of the activity in neural network models. *Physical Review E*, v. 51, p. 738–758, 1995.

GERSTNER, W. et al. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, n. 384, p. 76–78, september 1996.

GONZALES, R. C.; WOODS, R. E. *Digital Image Processing*. Massachusetts, USA: Addison-Wesley Publishing Company Inc., 1992.

GREGORY, R. L. *Eye and Brain - The psychology of seeing*. 4th ed. New York: Oxford University Press Inc., 1990.

GUHA, S.; RASTOGI, R.; SHIM, K. Cure: An efficient clustering algorithm for large databases. In: *Proceedings of the ACM Sigmod International Conference on Management of Data*. Seattle, WA, USA: ACM Press, 1998. p. 73–84. Apud (HALKIDI; BATISTAKIS; VARZIRGIANNIS, 2001).

HALKIDI, M.; BATISTAKIS, Y.; VARZIRGIANNIS, M. On clustering validation techniques. *Journal of Intelligent Information Systems*, v. 17, n. 2/3, p. 107–145, 2001.

HALL, D. J.; BALL, G. B. *ISODATA: a novel method of data analysis and pattern classification*. Stanford Research Institute. Menlo Park, CA, 1965.

HAYKIN, S. *Neural Networks: a comprehensive foundation*. 2nd ed. Upper Saddle River: Prentice Hall, 1999.

HEBB, D. O. *The organization of the behavior: a neuropsychological theory*. New York: Willey, 1949.

HODGKIN, A. L.; HUXLEY, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal Physiology*, v. 117, p. 500–544, 1952.

HOPFIELD, J. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, v. 376, p. 33–36, 1995.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, n. 72, 1982.

IZHIKEVICH, E. M. Weakly pulse-coupled oscillators, fm interactions, synchronization, and oscillatory associative memory. *IEEE Transactions on Neural Networks*, v. 10, n. 3, p. 508–526, 1999.

- IZHIKEVICH, E. M. Ressonate-and-fire neurons. *Neural networks*, n. 14, p. 883–894, 2001.
- JANCEY, R. C. Multidimensional group analysis. *Australian Journal of Botany*, n. 14, p. 127–130, 1966.
- JOHNSON, J. L. Pulse-coupled neural nets: translation, rotation, scale, distortion, and intensity signal invariance for images. *Applied Optics*, v. 33, n. 26, p. 6239–6253, september 1994.
- JOHNSON, J. L.; PADGETT, M. L. Pcn models and applications. *IEEE Transactions on Neural Networks*, v. 10, n. 3, p. 480–498, 1999.
- KATADA, Y.; OHKUDA, K. Artificial evolution of pulsed neural networks on the motion pattern classification system. In: *IEEE Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation*. Kobe, Japan: IEEE, 2003. p. 318–323.
- KINSER, J. M. Pulse-coupled image fusion. *Optical Engineering. Society of Photo-Optical Instrumentation Engineering*, v. 36, n. 3, p. 737–742, 1997.
- KISTLER, W.; GERSTNER, W.; HEMMEN, J. L. V. Reduction of Hodgkin-Huxley equations to a single-variable threshold model. *Neural Computation*, n. 9, p. 1015–1045, 1997.
- KOHONEN, T. Correlation matrix memories. *IEEE Transactions on Computers*, n. 21, 1972.
- KOHONEN, T. Analysis of a simple self-organizing process. *Biological Cybernetics*, n. 43, p. 135–140, 1982.
- KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, n. 43, p. 59–69, 1982.
- KOHONEN, T. *Learning vector quantization for pattern recognition*. Laboratory of Computer and Information Science, 1986. Technical Report - TKK-F-A-601.
- KOHONEN, T. Learning vector quantization. *Neural Networks*, v. 1, n. suppl. 1, p. 303, 1988.
- KOHONEN, T.; BARNA, G.; CHRISLEY, R. Statistical pattern recognition with neural networks: benchmarking studies. In: *Proc. IEEE Conf. on Neural Networks*. San Diego: IEEE Press, 1988. v. 1, p. 61–68.
- KOVÁCS, Z. L. *Redes Neurais Artificiais - Fundamentos e Aplicações*. 2nd ed. São Paulo: Collegium Cognito, 1996.
- KULLBACK, S. *Information theory and statistics*. New York: Willey, 1959.
- KUROYANAGI, S.; IWATA, A. An competitive learning pulsed neural network for temporal signals. In: WANG, L. et al. (Ed.). *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02)*. Cingapura: IEEE Press, 2002. v. 1, p. 348–352.
- LAPICQUE, L. Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *Journal Physiol. Pathol. Gen.*, n. 9, p. 620–635, 1907.

- MAASS, W. Lower bounds for the computational power of networks of spiking neurons. *Neural Computation*, v. 8, p. 1–40, 1996.
- MAASS, W. Networks of spiking neurons: the third generation of neural networks. *Neural Networks*, v. 10, p. 1659–1671, 1997.
- MAASS, W.; BISHOP, C. M. (Ed.). *Pulsed Neural Networks*. Cambridge: MIT Press, 1999.
- MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: LECAM, L. M.; NEYMAN, J. (Ed.). *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA: University of California press, 1967. v. 1, p. 281–297.
- MALSBURG, C. von Der; SCHNEIDER, W. A neural cocktail party processor. *Biological Cybernetics*, v. 54, p. 29–40, 1986.
- MARR, D. *Vision*. San Francisco, USA: Freeman publishers, 1982.
- MARTIN, D. et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of 8th International Conference on Computer Vision*. Vancouver, Canada: IEEE, 2001. v. 2, p. 416–423.
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, p. 115–133, 1943.
- MILNER, P. M. A model for visual shape recognition. *Psychology Review*, v. 81, p. 521–535, 1974.
- MITCHELL, T. M. *Machine Learning*. Boston: McGraw-Hill, 1997.
- MOORE, A. W. *K-means and hierarchical clustering*. 2001. School of Computer Science. Carnegie Mellon University. Tutorial. Disponível em <http://www.cs.cmu.edu/~awm/tutorials>. Acesso em 05/06/2005.
- MOORE, S. C. *Back-propagation in spiking neural networks*. Dissertação (Mestrado) — University of Bath, 2002.
- MURESAN, R. Visual scale independence in a network of spiking neurons. In: WANG, L. et al. (Ed.). Cingapura: Proceedings of the 9th International Conference on Neural Information, 2002. v. 4.
- MURPHY, R. R. *Introduction to AI robotics*. Massachusetts, USA: [s.n.], 2000.
- NAKANO, H.; SAITO, T. Grouping synchronization in a pulse-coupled network of chaotic spiking oscillators. *IEEE Transactions on Neural Networks*, v. 15, n. 5, p. 1018–1026, 2004.
- NATSCHLÄGER, T.; RUF, B. Spatial and temporal pattern analysis via spiking neurons. *Network*, v. 9, n. 3, p. 319–332, 1998.
- PARK, J.; SANDBERG, W. Universal approximations using radial basis function. *Neural Computation*, n. 3, 1991.

- PRECHELT, L. *PROBEN1 – A set of neural network benchmark problems and benchmarking rules*. Technical Report 21/94. University of Karlsruhe. Germany, september 1994.
- RICH, E.; KNIGHT, K. *Inteligência Artificial*. 2ª ed. São Paulo: Makron Books, 1993.
- RIPLEY, B. D. *Pattern recognition and neural networks*. Cambridge: Cambridge University Press, 1996.
- ROSENBLATT, F. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Washington, DC, USA: Spartan Books, 1962. Apud (HAYKIN, 1999).
- RUF, B. *Computing and learning with spiking neurons: theory and simulations*. Tese (Doutorado) — Institute of Theoretical Computer Science, Graz, Austria, 1998.
- RUF, B.; SCHMITT, M. Self-organization of spiking neurons using action potential timing. *IEEE Transactions on Neural Networks*, v. 9, n. 3, p. 575–578, 1998.
- RULLEN, R. V.; THORPE, S. J. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Computation*, n. 13, p. 1255–1283, 2001.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. *Parallel Distributed Processing*. MIT Press, p. 318–362, 1986.
- RUSSEL, S.; NORVIG, P. *Artificial Intelligence*. Upper Saddle River, New Jersey: Prentice Hall, 1995. (Series in Artificial Intelligence).
- SAPHIRO, J. *Unsupervised Learning: Competitive Learning*. february 2005. School of Computer Science. University of Manchester. Disponível em: <http://www.cs.man.ac.uk/~jls/>. Acesso em 04/06/2005.
- SCHÜRMANN, J. *Pattern classification: a unified view of statistical and neural approaches*. New York: Wiley Interscience, 1996.
- SIMÕES, A. S. *Segmentação de imagens por classificação de cores: uma abordagem neural*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2000.
- SIMÕES, A. S.; COSTA, A. H. R. Classificação de cores por redes neurais artificiais usando a representação hsv. In: MONARD, M. C.; SICHMAN, J. S. (Ed.). *International Joint Conference IBERAMIA '2000 and SBIA '2000. Open Discussion Track Proceedings*. Atibaia, SP: ICMC/USP, 2000. p. 314–323.
- SIMÕES, A. S.; COSTA, A. H. R. Segmentação de imagens por classificação de cores: uma abordagem neural para a representação rgb. In: RIBEIRO, C. H. C.; SAKUDE, M. T. (Ed.). *Workshop de Computação - WORKCOMP'2000*. Instituto Tecnológico de Aeronáutica - ITA. São José dos Campos, SP: Sociedade Brasileira de Computação - SBC, 2000. p. 25–31.

- SIMÕES, A. S.; COSTA, A. H. R. Using neural color classification in robotic soccer domain. In: BARROS, L. N. et al. (Ed.). *International Joint Conference IBERAMIA'2000 and SBIA'2000. Workshop proceedings: meeting on multi-agent collaborative and adversarial perception, planning, execution and learning*. Atibaia, SP: ICMC/USP, 2000. p. 208–213.
- SIMÕES, A. S.; COSTA, A. H. R. *Segmentação de imagens por classificação de cores: uma abordagem neural*. Boletim técnico do departamento de engenharia de computação e sistemas digitais - PCS. Escola Politécnica da Universidade de São Paulo - EPUSP. BT/PCS/0113. São Paulo, SP, 2001.
- SIMÕES, A. S.; COSTA, A. H. R. Classificação de laranjas baseada em padrões visuais. In: *Anais do VI Simpósio Brasileiro de Automação Inteligente - SBAI'2003*. Bauru, SP: Sociedade brasileira de automática - SBA, 2003.
- SIMÕES, A. S. et al. Applying neural networks to automated visual fruit sorting. In: ZAZUETA, F.; XIN, J. (Ed.). *Proceedings of the World Congress of Computers in Agriculture and Natural Resources - WCCA'2002*. Iguaçu Falls: American Society of Agricultural Engineers - ASAE publication 701P0301, 2002. p. 1–7.
- SIRISATHITKUL, Y.; AUWATANAMONGKOL, S.; UYYANONVARA, B. Color image quantization using distances between adjacent colors along the color axis with highest color variance. In: *Pattern recognition letters*. [S.l.]: Elsevier, 2004. p. 1025–1043.
- SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: an introduction*. USA: MIT Press, 1998.
- SWETS, J. A. Measuring the accuracy of diagnostic systems. *Science*, v. 240, p. 1285–1293, 1988.
- TAKITA, K.; HAGIWARA, M. Pulse neural network learning algorithm for POMDP environment. In: *Proceedings of the 2002 International Joint Conference on Neural Networks IJCNN'02. Volume 2*. Honolulu, HI: IEEE Press, 2002. p. 1643–1648.
- TAKITA, K.; HAGIWARA, M. A pulse network reinforcement learning algorithm for partially observable Markov decision process. *Systems and computers in Japan. Wiley Periodicals, Inc.*, v. 36, n. 3, p. 42–52, 2005.
- TAKITA, K.; OSANA, Y.; HAGIWARA, M. Reinforcement learning algorithm with network extension for pulse neural network. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. Nashville, TN, USA: IEEE Press, 2000. p. 2586–2591.
- THE MATHWORKS INC. 2002. <http://www.mathworks.com/>. Acesso em 25/01/2006.
- THEODORIDIS, Y.; KOUTROUBAS, K. *Pattern Recognition*. San Diego, CA, USA: Academic Press, 1999. Apud (HALKIDI; BATISTAKIS; VARZIRGIANNIS, 2001).
- TIMOSCZCZUK, A. P. *Reconhecimento automático do locutor com redes neurais pulsadas*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

- TORIKAI, H.; SAITO, T. Synchronization phenomena in pulse-coupled networks driven by spike-time inputs. *IEEE Transactions on Neural Networks*, v. 15, p. 337–347, 2004.
- TSENG, V. S.-M.; KAO, C.-P. An efficient approach to identifying and validating clusters in multivariate datasets with applications in gene expression analysis. *Journal of Information Science and Engineering*, n. 20, p. 665–667, 2004.
- VINOD, H. Integer programming and the theory of grouping. *Journal of the American Statistical Association*, n. 64, p. 506–517, 1969. Apud (RIPLEY, 1996).
- WALLIS, G.; BADDELEY, R. Optimal, unsupervised learning in invariant object recognition. *Neural Computation*, n. 9, p. 883–894, 1997.
- WANG, D. L.; TERMAN, D. Locally excitatory globally inhibitory oscillator networks. *IEEE Transactions on Neural Networks*, v. 6, n. 1, p. 283–286, 1995.
- WANG, R. A hybrid learning network for shift-invariant recognition. *Neural networks*, n. 14, p. 1061–1073, 2001.
- WATKINS, C. J.; DAYAN, P. Technical note: Q-learning. *Machine Learning*, n. 8, p. 279–292, 1992.
- WIDROW, B. Generalization and information storage in networks of adaline neurons. *Self-organizing Systems*, 1962.
- WINSTON, P. H. *Artificial Intelligence*. Reading, MA, USA: Addison-Wesley Longman Publishing Co., 1992.
- WYSS, R.; KÖNIG, P.; VERSHURE, P. F. M. J. Invariant representations of visual pattern in a temporal population code. *Proceedings of the National Academy of Sciences*, v. 100, n. 1, p. 324–329, january 2003.
- YEO, N. C. et al. Colour image segmentation using the self-organizing map and adaptative resonance theory. In: *Image and vision computing*. [S.l.]: Elsevier, 2005. p. 1060–1079.
- YU, B.; ZHANG, L. Pulse-coupled neural networks for contour and motion matching. *IEEE Transactions on Neural Networks*, v. 15, p. 1186–1201, 2004.
- ZHANG, T.; RAMAKRISHNMAN, R.; LINVY, M. Birch: An efficient method for very large databases. In: *ACM Sigmod*. Montreal, Canada: ACM Press, 1996. Apud (HALKIDI; BATISTAKIS; VARZIRGIANNIS, 2001).

Índice Remissivo

- agrupamento, 55
- antecipação do disparo, 45
- aprendizado baseado em instâncias, 54
- aprendizado de máquinas, 54
- aprendizado em RNPs, 36
- aprendizado não supervisionado, 44
- aprendizado por reforço, 49
- aprendizado proposto, 93
- aprendizado supervisionado, 37
- auto-inibição, 49
- códigos temporais, 10
- campos receptivos gaussianos, 30
- classificação, 55
- classificação de cores, 136
- codebook, 60
- compactividade, 65
- conclusão, 147
- convergência, 110
- desdobramento de sinapses, 29
- deslocamento vetor de pesos, 47
- espalhamento dos pesos, 47
- estabilidade na soma dos pesos sinápticos, 81
- fase, 12
- funções base radial, 26
- histórico RNPs, 12
- Hodgkin-Huxley, 17
- integra-e-dispara, 19
- introdução, 1
- k-means, 57
- k-vizinho mais próximo, 56
- latência, 12, 16
- lei de Hebb, 37
- materiais e métodos, 106
- matriz confusão, 66
- McCulloch-Pitts, 24
- mean square error, 112
- modelos de neurônios, 16
- neurônios da camada de saída, 40
- neurônios das demais camadas, 42
- objetivos, 5
- padrão, 55
- padrão dos pesos sinápticos, 47
- poder computacional do SRM, 24
- portas sigmoidais, 25
- potencial, 20
- potencial de ação, 20
- proben1, 108
- RBFs, 26
- redes neurais pulsadas, 9
- retropropagação do erro, 39
- revisitando RNPs, 70
- RNP-SRM-RBF, 28
- RNPs, 27
- ROC, 67
- separação, 66
- sincronia, 12
- Spikeprop, 39
- spiking response model, 20
- SRM, 20
- taxa de aprendizado, 38
- taxa média de disparos, 10
- teorema de cover, 63
- visão computacional, 1
- w_{max} , 48