

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

273192

**Aplicação de Sistemas
Híbridos em Problemas
de Otimização**

por
ANTÔNIO ROGÉRIO MACHADO RAMOS

Dissertação submetida à avaliação como requisito parcial para
a obtenção do grau de Mestre em
Ciência da Computação



Prof. Dr. Dante Augusto Couto Barone
Orientador

Porto Alegre, setembro de 1996.

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Ramos, Antônio Rogério Machado

Aplicações de sistemas híbridos em problemas de otimização/ por Antônio Rogério Machado Ramos. Porto Alegre: CPGCC da UFRGS, 1996.

101p.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1996. Orientador: Barone, Dante Augusto Couto.

1. Reconhecedores óticos de caracteres. 2. Redes neurais. 3. Algoritmos genéticos. 4. Hibridização. I. Barone, Dante Augusto Couto. II. Título

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA		
N.º CHAMADA 681.327 162 (043) R175a	N.º REG.º 37661	
ORIGEM: D	DATA: 05/05/00	30/08/00
FUNDO: II	FORN.: II	R\$ 30,00

Aplicações dos computadores - SB

Reconhecimento:
Padrões

Redes neurais

Algoritmos genéticos

Reconhecimento: caracteres óticos

CNPq 1.03.04.00-2

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Hélgio Casses Trindade

Pró-Reitor de Pesquisa e Pós-Graduação: Prof. Cláudio Scherer

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. Flávio Wagner

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

Agradecimentos

Agradeço ao meu grande amigo Fernando Osório, por ter me apresentado a Universidade Federal do Rio Grande do Sul, possibilitando meu contato com a pesquisa dentro de uma área que muito me fascina que é sistemas adaptativos. Agradeço também ao meu colega de mestrado Ivonei Marques pelo auxílio prestado na confecção das figuras que auxiliam no entendimento deste trabalho.

Sou grato também a meu orientador Prof. Dr. Dante Barone por ter apostado neste aluno, incentivando e proporcionando a realização das pesquisas que resultaram neste trabalho.

Meu agradecimento todo especial para minha noiva, Eunice, por toda ajuda proporcionada no desenvolvimento desta dissertação e, principalmente, no meu desenvolvimento como ser humano.

Finalmente, agradeço a Deus por me encontrar na época e nos lugares certos, onde posso vislumbrar todo este dinamismo tecnológico que me motiva a pesquisar cada vez mais e também aos meus pais Vanderlei e Sirlei Ramos pelo apoio e incentivo dados não apenas no decorrer do mestrado mas em toda a minha vida.

Sumário

Lista de Abreviaturas	6
Lista de Fórmulas	7
Lista de Tabelas	8
Lista de Figuras	10
Resumo	12
Abstract	13
1 Introdução	14
2 Reconhecimento Ótico de Caracteres	16
2.1 Etapas no Desenvolvimento de Um OCR	16
2.1.1 Definição do tipo de caractere a ser utilizado	17
2.1.2 Recursos de <i>hardware</i> e <i>software</i>	19
2.1.3 Pré-processamento a ser feito na entrada para o OCR	19
2.2 Dificuldades no Reconhecimento de Caracteres	20
2.2.1 Deformações dos caracteres	21
2.2.2 Desempenho do sistema independente da quantidade e qualidade dos dados manipulados	21
2.2.3 Capacidade de adaptação para novos caracteres	22
2.2.4 Dificuldades na avaliação dos algoritmos	22
2.3 Tecnologias Empregadas em OCR	22
2.3.1 Template-matching.....	23
2.3.2 Análise estrutural.....	24
3 Redes Neurais	30
3.1 Redes Neurais Biológicas	30
3.1.1 Neurônio biológico	31
3.2 Redes Neurais Artificiais	35
3.3 Aprendizado de Redes Neurais Artificiais	37
3.4 Arquitetura Básica de Redes Neurais Artificiais	38
3.5 O Modelo Computacional Perceptron	40
3.6 O Modelo de Kohonen	43
4 Algoritmos Genéticos	47
4.1 Características Básicas	47
4.2 Versão Computacional do Algoritmo Genético	49
4.2.1 Módulos de população e avaliação	49
4.2.2 Módulo de reprodução.....	53
4.3 Aplicações do Paradigma de Algoritmo Genético em Problemas de Otimização	57
5 Implementação do Modelo de Kohonen	58
5.1 Treinamento da Rede	59
5.2 Reconhecimento da Rede	60
5.3 Otimização da Rede Após o Reconhecimento	61
5.4 Execução do Sistema e Avaliação do Desempenho	66

6	Hibridização do Modelo de Kohonen	85
6.1	Implementação do Modelo Hibridizado	85
6.2	Execução do Modelo Hibridizado e Avaliação do Desempenho.....	88
7	Conclusão	97
	Bibliografia	98

Lista de Abreviaturas

ASCII: American Standart Code for Information Interchange. Padrão americano para caracteres codificados em computador.

Ca : Cálcio.

Cl : Cloro.

DNA : Ácido desoxirribonucleico.

DPI : Dot Per Inch, ou pontos por polegada. Unidade de resolução de um periférico de impressão.

ERA : Eletric Reading Automation.

HPR : Hand Printed Recognition ou reconhecimento de caracteres escritos a mão.

K : Potássio.

Na : Sódio.

OCR : Optical Character Recognition ou Reconhecedor Ótico de Caracteres.

RNA : Ácido ribonucleico.

SOM : Self-Organization Feature Map ou Mapa de características auto-organizados.

Lista de Fórmulas

FÓRMULA 3.1- Equação de Nerst para o potencial de cada íon	33
FÓRMULA 3.2- Cálculo da saída do neurônio.....	36
FÓRMULA 3.3- Cálculo do número de cada entrada para disparo do spike.....	37
FÓRMULA 3.4- Função de transferência do tipo hard limiter	37
FÓRMULA 3.5- Função de transferência do tipo sigmoidal	37
FÓRMULA 3.6- Ajuste do fator de transmissão sináptica.....	38
FÓRMULA 3.7- Atualização supervisionada dos pesos	38
FÓRMULA 3.8- Atualização supervisionada dos pesos no perceptron.....	42
FÓRMULA 3.9- Cálculo da distância para estimar o erro no neurônio	44
FÓRMULA 3.10- Ajuste de pesos dos neurônios	45
FÓRMULA 3.11- Cálculo do coeficiente de vizinhança	45
FÓRMULA 5.1- Cálculo da taxa de acerto.....	62
FÓRMULA 5.2- Ajuste de pesos do neurônio que reconheceu	63
FÓRMULA 5.3- Ajuste de pesos do neurônio que não reconheceu	63
FÓRMULA 6.1- Fórmula do crossover	85

Lista de Tabelas

TABELA 5.1- Classificação de reconhecimento da rede a.....	66
TABELA 5.2- Classificação de reconhecimento da rede b.....	66
TABELA 5.3- Reconhecimento da rede - treinamento com 100 padrões de entrada.....	67
TABELA 5.4- Reconhecimento da rede - treinamento com 200 padrões de entrada.....	67
TABELA 5.5- Reconhecimento da rede - treinamento com 300 padrões de entrada.....	68
TABELA 5.6- Reconhecimento da rede - treinamento com 400 padrões de entrada.....	69
TABELA 5.7- Reconhecimento da rede - treinamento com 500 padrões de entrada.....	69
TABELA 5.8- Reconhecimento da rede - treinamento com 600 padrões de entrada.....	70
TABELA 5.9- Reconhecimento da rede - treinamento com 700 padrões de entrada.....	71
TABELA 5.10- Reconhecimento da rede - treinamento com 800 padrões de entrada.....	71
TABELA 5.11- Reconhecimento da rede - treinamento com 900 padrões de entrada.....	72
TABELA 5.12- Reconhecimento da rede - treinamento com 1000 padrões de entrada.....	73
TABELA 5.13- Reconhecimento da rede - treinamento com 1100 padrões de entrada.....	73
TABELA 5.14- Reconhecimento da rede - treinamento com 1200 padrões de entrada.....	74
TABELA 5.15- Reconhecimento da rede - treinamento com 1300 padrões de entrada.....	75
TABELA 5.16- Reconhecimento da rede - treinamento com 1400 padrões de entrada.....	75
TABELA 5.17- Reconhecimento da rede - treinamento com 1500 padrões de entrada.....	76
TABELA 5.18- Reconhecimento da rede - treinamento com 2000 padrões de entrada.....	77
TABELA 6.1- Reconhecimento da rede hibridizada - treinamento com 100 padrões de entrada.....	87
TABELA 6.2- Reconhecimento da rede hibridizada - treinamento com 200 padrões de entrada.....	87
TABELA 6.3- Reconhecimento da rede hibridizada - treinamento com 300 padrões de entrada.....	87
TABELA 6.4- Reconhecimento da rede hibridizada - treinamento com 400 padrões de entrada.....	87
TABELA 6.5- Reconhecimento da rede hibridizada - treinamento com 500 padrões de entrada.....	88
TABELA 6.6- Reconhecimento da rede hibridizada - treinamento com 600 padrões de entrada.....	88
TABELA 6.7- Reconhecimento da rede hibridizada - treinamento com 700 padrões de entrada.....	88
TABELA 6.8- Reconhecimento da rede hibridizada - treinamento com 800 padrões de entrada.....	88
TABELA 6.9- Reconhecimento da rede hibridizada - treinamento com 900 padrões de entrada.....	88
TABELA 6.10- Reconhecimento da rede hibridizada - treinamento com 1000 padrões de entrada.....	88
TABELA 6.11- Reconhecimento da rede hibridizada - treinamento com 1100 padrões de entrada.....	89
TABELA 6.12- Reconhecimento da rede hibridizada - treinamento com 1200 padrões de entrada.....	89
TABELA 6.13- Reconhecimento da rede hibridizada - treinamento com 1300 padrões de entrada.....	89
TABELA 6.14- Reconhecimento da rede hibridizada - treinamento com 1400 padrões de entrada.....	89

TABELA 6.15- Reconhecimento da rede hibridizada - treinamento com 1500 padrões de entrada	89
TABELA 6.16- Reconhecimento da rede hibridizada - treinamento com 2000 padrões de entrada	89
TABELA 6.17- Taxas de reconhecimento para cada treinamento do modelo hibridizado em comparação ao modelo não hibridizado.....	91

Lista de Figuras

FIGURA 2.1- Exemplos de tamanhos dos frames.....	18
FIGURA 2.2- Exemplo de caracteres parecidos	19
FIGURA 2.3- Semelhança entre os números 2, apesar dos diferentes fontes	20
FIGURA 2.4- Deformações comuns em caracteres	21
FIGURA 2.5- Reconhecimento por casamento de padrões.....	23
FIGURA 2.6- Comparação com peepholes.....	24
FIGURA 2.7- Características topológicas e geométricas.....	25
FIGURA 2.8- Características de baixo nível.....	27
FIGURA 2.9- Possíveis falhas nos endpoints e tees.....	27
FIGURA 2.10- Apresentação do corner	28
FIGURA 2.11- Confundindo a identificação de um corner	28
FIGURA 2.12- Características de alto nível.....	28
FIGURA 2.13- Exemplo de modelagem de caractere.....	28
FIGURA 3.1- Esquema de um neurônio e seus componentes principais.....	32
FIGURA 3.2- Variação no potencial da membrana para os íons Na^+ e K^+	33
FIGURA 3.3- Modulador de pulso-frequência.....	35
FIGURA 3.4- Representação gráfica da função hard limiter	37
FIGURA 3.5- Representação gráfica da função sigmoideal	37
FIGURA 3.6- Topologias de redes neurais	39
FIGURA 3.7- Tipos de conexões de redes neurais.....	39
FIGURA 3.8- Tipos de funções de transferência	40
FIGURA 3.9- Reconhecimento com perceptron de um nível	41
FIGURA 3.10- Reconhecimento com perceptron multinível.....	42
FIGURA 3.11- Topologia dos neurônios piramidais do córtex cerebral	43
FIGURA 3.12- Rede neural de Kohonen.....	44
FIGURA 3.13- Espaço de cálculo do erro para rede de Kohonen	45
FIGURA 3.14- Disposição matricial dos neurônios.....	46
FIGURA 4.1- Exemplo de diversidade genética	48
FIGURA 4.2- Exemplo de comportamento emergente.....	48
FIGURA 4.3- Exemplo de cromossomos binários.....	51
FIGURA 4.4- Técnica de seleção da roleta.....	52
FIGURA 4.5- Técnicas de reprodução e número de filhos	53
FIGURA 4.6- Identificação dos esquemáticos e crossover de um ponto	54
FIGURA 4.7- Funcionamento do crossover de dois pontos.....	55
FIGURA 4.8- Comparação do tamanho da população.....	56
FIGURA 4.9- Mutações nos novos indivíduos	57
FIGURA 5.1- Exemplo de algarismos do correio britânico	58
FIGURA 5.2- Divisão das classes em subclasses.....	62
FIGURA 5.3- Mapa dos pesos dos neurônios da rede a - não otimizada.....	64
FIGURA 5.4- Mapa dos pesos dos neurônios da rede b - otimizada	65
FIGURA 5.5- Gráfico do reconhecimento - treinamento com 100 padrões de entrada.....	67
FIGURA 5.6- Gráfico do reconhecimento - treinamento com 200 padrões de entrada.....	68
FIGURA 5.7- Gráfico do reconhecimento - treinamento com 300 padrões de entrada.....	68
FIGURA 5.8- Gráfico do reconhecimento - treinamento com 400 padrões de entrada.....	69
FIGURA 5.9- Gráfico do reconhecimento - treinamento com 500 padrões de entrada.....	70
FIGURA 5.10- Gráfico do reconhecimento - treinamento com 600 padrões de entrada.....	70
FIGURA 5.11- Gráfico do reconhecimento - treinamento com 700 padrões de entrada.....	71

FIGURA 5.12- Gráfico do reconhecimento - treinamento com 800 padrões de entrada.....	72
FIGURA 5.13- Gráfico do reconhecimento - treinamento com 900 padrões de entrada.....	72
FIGURA 5.14- Gráfico do reconhecimento - treinamento com 1000 padrões de entrada...	73
FIGURA 5.15- Gráfico do reconhecimento- treinamento com 1100 padrões de entrada....	74
FIGURA 5.16- Gráfico do reconhecimento - treinamento com 1200 padrões de entrada...	74
FIGURA 5.17- Gráfico do reconhecimento- treinamento com 1300 padrões de entrada....	75
FIGURA 5.18- Gráfico do reconhecimento- treinamento com 1400 padrões de entrada....	76
FIGURA 5.19- Gráfico do reconhecimento- treinamento com 1500 padrões de entrada....	76
FIGURA 5.20- Gráfico do reconhecimento- treinamento com 2000 padrões de entrada....	77
FIGURA 5.21- Vetor de pesos do neurônio 16 - treinamento com 100 padrões de entrada	78
FIGURA 5.22- Vetor de pesos dos neurônios 11, 15, 16, 18 e 19 - treinamento com 200 padrões de entrada.....	78
FIGURA 5.23- Gráfico com as taxas de reconhecimento do padrão 0	80
FIGURA 5.24- Gráfico com as taxas de reconhecimento do padrão 1	80
FIGURA 5.25- Gráfico com as taxas de reconhecimento do padrão 2	80
FIGURA 5.26- Gráfico com as taxas de reconhecimento do padrão 3	81
FIGURA 5.27- Gráfico com as taxas de reconhecimento do padrão 4	81
FIGURA 5.28- Gráfico com as taxas de reconhecimento do padrão 5	81
FIGURA 5.29- Gráfico com as taxas de reconhecimento do padrão 6	82
FIGURA 5.30- Gráfico com as taxas de reconhecimento do padrão 7	82
FIGURA 5.31- Gráfico com as taxas de reconhecimento do padrão 8	82
FIGURA 5.32- Gráfico com as taxas de reconhecimento do padrão 9	83
FIGURA 6.1- Uso do crossover para a redução da rede	86
FIGURA 6.2- Gráfico com as taxas de reconhecimento do padrão 0 - rede hibridizada.....	91
FIGURA 6.3- Gráfico com as taxas de reconhecimento do padrao 1 - rede hibridizada.....	92
FIGURA 6.4- Gráfico com as taxas de reconhecimento do padrao 2 - rede hibridizada.....	92
FIGURA 6.5- Gráfico com as taxas de reconhecimento do padrao 3 - rede hibridizada.....	92
FIGURA 6.6- Gráfico com as taxas de reconhecimento do padrao 4 - rede hibridizada.....	93
FIGURA 6.7- Gráfico com as taxas de reconhecimento do padrao 5 - rede hibridizada.....	93
FIGURA 6.8- Gráfico com as taxas de reconhecimento do padrao 6 - rede hibridizada.....	93
FIGURA 6.9- Gráfico com as taxas de reconhecimento do padrao 7 - rede hibridizada.....	94
FIGURA 6.10- Gráfico com as taxas de reconhecimento do padrao 8 - rede hibridizada...	94
FIGURA 6.11- Gráfico com as taxas de reconhecimento do padrao 9 - rede hibridizada...	94

Resumo

Este trabalho discorre sobre o emprego de sistemas híbridos voltados à resolução de problemas de otimização. Como problemas de otimização entende-se como sendo o emprego de técnicas que visam aumentar a produtividade de alguma tarefa, otimizando seus procedimentos. Desta forma, utiliza-se neste trabalho o paradigma de Algoritmos Genéticos sobre um modelo de Redes Neurais para otimizar seu funcionamento, tornando-o mais rápido e de tamanho menor na tarefa de reconhecimento de padrões.

O modelo de Rede Neural escolhido para o reconhecimento de padrões foi o modelo de Teuvo Kohonen, também conhecido como modelo dos mapas auto organizados (SOM - *Self Organization Feature Map*). Este modelo tem sido empregado, obtendo ótimos resultados, no reconhecimento dos mais diversos padrões, como padrões fonéticos e padrões visuais, destacando sua aplicação em sistemas de reconhecimento ótico de caracteres (OCR - *Optical Character Recognition*), que será explorado em detalhes no decorrer deste trabalho.

O paradigma de Algoritmos Genéticos, criado por John Holland, alcança ótimo desempenho na resolução de problemas de otimização, seja na classificação e seleção do melhor procedimento, seja no desenvolvimento de um novo procedimento baseado na interação do sistema com procedimentos anteriores. Desta forma, os algoritmos genéticos podem ser aplicados em atividades como seleção e classificação, tal como a aplicação para resolver o problema do caixeiro viajante, ou na geração de uma nova estrutura baseada em estruturas anteriores, a citar o redimensionamento de uma rede neural artificial para reduzir o seu tamanho.

Em última instância, este trabalho se propõe a otimizar um sistema de reconhecimento de caracteres utilizando o melhor dos dois paradigmas anteriormente discutidos, obtendo resultados muito satisfatórios na realização dos procedimentos.

PALAVRAS-CHAVES: Redes neurais artificiais, Algoritmos genéticos, sistemas híbridos, reconhecimento de caracteres.

TITLE: Hybrid System applications to solve optimization problems

Abstract

This work is about applying hybrid systems to the solving of optimization problems. We consider optimization problems as the productivity increase of some tasks by fine tuning their procedures using a Genetic Algorithm paradigm on a neural network model, optimizing its functionality, making it faster and decreasing the size of neural network.

We choose Teuvo Kohonen's model for pattern recognition, also known as Self-Organization Feature Map - SOM, which has been used on a wide range of pattern recognition problems, such as phonetic and visual patterns, specially on Optical Character Recognition - OCR systems, which we will discuss later.

The Genetic Algorithm paradigm, created by John Holland, reaches high score performances on solving optimization problems applied on classification and selection process. In this way, genetic algorithms are suitable for selection and classification problems, such as solving the travelling sales person problem or on generating new structures based on prior ones as neural network redimensioning to reduce its size.

The proposal is optimizing the Optical Character Recognition mixing the best properties of both paradigm, aimed very satisfactory results on process execution.

KEYWORDS: Artificial neural network, Genetic Algorithms, hybrid systems, optical character recognition.

1 Introdução

Aplicação de sistemas híbridos em problemas de otimização. Este título merece uma criteriosa redução de seu domínio para viabilizar a compreensão deste trabalho. A começar pelo vocábulo otimização, que significa tornar ótimo, ou simplesmente, fazer o melhor possível. Problemas de otimização, portanto significam resolver um problema da melhor forma possível. O problema abordado neste trabalho é o reconhecimento de caracteres numéricos através de um modelo de rede neural de Teuvo Kohonen. Este trabalho pretende melhorar o desempenho deste reconhecedor diminuindo seu tamanho sem reduzir sua taxa de reconhecimento, aumentando desta forma a velocidade de processamento utilizando para isto menos memória.

Existem várias técnicas de reconhecimento ótico de caracteres, muitas delas construídas para reconhecer um determinado tipo de caractere, outras, extensíveis a uma gama maior de tipos, possibilitando uma flexibilidade maior ao sistema. De um modo geral, estas técnicas possuem uma pequena taxa de reconhecimento quando comparadas com o processo natural de leitura.

Estas técnicas operam através de um conjunto de processos que tomam como referência um modelo de dados, geralmente montado por um especialista, para efetuar o reconhecimento. Quanto mais completo e preciso for este modelo, maior será o conjunto de informações a disposição do reconhecedor para fazer o seu trabalho. A dificuldade maior encontra-se em definir o melhor modelo para o reconhecimento.

O processo natural de leitura é realizado sem o conhecimento prévio por parte do leitor a respeito do modelo de dados empregado no cérebro para efetuar o reconhecimento de caracteres. Neste caso, acredita-se que o modelo seja montado automaticamente pelo próprio cérebro durante a fase de alfabetização do indivíduo, tornando o processo de leitura uma característica adquirida pelo aprendizado.

Espera-se que as modelagens automáticas sejam melhores que as modelagens humanas. É por isso que surgem a toda hora novos modelos de redes neurais artificiais, algumas biologicamente plausíveis e outras, nem tanto. Estes modelos sempre vem com algum diferencial que melhora a qualidade de seu modelo de dados para o reconhecimento de um tipo específico ou tipos genéricos de padrões. O modelo escolhido para a realização deste trabalho foi a rede neural de Teuvo Kohonen por ser um modelo biologicamente plausível baseado no funcionamento dos neurônios do cortex cerebral. Esta característica será constatada com a explicação dos modelos neurais biológicos e do modelo neural de Kohonen presentes neste trabalho.

A técnica de Algoritmos Genéticos, desenvolvida por John Holland em 1975, é empregada neste trabalho para otimizar o funcionamento da rede neural artificial de Kohonen. O motivo que influenciou na escolha deste paradigma para melhorar o modelo neural de Kohonen, deve-se à sua aplicação na resolução de problemas de otimização, utilizando técnicas conhecidas como seleção natural e reprodução. Estes termos, muito conhecidos na biologia, são os responsáveis pela constante evolução da qual fazem parte todos os seres vivos. Acredito que não exista argumento mais favorável à esta escolha. A propósito, os termos evolução, seleção natural e reprodução são discutidos no capítulo sobre algoritmos genéticos.

A partir do capítulo 2 é discutida a finalidade do sistema desenvolvido, que é o de reconhecer caracteres. Neste capítulo, comenta-se sobre as etapas e dificuldades

de se fazer um reconhecimento de caracteres e se apresenta as técnicas de reconhecimento tradicionais mais utilizadas.

No capítulo 3, fala-se sobre o paradigma de redes neurais artificiais, onde se explica a origem desta tecnologia explorando o mais antigo dos modelos, que é o biológico. A partir da compreensão do modelo biológico, passa-se a entender o porque dos pesos dos neurônios, a origem das funções de transferência e de outras equações e topologias empregadas. Entende-se também a origem de muitos modelos neurais bastante conhecidos, como o Perceptron e o próprio modelo de Kohonen. Ainda no capítulo 3, apresentam-se os conceitos básicos dos modelos artificiais, passando pela descrição do Perceptron e encerrando o capítulo com o modelo neural de Kohonen.

Os algoritmos genéticos são apresentados no capítulo 4, onde se verifica as características básicas deste modelo, apresentando uma visão do funcionamento do modelo biológico. A seguir, são explicados os modelos artificiais de Holland e Davis, onde são vistas as etapas de um algoritmo genético, tomando como exemplo uma tradicional aplicação para otimização, que é o controle automático da trajetória de um veículo.

É apresentado no capítulo 5 a implementação do modelo neural de Kohonen num OCR, para reconhecimento dos algarismos contidos na base de dados do correio britânico. São descritas neste capítulo as fases de treinamento, otimização e reconhecimento, apresentando e discutindo os resultados para comparação, a posteriori, com o modelo hibridizado.

No capítulo 6, é descrito o processo de hibridização do modelo de Kohonen apresentando os mecanismos de seleção natural e reprodução responsáveis pela otimização da rede. Os resultados obtidos são discutidos e comparados com os resultados anteriores, apontando as muitas vantagens obtidas e as poquíssimas desvantagens desta técnica.

O capítulo 7 encerra este trabalho analisando as etapas discutidas nos capítulos anteriores e atestando a viabilidade deste sistema híbrido na aplicação de um OCR.

2 Reconhecimento Ótico de Caracteres

A denominação Reconhecimento Ótico de Caracteres (*Optical Character Recognition* - OCR)[OSO 91][FER 94][MAY 94], destina-se a todo o sistema que reconhece caracteres visualizados, seja por periféricos tais como câmaras digitalizadoras que transformam a imagem do respectivo caractere num tipo de informação processável por um sistema de computação, ou pela modelagem deste caractere num meio que possa ser trabalhado como por exemplo a escrita direta na tela do computador ou numa mesa digitalizadora, ou pela recuperação deste caractere de um meio qualquer em que ele tenha sido armazenado anteriormente.

O OCR constitui uma das sub-áreas do reconhecimento de padrões que visa reconhecer toda espécie de padrões, sejam eles visuais, sonoros, de comunicação, etc. Dentro do reconhecimento de caracteres, bem como no reconhecimento de padrões, várias tecnologias estão sendo desenvolvidas, sendo que, para cada conjunto específico de padrões, existe um grupo de técnicas que apresentam melhor desempenho do que outras.

O resultado do reconhecimento feito por um sistema OCR resume-se na transformação do padrão analisado em outro que possa vir a ser processado por um sistema computacional.

Um OCR típico pode ser caracterizado [FER 94] como um conjunto de subsistemas envolvendo um dispositivo responsável pela visualização ou recuperação do caractere e posterior transferência desta entrada para um comparador, que utilizará um banco de dados, contendo informações sobre todos os caracteres a serem reconhecidos pelo sistema, para comparar o caractere enviado pelo dispositivo visualizador/recuperador com o caractere modelado. A entrada pode corresponder a um padrão específico e este, se for identificado pelo sistema, retornará um código correspondente a essa entrada reconhecida. Este código pode ser, por exemplo, um valor hexadecimal no padrão ASCII. Dependendo da técnica de comparação, o sistema pode reconhecer apenas os caracteres modelados em seu banco de dados ou ir mais adiante e partir para generalização, passando a reconhecer não só os caracteres modelados, mas também todos os caracteres semelhantes. A segunda meta é, obviamente, a mais cobiçada pelos pesquisadores.

A área de reconhecimento de padrões tem recebido grande atenção por parte dos pesquisadores, despontando reconhecedores que utilizam técnicas convencionais [KER 88][LJI 63] e reconhecedores que empregam redes neurais artificiais [BRI 92][ROG 92], descritas no capítulo 3.

No início deste capítulo, visando um entendimento mais completo sobre reconhecedores de caracteres, são apresentados as principais etapas no desenvolvimento de um OCR, dificuldades no reconhecimento de caracteres e tecnologias empregadas em OCR.

2.1 Etapas no Desenvolvimento de Um OCR

O desenvolvimento de um sistema de OCR deve utilizar a tecnologia mais adequada ao tipo de caractere a ser reconhecido. Se as entradas são padronizadas, um sistema mais simples, porém menos dispendioso e mais rápido pode vir a preencher as expectativas. Portanto, para um desenvolvimento do sistema de OCR, faz-se necessário

a definição do tipo de caractere a ser utilizado, os recursos de *hardware* e *software* empregados e o pré-processamento a ser feito na entrada para o OCR.

2.1.1 Definição do tipo de caractere a ser utilizado

Se os caracteres forem padronizados, respeitarão uma definição de fonte (forma do caracter) e tamanho do caracter. Para cada modificação em um ou outro parâmetro, faz-se necessário uma mudança no processo de entrada destes caracteres no OCR. O processo de entrada pode ser feito via interfaces gráficas, tais como mesas digitalizadoras, canetas óticas, *mouses*, conversores de fontes *true types* (fontes de tamanhos escalonáveis sem perda da qualidade) em *bit-maps* (matriz de pontos) ou através de digitalizadores óticos, possibilitando a leitura de textos já impressos no papel. Estes dispositivos geralmente retornam, no final do processamento, uma matriz bidimensional contendo elementos setados em 1, representando um segmento do caractere e elementos não ativados (0), fazendo o *background* (região onde o caracter está desenhado) do caracter. Tal matriz bidimensional é conhecida como *frame*, e cada *frame* representa um caractere por vez. Os elementos de um *frame* são conhecidos como *pixels*, uma analogia feita com a menor unidade de uma imagem computadorizada. Os procedimentos importantes no processo de definição de caracteres são a resolução do frame, a resolução dos sistemas de impressão, a resolução dos sistemas de digitalização e a estruturação dos dados.

Os elementos digitalizados correspondem a maior quantidade de entradas para sistemas de OCR. Uma boa digitalização de caracteres resulta em um frame de pelo menos 20 x 15 pixels, como foi reportado por Osório [OSO 91], considerando a média dos fontes utilizados. É importante observar que esta definição tem que tornar compatível o bom desempenho do sistema. Não pode se desconsiderar características de *hardware* e *software* nesta definição, sob pena de se perder itens importantíssimos, como velocidade e memória para o armazenamento de padrões. Este limite pode ser aumentado dependendo das considerações sobre o tipo de dispositivo utilizado na impressão do texto, o tipo de letra utilizada (fonte), e a relação sinal/ruído presente na digitalização.

Dependendo do fonte utilizado, o *frame* pode ser reduzido às dimensões de 8 x 8 ou 10 x 7, considerando que, no caso das digitalizações, praticamente todas as características ficam comprometidas. Estas dimensões são válidas para construção e representação de caracteres alfanuméricos. Uma resolução ainda menor pode ser obtida, como 6 x 4, mas seu domínio restringe-se apenas a representação de caracteres maiúsculos. A figura 2.1 apresenta diferentes conjuntos de *frames*.

Evidentemente, o menor *frame* não serve como parâmetro para muitos caracteres a serem reconhecidos. O que efetivamente deve ser tomado como regra é que a quantidade de características obtidas de um caractere digitalizado é proporcional ao tamanho do *frame*. O ponto de equilíbrio para um *frame* de tamanho razoável está na relação área de armazenamento X tamanho do caractere. Um cálculo considerando o tamanho do caractere ajuda na escolha de uma matriz satisfatória visando um bom aproveitamento de memória. O menor *frame* passível de digitalização só foi apresentado como uma informação sobre até que ponto pode-se reduzir as entradas considerando a resolução que se deseja e as limitações de memória e processamento apresentadas.

como modelagem das classes ou automática, através de sistemas adaptativos tais como redes neurais artificiais [LIP 87][NEV 92][CHE 94].

A estrutura adequada dos dados, possibilita uma organização das rotinas de modelagem dos templates de um OCR, aproveitando o máximo dos atributos das entradas para facilitar o reconhecimento. Quanto melhor a apresentação dos dados, mais fácil o processamento dos mesmos, empregando menos recursos, aumentando assim consideravelmente a performance do sistema.

2.1.2 Recursos de *hardware* e *software*

A avaliação do *hardware* e *software* utilizados deve levar em conta a aplicação do sistema de OCR no mundo real. Caso o reconhecimento deva ser feito em tempo real, os recursos computacionais devem ser maiores, com computadores mais rápidos, maior capacidade de memória, métodos de acesso otimizados ao máximo, algoritmos adequados ao tipo de fonte a ser reconhecido, entre outros itens. No caso de OCRs HPRs (*Hand Printed Recognition*, ou reconhecimento de caracteres escritos a mão) o limite para o reconhecimento em tempo real deve ser menor que 50 milissegundos por caractere escrito [KER 88]. Os recursos também estão vinculados ao grau de precisão desejado num OCR, sendo que, quanto maior a precisão de um reconhecedor, mais apto o sistema encontra-se para reconhecer caracteres parecidos. Por exemplo, os números 5 e 6, 8 e 3, e o número 1 com a letra I são parecidos, ou seja, possuem considerável semelhança na forma como foram desenhados. Quanto mais preciso for um sistema de OCR, mais diferenças ele consegue apurar entre os caracteres, possibilitando um reconhecimento mais eficiente. A figura 2.2 apresenta alguns caracteres parecidos.

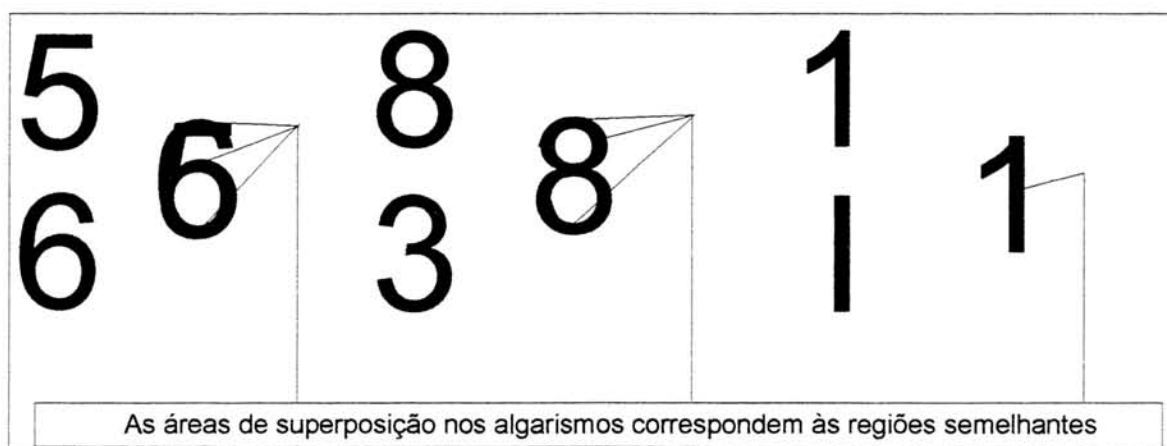


FIGURA 2.2- Exemplo de caracteres parecidos

2.1.3 Pré-processamento a ser feito na entrada para o OCR

Dependendo da forma de obtenção do caractere a ser analisado, faz-se necessário a implementação de um pré-processamento efetuando uma melhoria na qualidade das entradas, realçando características dos caracteres lidos e facilitando o processamento do comparador. O pré-processamento é muito empregado na leitura de textos digitalizados, onde a qualidade do caractere é menor se comparada com a sua geração no computador e também onde surge a necessidade de individualizar os caracteres via segmentação para serem analisados.

No processo de segmentação [LIS 92], a imagem inicial, usualmente apresentada em escalas de cinza, é processada até o isolamento de algumas imagens binárias representando símbolos individuais. O pré-processamento, realizado após a segmentação dos caracteres, consiste basicamente em fazer alguma correção em termos de distorções do caractere [LIS 92]. O principal objetivo é normalmente o de ajustar o tamanho do *frame* resultante da segmentação, de forma que este tenha um tamanho compatível com os templates das classes dos caracteres a serem identificados. Logo, o pré-processamento envolve o ajuste de escala, podendo incluir outros procedimentos como correção de distorções, ajuste de posição, correção de inclinações e rotações e até um segundo nível de tratamento de imagens.

O tipo de caractere a ser reconhecido influencia muito no pré-processamento [MIR 94]. No caso de números arábicos, existem poucas variações numa mesma classe, ou seja, fora o tamanho, todos os números 2, por exemplo, a serem apresentados serão muito parecidos. Além disso, numa análise de estrutura, as características praticamente não variam de fonte para fonte, como pode se observar na figura 2.3.

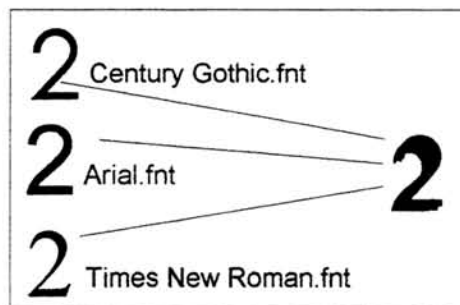


FIGURA 2.3- Semelhança entre os números 2, apesar dos diferentes fontes

Um problema encontrado no reconhecimento de todo o caractere reside na posição em que este é incorporado à matriz de entrada. Existem métodos de reposicionamento dos caracteres na matriz levando em consideração o centro de gravidade (meio) deste caractere. Uma das formas mais simples de se fazer esse tipo de ajuste é identificar o primeiro elemento do caractere que esteja mais acima e o primeiro elemento que esteja mais a esquerda. Após a identificação, desloca-se todos os elementos setados em 1 da matriz de entrada para esquerda e para cima até que os primeiros elementos ocupem as posições das linhas mais acima e à esquerda, respectivamente.

Outros fatores que levam ao pré-processamento estão na qualidade de impressão (intensidade da impressão, falhas, borrões, ruídos de modo geral) que baixam a padronização dos caracteres a serem reconhecidos. Se a entrada dos caracteres foi feita por outra interface, como um *mouse*, por exemplo, necessita-se de um pré-processamento para transformar o conteúdo da matriz representando a página de memória gráfica da interface, cujo elementos foram setados pela trajetória do *mouse* na matriz de entrada para o OCR.

2.2 Dificuldades no Reconhecimento de Caracteres

Existem diversos elementos que dificultam a tarefa de reconhecimento de caracteres, influenciando diretamente no desempenho [OSO 91]. Os problemas de maior relevância são as deformações dos caracteres, o desempenho do sistema

independente da quantidade e qualidade dos dados manipulados, a capacidade de adaptação aos novos caracteres e dificuldades na avaliação dos algoritmos.

2.2.1 Deformações dos caracteres

As deformações mais comuns em caracteres são citadas abaixo:

- Ruídos, muitas vezes causados por falhas no processo de digitalização, transmissão do frame ou inerentes à própria entrada.
- Distorções, relacionadas muitas vezes aos efeitos aplicados nos caracteres (sobreposição, negrito, itálico, etc.), mau posicionamento dos caracteres frente ao digitalizador, etc.
- Translação do caractere, causada pelo posicionamento incorreto do mesmo no *frame*.
- Rotação do caractere, onde o mesmo fica inclinado em relação ao plano do *frame*.
- Mudança de escala do caractere, encontrando em uma mesma série de entradas os mesmos caracteres com tamanhos diferentes.
- Mudança de fonte, onde os caracteres apresentados na entrada podem possuir fontes variadas, tais como *courier*, *times new roman*, etc.
- Segmentação de caracteres, caracterizadas pela separação de textos em decorrência das formatações gramaticais, pela mixagem de textos com gráficos ou pelo *lay-out* definido na construção do texto.
- União de caracteres, causada pela proximidade entre dois caracteres distintos ou pela característica do fonte.

A figura 2.4 exemplifica algumas das deformações mais comuns em caracteres.

Ruídos	Digitalização do caracter com a mancha do background
<i>Distorções</i>	Causada pela função Itálico
<i>Translação</i>	Falha na Impressão do caracter
<i>Escala</i>	Mudança no tamanho dos caracteres
<i>União</i>	Característica do próprio fonte
Fonte	Mudança do fonte dos caracteres

FIGURA 2.4- Deformações comuns em caracteres

2.2.2 Desempenho do sistema independente da quantidade e qualidade dos dados manipulados.

Para melhorar o reconhecimento há um grande acréscimo no número de informações manipuladas. No caso de sistemas *on-line*, existe uma restrição adicional que é referente ao tempo de resposta de processamento, que não deve causar atraso ao sistema de entradas, princípio de integridade de sistemas *on-line*. Uma forma de acelerar o processo de reconhecimento está na alternativa do paralelismo [LIS 92], que pode ser obtido por sistemas adaptativos como característica básica deste paradigma. A

velocidade do *hardware* contribui sensivelmente para elevação da velocidade no reconhecimento [KER 88], bem como uma quantidade maior de memória contribui para um maior número de padrões de entrada, os quais podem ser mais ricos em características, o que apura a exatidão do sistema.

2.2.3 Capacidade de adaptação para novos caracteres

Considerando a grande quantidade de escalas e fontes disponíveis para cada caractere, é importante o acréscimo de uma propriedade para os atuais sistemas de OCR denominada generalização. Com ela, o sistema pode, através de uma base de conhecimento definida, descobrir como cada caractere se parece e generalizar seu formato, reconhecendo como tal todo o frame que possua desenhos parecidos como caracteres. Para isso, pesquisadores estudam sistemas adaptativos que aprendem novos caracteres, incorporando novas características a uma base de conhecimento para criar o conceito de um novo caractere ou reforçar as características de um caractere reconhecido.

2.2.4 Dificuldades na avaliação dos algoritmos

Considerando que para cada fonte a ser reconhecido existe sempre um algoritmo mais adequado, torna-se difícil uma criteriosa avaliação dos sistemas desenvolvidos. Por isso, fez-se um levantamento dos fontes mais comuns encontrados e criou-se bases de dados com estes fontes digitalizados para padronizar o método de avaliação dos OCRs. Entre as bases de dados mais empregadas está a do Correio Britânico [GUA 92][RAM 95a]. O *benchmark* do Correio Britânico é um conjunto de arquivos contendo diversos caracteres digitalizados de fontes impressas em correspondências manipuladas pelo correio britânico. Um destes arquivos foi utilizado pelos OCRs desenvolvidos neste trabalho. Este arquivo contém 3000 fontes com números de 0 a 9 (300 fontes de cada número) repetidos em seqüência. As matrizes binárias bidimensionais dos caracteres são de dimensão 16x24. Os dados do arquivo de Munson possuem 12760 amostras de 46 tipos diferentes de caracteres com uma resolução de 24 x 24 pixels, e os dados de Highleyman possuem 1800 amostras de letras de 500 numerais, com um total de 36 tipos diferentes de caracteres, com uma resolução de 12 x 12 pixels.

Os principais parâmetros a serem tomados como referência são os índices de avaliação de desempenho dos sistemas de OCR. Estes índices consistem de três indicadores:

- Taxa de caracteres reconhecidos corretamente.
- Taxa de caracteres substituídos (classificações incorretas).
- Taxa de rejeição (caracteres não identificados).

2.3 Tecnologias Empregadas em OCR

A análise das diversas técnicas de reconhecimento está intrinsecamente vinculada ao tempo em que elas foram desenvolvidas. As primeiras técnicas, por exemplo, não mencionam nem mesmo a aplicação de sistemas computadorizados no processo de reconhecimento.

As tecnologias explicadas a seguir apresentam duas famílias de reconhecedores que por muitas vezes se fundem em sistemas que as utilizam simultaneamente. São elas *Template-Matching* [HAN 62] e Análise Estrutural [FER 94], que foram selecionados porque durante o levantamento feito sobre alguns sistemas foram encontrados somente OCRs utilizando uma, outra ou ambas as técnicas.

2.3.1 Template-matching

Esta técnica foi desenvolvida por Handel, em 1933 [FER 94], nos Estados Unidos e baseia-se na teoria dos conjuntos. O sistema reconhece os padrões de entrada, interseccionando-os com os padrões modelados no sistema, denominados templates [HAN 62]. Caso a área interseccionada fosse igual a área do template, o caractere de entrada seria reconhecido.

O princípio de funcionamento resumia-se em uma matriz bidimensional de pontos, representando o *template*, e o próprio caractere a ser reconhecido que seria a matriz bidimensional de entrada. O processamento dava-se na sobreposição da matriz do *template* com a matriz do caractere, acusando o reconhecimento caso houvesse uma exata sobreposição. Uma desvantagem apontada nesta técnica é a sua limitação em reconhecer caracteres um pouco diferentes daqueles modelados no sistema. Para isso, criou-se um *bias* para ajustar a intersecção. A medida em que se aumenta o valor do *bias* (ajuste), diminui-se a exigência da igualdade da intersecção do caractere de entrada com o *template*, criando uma flexibilidade que possibilita pequenas diferenças entre a entrada e o padrão armazenado. A seguir são apresentadas as principais técnicas que seguem o paradigma do *template matching*.

O casamento de padrões é uma técnica que toma os atributos de estado (preto ou branco) de todos os pontos pertencentes ao caractere apresentado [OSO 91]. Ela mede simplesmente a similaridade entre o caractere de entrada e as referências armazenadas, pelo casamento e obtenção da correlação de pontos ou grupos de pontos do *frame*, tal como mostra a figura 2.5.

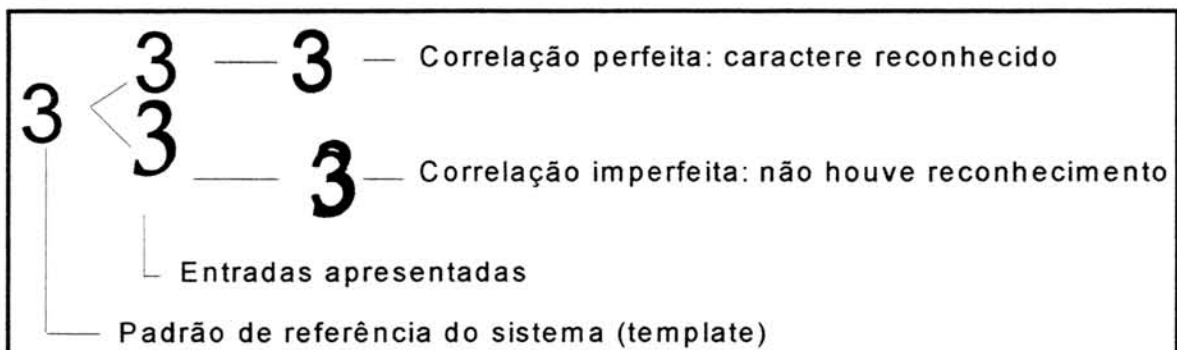


FIGURA 2.5- Reconhecimento por casamento de padrões

Um problema que advém do uso dos próprios pontos do *frame* como atributos resulta na alta dimensão do vetor de atributos usado no casamento de padrões. Para reduzir esta dimensão e extrair atributos não variantes a deformações globais (como por exemplo, rotação e translação), fazem-se necessárias transformações ou expansões em séries. Transformações como de Fourier, Walsh, Haar e Hadamarr, bem como expansões em séries de Karhinen-Loeve tem sido bastante exploradas.

Os pesquisadores Kelner e Glauberman [GLA 56], conseguiram em 1956, a transformação de um padrão de entrada, correspondente à tradicional matriz bidimensional do *template-matching* em um vetor. O processo compreendia a leitura do caractere, feita por um fotodetector, de cima para baixo e um contador digital acoplado a esse fotodetector para contar o grau de luminosidade captado na leitura. A comparação deste valor com um valor correspondente ao caractere armazenado resultaria no reconhecimento.

Desenvolvido em 1957 [ERA 57], o OCR chamado *Electric Reading Automation* (ERA), possuía um digitalizador que colocava o caractere de entrada numa matriz bidimensional (*frame*) e comparava alguns elementos desta matriz, denominados *peepholes* [LJI 63], com elementos equivalentes aos padrões armazenados no sistema. Caso os elementos sejam equivalentes, o reconhecimento será positivo. O sistema poderia reconhecer 120 caracteres padronizados por segundo. Estes caracteres eram números impressos por uma caixa registradora.

Uma desvantagem encontrada no reconhecimento através dos *peepholes* encontra-se na limitação das variações dos caracteres de entrada. Os caracteres apresentados de uma mesma classe não podem ser diferentes e, muito menos encontrarem-se em posições diferentes dentro da matriz bidimensional. Isto porque os *peepholes* a serem analisados correspondem à zonas ativadas (1) e desativadas (0) desta matriz. Se o caractere for muito diferente ou estiver muito deslocado, um ponto setado em 1 para determinado caractere poderá estar em 0. Por isso, é imperativo a escolha criteriosa dos *peepholes* correspondentes a determinado caractere. A figura 2.6 mostra um exemplo de reconhecimento com os *peepholes*.

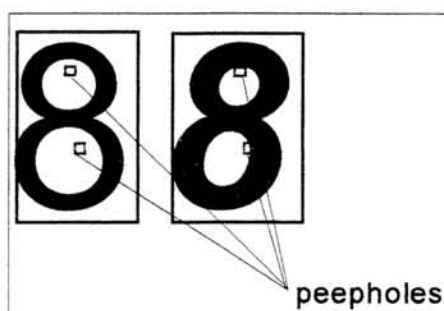


FIGURA 2.6- Comparação com peepholes

Na técnica de zoneamento, a matriz de entrada do caractere, ou *frame*, que contém o caractere é subdividida em várias zonas, sobrepostas ou não, e a densidade de pontos nestas diferentes regiões formarão os atributos [OSO 91]. Este método é usado comumente em reconhecimento de caracteres manuscritos em letra de imprensa, onde as zonas do *frame* são comparadas com as zonas dos *templates* do OCR para processar o reconhecimento.

2.3.2 Análise estrutural

O princípio da análise estrutural é de que uma estrutura, no caso o caractere a ser analisado, pode ser dividido em partes e estas partes podem possuir características e/ou interações com as outras partes do referido caractere [FER 94]. A caracterização das partes leva às características pertencentes ao caractere proporcionando o seu reconhecimento. A análise estrutural surgiu a partir das limitações do *template-matching* na identificação de caracteres muito diferentes, porém pertencentes à mesma

classe. Descobriu-se que a variedade de caracteres poderia exigir uma variedade maior de *templates*, tornando esse número inviável em muitos casos para implementação do sistema. O problema destaca-se na tentativa de reconhecimento de caracteres escritos a mão.

Na análise estrutural, prima-se pela extração das características dentro de dois parâmetros básicos que são a captura total das essências do conjunto de caracteres e o requerimento de um tempo mínimo de processamento para extração.

A primeira técnica demonstrada trata de uma migração do *template-matching* para a análise estrutural, considerando uma tentativa de hibridização onde regras de comparações (*template-matching*) são usadas sobre as características dos segmentos de um caractere. A segunda técnica, mais recente, versa sobre um método de codificação através de regras de inferência sobre a forma de como um caractere deva ser construído, partindo para a generalização do mesmo.

A técnica de análise de abertura consiste em tratar cada *peephole* não como um elemento ou conjunto fixo de elementos de uma matriz bidimensional correspondente a um *template*, mas com uma abertura desta matriz representando uma característica deste *template*. Um exemplo típico seria a chamada técnica da contagem cruzada, onde as aberturas são as linhas da matriz e a quantidade de elementos setados (valor em 1) na abertura corresponde à sua característica. O sistema reconhece o caractere que possui as características de suas aberturas semelhantes às características das aberturas do *template* correspondente.

Além de características quanto ao número de elementos ativados numa abertura (método topológico) existem também características geométricas (método geométrico) [DOY 60]. Estas características relacionam a distância de um elemento setado na matriz com uma linha ou outro referencial colocado na matriz pelo sistema. A figura 2.7 apresenta um exemplo dos métodos topológicos e geométricos na análise de abertura.

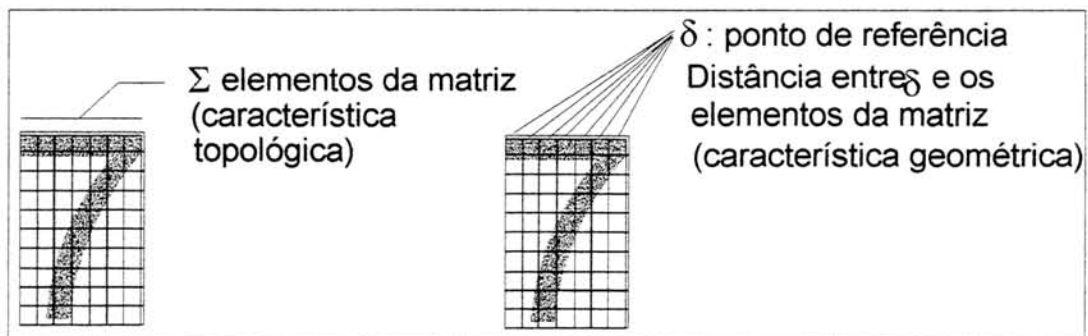


FIGURA 2.7- Características topológicas e geométricas

A extração de características corresponde a uma definição das características dos caracteres onde o número de linhas retas, curvas, vértices ou qualquer outro aspecto geométrico vem a diferenciar um caractere do outro. Kerrick [KER 88] desenvolveu um sistema descritivo das estruturas dos caracteres, baseado na definição do modelo ou número do *template*, seguido da identificação das linhas que compõem o caractere, e a identificação da forma e orientação destas linhas. A próxima etapa compreende a classificação das características geométricas do caractere mediante a comparação destas com as características geométricas dos *templates*. Esta

comparação pode ser feita empregando vários métodos como por exemplo os métodos estatísticos e a comparação simples ou com *bias* de tolerância.

Fazem parte da estrutura do caractere propriedades como seu nome e identificação (vogal, consoante, tônica, átona, etc.). Esta característica extraída encontra utilidade numa análise de contexto, onde regras de construção de palavras (conjunto de caracteres) podem ser empregadas, visando selecionar apenas os *templates* relevantes ao reconhecimento de determinado caractere. Por exemplo, dentro de uma palavra, após uma consoante, existe uma grande probabilidade de apresentar-se uma vogal. Desta forma, os *templates* mais utilizados nesta etapa seriam os que representam as vogais armazenadas no OCR.

No levantamento dos momentos de um caractere, os momentos dos pontos da matriz de entrada em relação a um centro escolhido, como por exemplo o centro de gravidade do caractere, ou de um sistema de coordenadas definido, são usados como atributos de reconhecimento. Estes atributos são comparados com os atributos de referência do sistema para o processo de reconhecimento.

As características estruturais para caracteres escritos à mão demonstram um processo para a leitura de caracteres em tempo real para aplicação em sistemas HPR - *Hand Printed Recognition* [KER 88][MAY 94][GRO 93]. Evidentemente muitas destas regras podem ser empregadas na análise de estruturas de outros tipos de caracteres. A análise estrutural das características divide estas em dois grupos que são as características locais e globais.

As características locais são modificações no traçado do caractere que lhe dá forma [KER 88]. As descontinuidades no traçado, também chamados *endpoints*, são pontos onde o traçado termina.

Os pontos de alta curvatura são traços que fazem curvas para desenho de determinados caracteres tais como as letras *p* e *b*. Os cantos, também conhecidos como *corners* identificam a união de dois traços específicos. As características mais simples são facilmente extraídas e servem de entrada para os módulos de pré-processamento que filtram muitos caracteres a serem comparados considerando a relevância destas características simples. Por exemplo, determinadas características não apresentam-se em nenhum caractere definido nos templates do sistema. Estas entradas já estão descartadas da análise.

As características locais de baixo nível são os *endpoints* e as ligações entre dois ou mais traços, também conhecidas como *tees* [KER 88]. Os *endpoints* são considerados como o traçado entre os *pen-downs* (momento em que o traço começa) e *pen-ups* (momento em que o traço termina). Os caracteres feitos a mão geralmente possuem linhas imperfeitas e pode ocorrer inclusive um *pen-up* indesejado, antes que o traçado termine. Isto pode complicar porque em vez de um traçado tem-se dois. Técnicas de extração de características devem considerar estas possíveis falhas. O tratamento mais comum é definir uma distância limite entre um traço e outro para considerar um *endpoint*. As *tees* indicam a união de dois ou mais traços dando forma ao caractere. Esta forma é analisada pela inclinação dos traços no ponto de contato. A figura 2.8 apresenta as características de baixo nível e a figura 2.9 apresenta as possíveis falhas de um caractere que podem ser confundidas com endpoints ou *tees*.

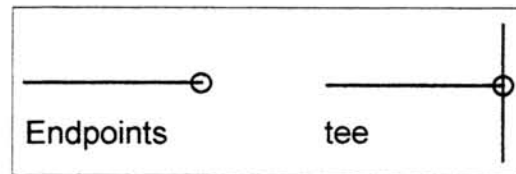


FIGURA 2.8- Características de baixo nível

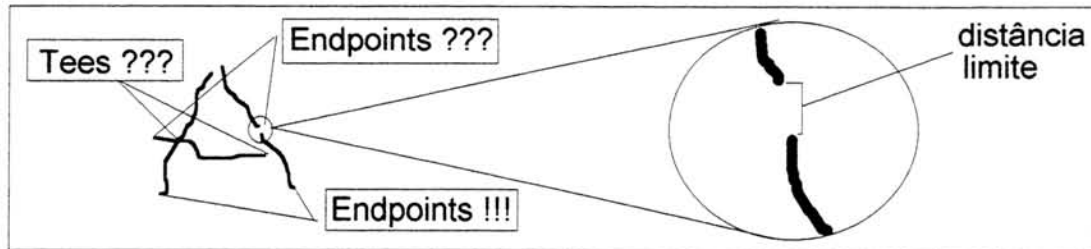


FIGURA 2.9- Possíveis falhas nos endpoints e tees

Denomina-se característica global a localização relativa das características locais dentro do caractere [KER 88], situando onde estão os *endpoints*, os *corners*, as curvas, etc. Também se considera a quantidade de cada uma destas características possuídas pelo caractere bem como a seqüência das mesmas (primeiro uma curva, seguida por um traço reto, etc.).

Existem vários níveis de características globais nos caracteres, sugerindo diversos níveis de análise por parte do reconhecedor. Cada característica de um nível pode ser a raiz de um conjunto de características de níveis inferiores. A organização segue o padrão de árvores binárias de pesquisa, onde elimina-se um grande conjunto de análises percorrendo apenas o caminho indicado pela hierarquia das características.

Outro fator que contribui para a análise das características globais é a divisão do caractere em quadrantes, onde cada quadrante corresponde a uma matriz bidimensional analisada separadamente. De acordo com o resultado da análise de um único quadrante, pode-se excluir alguns templates da comparação, ou determinar antecipadamente qual é o caractere reconhecido, ou mesmo excluir da análise este caractere de entrada. Existem dois níveis de características globais encontrados que são as características de nível intermediário e as características de alto nível.

As características de nível intermediário são identificadas pelos *corners* [KER 88] que são junções de dois traços, fazendo um ângulo que define determinada forma de um caractere. Os *corners* servem de origem para as informações globais. O *corner* retorna como característica de nível intermediário quando se considera como um único traço que em determinado ponto faz uma grande curvatura. Dependendo do grau da curvatura, pode-se flexibilizar a análise de características e aceitar, além dos caracteres angulosos, caracteres mais arredondados. O processo de extração dos *corners* torna-se muito mais complexo que a extração das características de baixo nível. Os *corners* são apresentados na figura 2.10. A figura 2.11 apresenta possíveis imperfeições num caractere que pode confundir a localização dos *corners*.

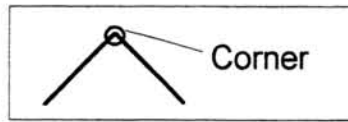


FIGURA 2.10- Apresentação do corner

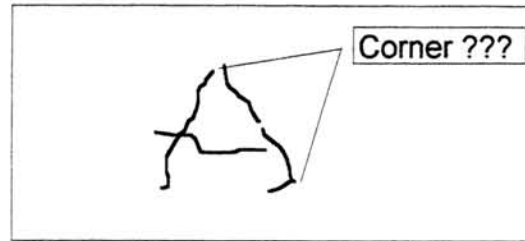


FIGURA 2.11- Confundindo a identificação de um corner

As características de alto nível correspondem à orientação da linha, grau de curvatura do traço e proporção do tamanho dos traços no caractere [KER 88]. A orientação da linha determina a inclinação do caractere na matriz de entrada, o grau de um *corner* ou dos traços numa *tees*. O grau de curvatura de um traço determina os arredondamentos de um caractere, bem como a posição desta curvatura dentro de um quadrante. A proporção do tamanho dos traços num caractere analisa o tamanho de um traço em relação a outro dentro de uma tabela de dimensões. Devido a este tipo de análise, o sistema reconhece a diferença entre a letra *a* minúsculo e a letra *q*, bem como a letra *n* da letra *h*. A figura 2.12 apresenta as características de alto nível.

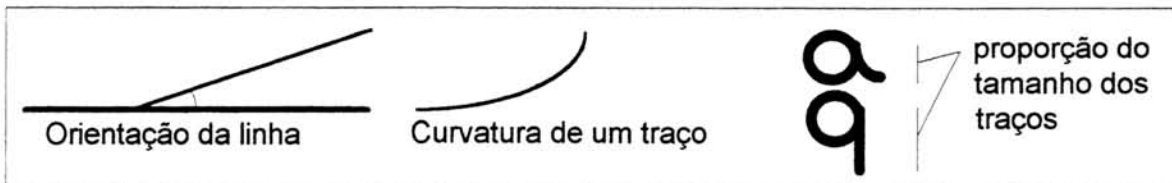


FIGURA 2.12- Características de alto nível

O processo de modelagem das características de alto nível vem encerrar as modelagens de todos os níveis anteriores e é um método empregado em muitos sistemas OCR-HPR existentes. A modelagem gera um template a partir de regras básicas de orientação dos traços e curvas descritos pelo usuário. As regras limitam-se a definir o traço, especificar a qual modelo (caractere) pertence, informar se o traço é reto ou curvo, fornecendo o grau de curvatura e informar também sobre a inclinação do traço em relação a matriz de caracteres. Um exemplo de modelagem é demonstrado na figura 2.13.

	modelo[8].numero_linhas = 3		
	linha 2	modelo[8].linha[1].forma = reta	modelo[8].linha[1].orientação = 90°
	linha 3	modelo[8].linha[2].forma = reta	modelo[8].linha[2].orientação = 90°
linha 1	modelo[8].linha[3].forma = reta	modelo[8].linha[3].orientação = 0°	

FIGURA 2.13- Exemplo de modelagem de caractere

Finalizando, neste capítulo, apresentou-se uma breve explanação sobre reconhecedores de caracteres, evidenciando duas formas de abordagens, *template-*

matching e análise estrutural, que, apesar de distintas, se complementam na construção dos OCRs. De uma forma geral, os reconhecedores não fogem das características referidas nos capítulos anteriores. O que efetivamente muda é o mecanismo comparador, que pode empregar as mais variadas técnicas. As técnicas abordadas nos capítulos subseqüentes baseiam-se em sistemas híbridos envolvendo redes neurais e algoritmos genéticos. Pretende-se provar a viabilidade dos sistemas adaptativos em reconhecer caracteres com uma vantagem singular em relação às demais técnicas empregadas atualmente - dispensar a intervenção humana na modelagem dos templates.

3 Redes Neurais

Como foi visto no capítulo anterior, problemas do mundo real de grande complexidade, como o reconhecimento ótico de caracteres, tem recebido, por parte dos pesquisadores, novas abordagens, através da utilização de sistemas que venham a se adaptar automaticamente às características do problema. Entre os sistemas adaptativos, sobressaem as redes neurais artificiais, descritas neste capítulo, e algoritmos genéticos descritos no capítulo 4.

As redes neurais artificiais representam uma tentativa do homem em mimetizar o comportamento das redes neurais naturais, responsáveis por determinadas atividades nos seres vivos que seriam muito úteis se fossem realizadas por sistemas computadorizados.

As áreas de aplicação onde se empregam redes neurais artificiais eram de domínio exclusivamente humano e sujeitas a erro, causadas pela repetitividade dos procedimentos. A exemplo destas áreas pode-se referenciar aplicação de redes neurais artificiais no reconhecimento de padrões [RYB 93][ROG 92] com aplicação direta no tratamento de ruídos e interferências, comportando-se como filtros adaptativos [SIM 94], o emprego no reconhecimento ótico de caracteres, onde os textos impressos podem ser lidos por computadores, otimizando a entrada dos sistemas de informação [NEV 92][RAM 95a], e no controle automático de trajetórias para veículos autônomos objetivando seu deslocamento e dirigibilidade sem a intervenção humana [KOS 91]. Além disso, faz-se necessário que estas aplicações executem suas funções em tempo real [GUA 93].

O conjunto de atributos oferecidos ao homem por sua rede neural compreende: a capacidade para tomada de decisões; raciocínio por dedução, inferencial e lógico; transmissão de informações através da linguagem; adaptação do organismo através da percepção do mesmo; percepção da realidade através da observação e análise; e, finalmente, o aprendizado, que é o atributo fundamental do qual se derivam os demais. Todos estes atributos permitem ao homem uma percepção dinâmica da realidade, importante recurso para a sua sobrevivência.

Como as redes neurais artificiais baseiam-se no estudo dos modelos neurais biológicos, pensa-se ser importante ter uma breve introdução dos conceitos básicos de redes neurais naturais, mais especificamente de sua unidade elementar, o neurônio biológico, e depois apresentar alguns conceitos introdutórios sobre redes neurais artificiais.

3.1 Redes Neurais Biológicas

Considerando que as redes neurais naturais servem de base para as redes neurais artificiais, é imprescindível o conhecimento do funcionamento básico dos modelos biológicos, o que facilita muito a compreensão das características supracitadas. Uma rede neural biológica pode ser vista como um conjunto de neurônios que desempenham funções elementares de aprendizado e reconhecimento de padrões específicos [ARB 94]. Segundo a teoria geral de sistemas, o conjunto dos comportamentos simples dos neurônios resulta num comportamento complexo da rede neural, caracterizando um comportamento emergente. Rodney Brooks formulou a hipótese da assunção, onde define comportamento emergente como sendo um

comportamento complexo sendo feito por vários comportamentos elementares, confirmando este enfoque sistêmico para as redes neurais. Observa-se, portanto, que o ponto inicial da compreensão de uma rede neural natural parte da compreensão do próprio neurônio.

3.1.1 Neurônio biológico

Na figura 3.1, representa-se o modelo básico dos sistema neurais, o neurônio, com seus organóides identificados. O neurônio recebe estímulos vindo de sensores ou de outros neurônios e estes estímulos representam um padrão de entrada. Os padrões de entrada mais apresentados são aprendidos pelo neurônio e, quando estes padrões aparecem novamente, o neurônio reconhece e emite uma saída. Considerando que cada neurônio pode aprender um padrão de entrada específico, um conjunto de neurônios representando uma rede neural pode classificar todo um conjunto de padrões de entrada, como caracteres, por exemplo. A seguir serão apresentados os mecanismos que possibilitam ao neurônio fazer este processamento.

O neurônio biológico é dividido em corpo celular, junção corpo-axônica e axônio [GUY 84][KOS 91]. No corpo celular é feita a recepção das entradas e o processamento da saída. A junção corpo-axônica canaliza a saída para o axônio que conduz até os botões sinápticos que transmite para fora do neurônio, num processo denominado sinapse. O neurônio encontra-se imerso em uma solução eletrolítica, denominada líquido extra-celular, onde existe concentrações de íons sódio (Na^+), potássio (K^+), cloro (Cl^-) e cálcio (Ca^{++}). Estes íons também existem no interior do neurônio, diluídos no líquido intra-celular. A diferença de concentração destes íons dentro e fora do neurônio determina a diferença de potencial do mesmo, sendo que, quando esta diferença ultrapassa um certo limiar, mudanças ocorrem na membrana do neurônio modificando as concentrações iônicas em seu interior, caracterizando a transmissão de um estímulo nervoso pela variação do potencial da membrana.

Quando o neurônio não recebe nenhum estímulo, ele encontra-se no potencial de repouso. Neste potencial a diferença de potencial da membrana do corpo celular é de -70mV [LYN 93], juntamente com a da junção corpo axônica, e no axônio o potencial fica em torno de -90mV . Os fatores que mantém o potencial de repouso são elucidados a seguir.

O transporte ativo de íons atua como primeiro fator. Seu funcionamento baseia-se no fenômeno eletrogênico, onde bombas iônicas [GUY 84][ROB 85], espalhadas pela superfície da membrana celular, bombeiam, cada uma, 2 íons (K^+) para dentro e jogam 3 íons (Na^+) para fora do neurônio. Como saem mais íons do que entram, caracteriza-se um processo de eletronegatividade.

O segundo fator é a difusão que baseia-se no transporte iônico pela membrana acionado pelo gradiente de concentração e gradiente elétrico dos íons dentro e fora do neurônio [GUY 84]. Deve-se considerar que existe uma concentração de Na^+ de 142mEq/l fora da célula e 14mEq/l dentro da célula fazendo com que este íon exerça um pressão para entrar no neurônio, bem como tem-se uma concentração de K^+ de 140mEq/l no interior da célula e 4mEq/l no meio externo, exercendo uma pressão para sair. Quem controla a vazão de entrada/saída são os canais iônicos distribuídos pela membrana, que são como buracos por onde os íons atravessam.

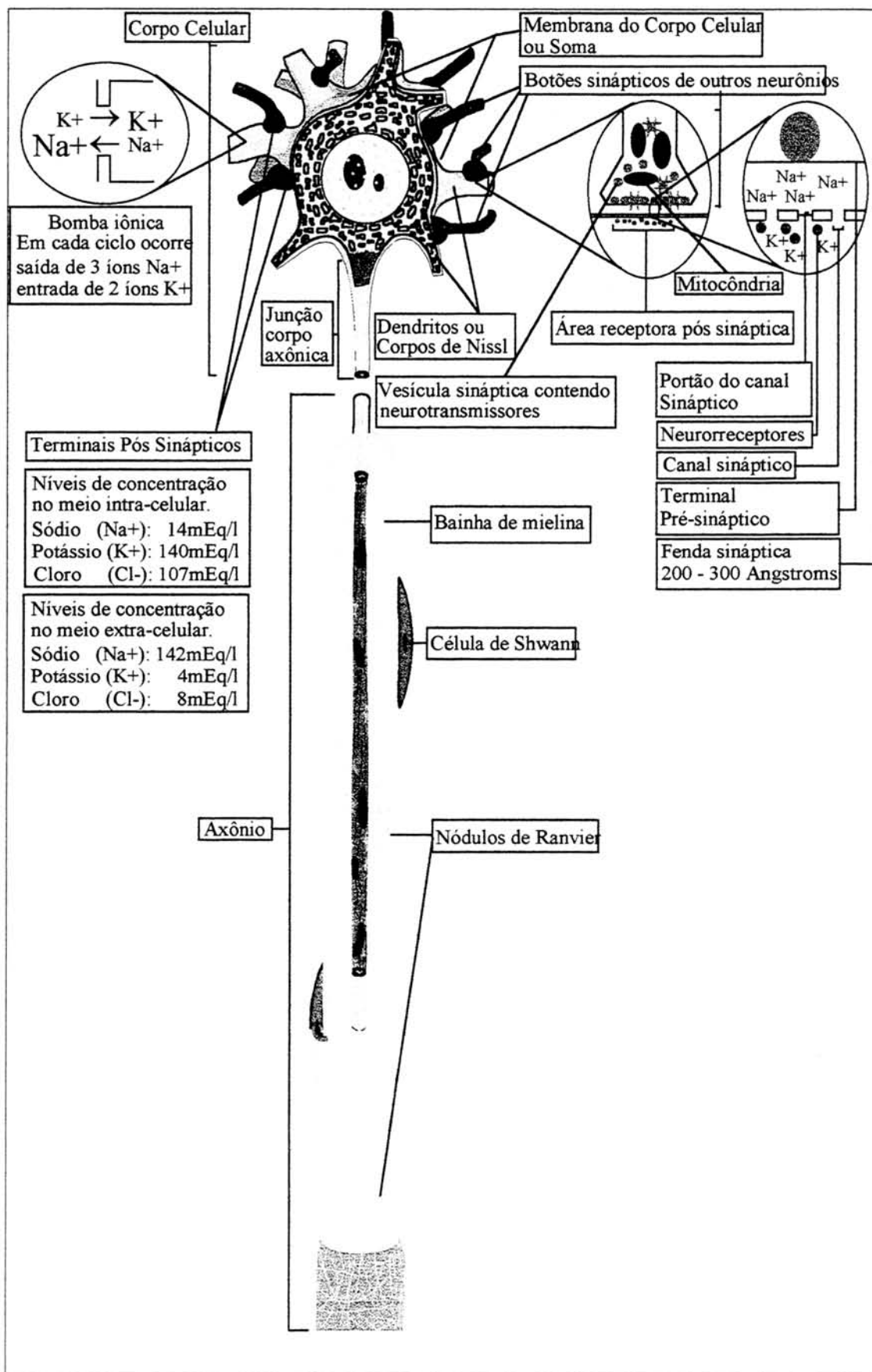


FIGURA 3.1- Esquema de um neurônio e seus componentes principais

Estes canais possuem portões, que são estruturas constituídas de Ca^{++} e servem para bloquear a passagem dos íons por eles. Existe um tipo de canal para cada tipo de íon e, os canais de K^+ , deixam passar de 50 a 100 vezes mais íons do que os canais de Na^+ durante o potencial de repouso. Nerst [GUY 84] desenvolveu uma equação (fórmula 3.1) para calcular o potencial no interior da membrana dado por cada íon, dependendo de sua concentração, onde c_i corresponde à concentração iônica no interior da célula e c_e corresponde à concentração no meio externo.

$$\text{FEM(milivolts)} = -61 \log (c_i/c_e)$$

FÓRMULA 3.1- Equação de Nerst para o potencial de cada íon

Os gráficos da figura 3.2, obtidos com a aplicação da equação de Nerst, apresentam a variação dos potenciais da membrana de acordo com a variação nas concentrações de Na^+ e K^+ . Importante observar que o potencial de repouso da membrana do axônio é o mesmo obtido com a máxima concentração de K^+ em c_i .

O terceiro e último fator está na eletronegatividade residual do neurônio causada por íons não difusíveis, tais como o sulfato e o fosfato que estão no interior do neurônio [GUY 84]. Esta eletronegatividade cria um gradiente elétrico entre o meio intra e extra-celular, fazendo com que os íons K^+ sejam atraídos para dentro do neurônio e os íons Na^+ empurrados para fora via difusão. Este jogo de forças entre concentração e potencial elétrico mantém as taxas de concentração dentro dos níveis supra-citados bem como estabiliza o potencial do neurônio.

Existem estímulos mecânicos, térmicos, elétricos e, principalmente químicos que podem enfraquecer os portões dos canais de Na^+ na membrana do corpo celular [GUY 84][LYN 93]. Quando isto acontece, os íons Na^+ invadem o corpo celular, reduzindo a eletronegatividade de -70mV para -59mV . Quando este potencial se distribui por toda a membrana do neurônio, a junção corpo-axônica encontra-se com -40mV , ou seja com uma eletropositividade de 30mV em relação ao potencial de repouso da membrana do corpo celular. Quando a região da membrana atinge os -59mV o Na^+ reduz sua vazão de entrada por causa da repulsão elétrica causada pela redução da eletronegatividade.

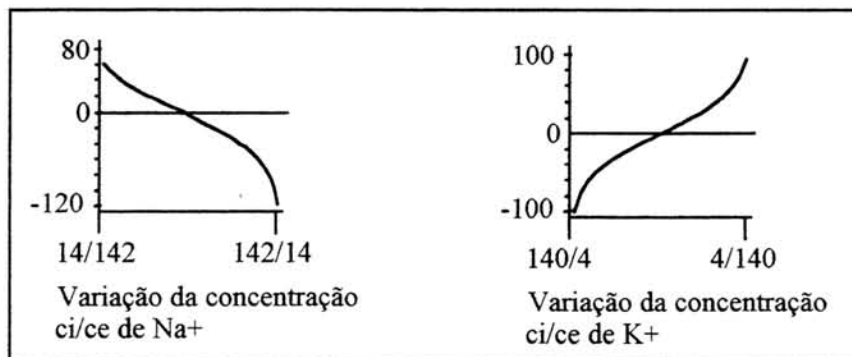


FIGURA 3.2- Variação no potencial da membrana para os íons Na^+ e K^+ .

Com isso, os portões dos canais iônicos se reconstituem tornando a bloquear o Na^+ . Juntamente com o bloqueio do Na^+ , o K^+ aumenta sua vazão para fora da célula, causada pela alta concentração no meio intra-celular aliada a uma queda na atração elétrica deste íon pelo corpo celular causada pela redução da eletronegatividade. A medida em que o K^+ vai saindo, a eletronegatividade vai aumentando até o

reestabelecimento do potencial de repouso, conhecido como potencial de inversão. As concentrações iônicas originais são reestabelecidas pelas bombas iônicas.

Os canais de Na^+ na membrana do axônio, próximos à junção corpo-axônica, abrem-se permitindo a entrada deste íon [GUY 84][LYN 93]. A eletronegatividade baixa de -90mV para 45mV , constituindo um *spike* ou potencial de ação. Quando a região da membrana atinge os 45mV , o Na^+ desacelera, os portões dos canais de Na^+ tornam a fechar-se ao mesmo tempo em que o K^+ sai para fora do axônio (potencial de inversão). O potencial retorna para os -90mV e o equilíbrio das concentrações iônicas é reestabelecido pelas bombas iônicas. Assim como a região da membrana do axônio próxima à junção corpo-axônica entrou em potencial de ação, outras regiões da membrana que estão em contato passam pelo mesmo processo, conduzindo o *spike* por todo axônio até o botão sináptico. Muitas vezes o axônio é mielinizado por um processo denominado abrassamento onde uma célula de Schwann reveste o axônio com uma blindagem de mielina. Estas blindagens são separadas por nódulos onde são feitas as trocas iônicas. Com isso, aumenta-se a capacitância da membrana em até 1000 vezes, juntamente com a velocidade de propagação dos íons, que são feitas ponto a ponto. Como as trocas são feitas em pontos separados pela distância média de 1mm , a energia utilizada é milhares de vezes menor em relação ao sistema de trocas dos axônios não mielinizados.

O botão sináptico recebe o *spike* do axônio [GUY 84][LYN 93]. Este *spike* enfraquece os portões dos canais de Ca^{++} do terminal pré-sináptico, nome dado à região da membrana do botão sináptico correspondente à saída do neurônio. Ao enfraquecerem, os canais deixam o Ca^{++} entrar no botão sináptico e fixar-se nas paredes das vesículas sinápticas, que são bolsas contendo neurotransmissores. As vesículas aumentam de volume, se aglutinam e explodem, liberando os neurotransmissores através da membrana. Estes neurotransmissores atravessam a fissura sináptica, que é um espaço entre o terminal pré-sináptico e a região pós-sináptica de um outro neurônio. Por região pós-sináptica entende-se todas as regiões da membrana do corpo celular mais próximas o possível de terminais pré-sinápticos. Ao chegarem nos terminais pré-sinápticos, os neurotransmissores entram em reação com os neurorreceptores existentes nos terminais pós-sinápticos para enfraquecerem os portões dos canais iônicos. Para cada neurotransmissor existe um neurorreceptor específico e a reação destes atuam sobre canais iônicos também específicos.

No caso de uma sucessiva sequência de *spikes*, existe a possibilidade de esgotarem as reservas de vesículas sinápticas [GUY 84]. Tal esgotamento diminui ou até mesmo interrompe a emissão de neurotransmissores, tornando os potenciais de ação deste neurônio totalmente inócuos. Este fenômeno denomina-se fadiga na transmissão sináptica e ocorre a partir de uma determinada frequência de pulsos variável de acordo com o tipo de neurônio.

Todo o estímulo que enfraquece os portões dos canais de Na^+ é excitatório porque induz o neurônio a um potencial de ação. Existem também os estímulos inibitórios que impedem o neurônio de entrar num potencial de ação [GUY 84]. Certas reações de neurotransmissores e neurorreceptores afetam os canais dos íons Cl^- , permitindo sua entrada pela membrana. Ao fazer isso, o corpo celular do neurônio hiperpolariza, tornando-se ainda mais eletronegativo, afastando o limiar de -59mV para disparar um *spike* no axônio. A hiperpolarização de um neurônio é tão comum quanto

os potenciais de ação e é este jogo de forças que vai decidir se o neurônio vai disparar ou não.

A taxa de Ca^{++} influencia na atuação dos portões dos canais iônicos de Na^+ . Dependendo da taxa de Ca^{++} , o neurônio pode entrar num processo de ritmividade onde, mesmo sem receber nenhum estímulo, os potenciais de ação são gerados. Isto se deve ao fato de que o Na^+ pode difundir-se mais rapidamente para dentro da célula, alterando a capacitância da membrana, levando a despolarização da membrana do corpo celular. Esta ritmividade também é conhecida como atividade residual (background) do neurônio.

3.2 Redes Neurais Artificiais

O primeiro modelo formal de um neurônio foi feito por McCulloch e Pitts [KOH 88][LIP 87][WID 90] numa tentativa de reproduzir o comportamento da membrana do corpo celular. Eles concluíram que o neurônio não passa de um somador de entradas (sinais vindos de outros neurônios) com um limiar para disparar. Caso o somatório seja maior que o limiar, a saída (potencial de ação) ocorre. O modelo artificial desenvolvido por eles foi um modulador de pulso frequência, apresentado na figura 3.3, que consiste em um capacitor que vai carregando lentamente a medida em que vai sendo alimentado por um gerador de corrente. Ao atingir um determinado limiar, o mesmo descarrega completamente, iniciando o ciclo novamente. Foi observado que o gerador de corrente alimentava o modulador numa frequência constante e que a frequência de saída era proporcional à frequência de entrada dentro das condições normais de operação do circuito. Deduziram que este comportamento também pode ser feito pela membrana do corpo celular e, a partir das confirmações em testes feitos com neurônios naturais, formularam a hipótese da linearização (*linearization assumption*) [KOH 88], que trata da proporcionalidade entre a frequência de entrada e a frequência de saída de um neurônio.

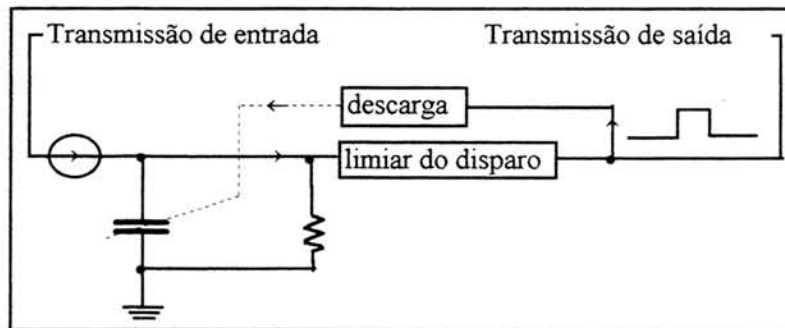


FIGURA 3.3- Modulador de pulso-frequência

Seja o neurônio i que recebe em cada uma de suas três entradas a saída de um neurônio j (um neurônio j para cada uma das entradas do neurônio i). Quando i atinge seu potencial de ação, constata-se que a entrada 1 deu 20 pulsos ($\epsilon_1=20$), a entrada 2 deu 10 pulsos ($\epsilon_2=10$) e a entrada 3 deu 30 pulsos ($\epsilon_3 = 30$) num mesmo intervalo de tempo. Os neurônios j trabalham em frequências distintas porque levam tempos diferentes para dispararem um spike. Por isso, analisa-se o conjunto de entradas e saídas dentro de um intervalo de tempo (segundo). Portanto, caso o neurônio i possua uma saída de 5 Hz (ciclos por segundo), os valores para as entradas ficam em $\epsilon_1 = 100$ Hz, $\epsilon_2 = 50$ Hz e $\epsilon_3 = 150$ Hz.

Considerando:

1. A frequência dos pulsos de entrada ϵ_j , $j = 1, 2, \dots, n$ onde n representa o número de neurônios j (entradas) conectadas ao neurônio i .

2. Um fator μ_{ij} ou fator de transmissão sináptica que representa:

Taxa de neurotransmissores liberados por neurônio j em cada ϵ_j , sendo que fadiga na transmissão sináptica do neurônio j pode reduzir esta taxa com o passar do tempo caso ϵ_j mantenha-se elevado.

Taxa de neurorreceptores em cada entrada j , do neurônio i . No processo de aprendizado do neurônio verifica-se que esta taxa pode aumentar em função da frequência ϵ_j com passar do tempo.

Proximidade da entrada ϵ_j da junção corpo axônica, sendo que quanto mais perto da junção, maior a influência da entrada na produção de um potencial de ação.

Natureza do íon (positivo ou negativo) que atravessa os canais sinápticos (o sinal do íon é o sinal de μ_{ij}).

3. Um fator ηb_i como sendo a atividade residual ou de fundo (*background*) da membrana do neurônio i .

Conclui-se que:

1. A entrada efetiva de cada neurônio j no neurônio i é $\mu_{ij}\epsilon_j$, onde se observa que o fator de transmissão sináptica μ_{ij} atua como um peso na ponderação da entrada ϵ_j .

2. O somatório das entradas efetivas do neurônio i ($\sum \mu_{ij}\epsilon_j$) elevam o potencial da membrana, aproximando o neurônio de um potencial de ação.

3. O fator ηb_i deve ser adicionado às mudanças geradas pelas entradas, uma vez que a atividade residual representa o número de *spikes* dados espontaneamente pelo neurônio, resultando na equação $\eta b_i + \sum \mu_{ij}\epsilon_j$.

4. Segundo a hipótese da linearização, a frequência de saída do neurônio, representada por \square_i , é diretamente proporcional à $\square \square_{ij} \square_j$. A relação entre os dois termos é mostrada na fórmula 3.2.

$$\eta_i = \eta b_i + \sum_{j=1}^n \mu_{ij} \epsilon_j$$

FÓRMULA 3.2- Cálculo da saída do neurônio

A fórmula 3.3 determina o número de impulsos de cada entrada j , representado por ϵ'_j , necessários para disparar um spike no neurônio i . Caso ϵ_j não atinja o valor mínimo de ϵ'_j , o *spike* não é disparado. Portanto, existe um limiar de ativação a ser considerado no cálculo para ativação do neurônio. Este limiar, também conhecido como *hard limiter*, discretiza a saída do neurônio. A fórmula 3.4 apresenta a função limiar e a figura 3.4 apresenta o gráfico da função.

$$\varepsilon_j' = \varepsilon_j / \eta_i$$

FÓRMULA 3.3- Cálculo do número de cada entrada para disparo do spike

$$\text{se } \sum \mu_{ij} \varepsilon_j \geq \sum \mu_{ij} \varepsilon_j' \text{ então } \eta_i = 1 \text{ senão } \eta_i = 0$$

FÓRMULA 3.4- Função de transferência do tipo hard limiter

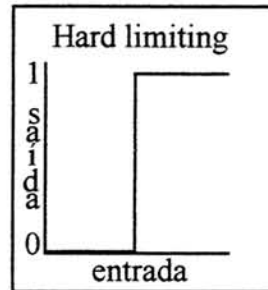


FIGURA 3.4- Representação gráfica da função hard limiter

Alguns fatores modificam a proporcionalidade entre as frequências de entrada e saída, como a fadiga na transmissão sináptica ou uma junção de um ou mais *spikes* causada por uma alta frequência de entrada [GUY 84][KOH 88]. Esta junção ocorre porque o ciclo de um *spike* dura alguns milissegundos e neste meio tempo o neurônio pode ter recebido entradas suficientes para disparar um novo *spike*. Estes fatores colocam o neurônio fora da hipótese da linearização, não podendo mais representar a saída em função da entrada por um *hard limiter*. Neste caso, a função que representa o comportamento do neurônio nestes casos é uma *sigmoidal* como a representada na fórmula 3.5, seguida da figura com o gráfico da função na figura 3.5.

$$\eta_i = 1 / (1 + e^{-(\sum \mu_{ij} \varepsilon_j)}), \text{ para } j=(1...n)$$

FÓRMULA 3.5- Função de transferência do tipo sigmoidal

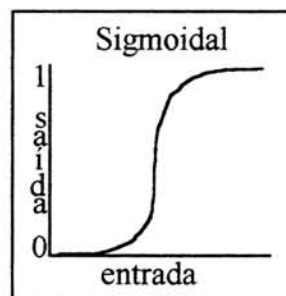


FIGURA 3.5- Representação gráfica da função sigmoidal

3.3 Aprendizado de Redes Neurais Artificiais

Seja um neurônio i que recebe vários padrões de entrada, cada um composto de n entradas elementares ε_j com $j=(1...n)$ que são submetidas a uma ponderação pelo

neurônio. Este neurônio, para retornar uma saída correspondente ao reconhecimento ou não do padrão de entrada, precisa aprender algo sobre esse padrão. O fator de transmissão sináptica (μ_{ij}) atua como um peso que intensifica ou atenua o valor de ϵ_j . Observando a fórmula 3.2 verifica-se que a alteração do valor de μ_{ij} para cada ϵ_j altera também o valor da saída η_i até que esta saída possua o valor correspondente às características do padrão de entrada.

Marr e Eccles, baseados nos trabalhos de D.O.Hebb [KOH 88], elaboraram a teoria da conjunção do aprendizado. Esta teoria afirma que se uma entrada ϵ_j apresenta uma frequência elevada, μ_{ij} tende a mudar seu valor para facilitar esta entrada para o neurônio. Esta facilitação é proporcional à frequência de ϵ_j . Para controlar a velocidade de convergência de μ_{ij} foi definido um coeficiente de plasticidade sináptica (γ), que espelha a velocidade da facilitação do neurônio para ϵ_j . Se γ for muito elevado, μ_{ij} apresentará o valor de ϵ_j em cada interação, trazendo como desvantagem a perda da generalização, pois o neurônio só poderá reconhecer aquele padrão de entrada. Por outro lado, caso γ seja muito baixo, μ_{ij} poderá ficar estático, sem atualizar seu valor fazendo com que o neurônio nunca aprenda. O valor de γ deve possibilitar que μ_{ij} convirja lentamente para que assuma o valor médio dos vários padrões de entrada. A atualização de μ_{ij} fica de acordo com a fórmula 3.6. A fórmula 3.7 apresenta uma alternativa proposta por Hebb de atualizar, de forma supervisionada, o valor de μ_{ij} , utilizando como referência uma saída desejada d_i . Nesta fórmula segue a nova notação dos coeficientes onde $w_{ij} = \mu_{ij}$, $x_j = \epsilon_j$ e $y_i = \eta_i$.

$$\mu_{ij} = \gamma \epsilon_j$$

FÓRMULA 3.6- Ajuste do fator de transmissão sináptica

$$\Delta w_{ij} = \gamma (d_i - y_i) x_j$$

FÓRMULA 3.7- Atualização supervisionada dos pesos

A partir das fórmulas abstraídas do processamento natural, os pesquisadores desenvolveram algumas versões computadorizadas. Uma das redes pioneiras foi o Perceptron, que utiliza a função do aprendizado supervisionado de Hebb, obtendo resultados satisfatórios o suficiente para dar início a uma grande família de redes neurais baseadas em sua arquitetura.

3.4 Arquitetura Básica de Redes Neurais Artificiais

Uma rede neural artificial pode ser considerada como um conjunto de processadores fortemente conectados. Cada processador é responsável por uma tarefa simples, e estas tarefas são executadas simultaneamente, conferindo à rede neural um massivo processamento paralelo. Possuem capacidade de aprender, de maneira supervisionada ou independente, adaptando-se às condições apresentadas.

As topologias de redes neurais são enquadradas em dois tipos bem diferenciados. A primeira topologia é a disposição linear dos neurônios, onde estes se encontram em um mesmo nível dentro da rede, como por exemplo, o modelo de Hopfield [RYB 93][CAV 93][FRE 92]. O segundo tipo é uma topologia multinível,

onde os neurônios estão distribuídos em várias camadas hierárquicas dentro da rede, a exemplo dos Perceptrons multiníveis, [LIP 87][WER 90][MIL 94]. As topologias são apresentadas na figura 3.6.

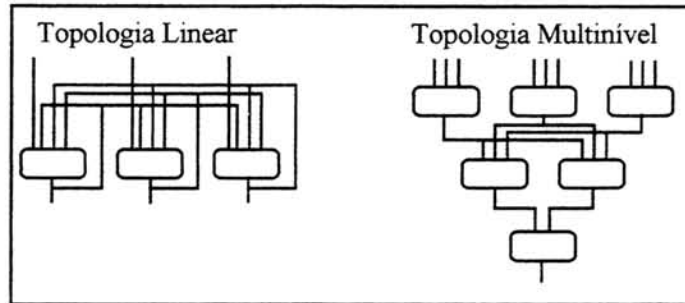


FIGURA 3.6- Topologias de redes neurais

Existem três tipos de conexão empregadas em diversos modelos. São as conexões retro-propagadas, propagadas lateralmente e propagação adiante. As conexões retro-propagadas e propagadas lateralmente definem uma rede em que cada neurônio possui sua saída ligada com uma de suas entradas e/ou com as entradas de outros neurônios. A diferença entre as conexões é que a primeira ocorre em redes multinível, e a segunda ocorre em redes lineares. Já a conexão do tipo propagação adiante representa uma conexão típica de redes multinível, onde a saída dos neurônios do nível mais acima são as entradas dos neurônios dos níveis abaixo. Os tipos de conexões são mostrados na figura 3.7.

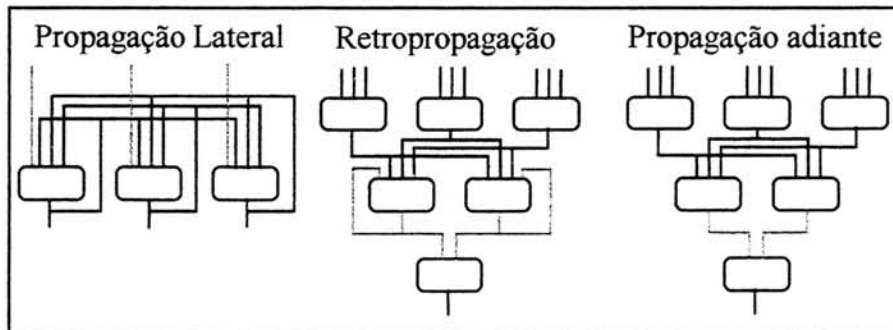


FIGURA 3.7- Tipos de conexões de redes neurais

O modo de aprendizado utilizado pelas redes neurais enquadram-se em duas categorias. A primeira é a categoria do auto-aprendizado, ou *self learning*. Numa rede *self learning*, bastam apenas a apresentação dos padrões de entrada para que a rede possa aprendê-los através de um processo de classificação dos padrões, separando-os pelas suas características, a exemplo do modelo de Kohonen [KOH 90]. A outra categoria é a do aprendizado supervisionado onde, além dos padrões de entrada, faz-se necessário também a apresentação das saídas desejadas para cada padrão, como no Back-propagation [WID 90].

Referindo-se às características funcionais de uma rede neural tem-se que os neurônios podem assumir um comportamento competitivo, onde os mesmos competem por armazenar cada padrão de entrada. Com o passar das interações, os neurônios vão se distribuindo formando vizinhanças que representam os padrões de entrada aprendidos. Outra característica funcional inerente às redes neurais é o aprendizado por correção de erros, onde cada neurônio vai ajustando seus parâmetros para aprender o padrão de entrada apresentado. Cada padrão apresentado pode ser armazenado num

neurônio, característica denominada memória localizada, ou o padrão pode ser armazenado por diversos neurônios caracterizando a memória como distribuída [OSO 91].

A rede neural aprende devido ao ajuste individual dos pesos de cada neurônio. Por pesos entende-se que são variáveis associadas à entrada dos neurônios que podem alterar o padrão de entrada. Esta alteração pode promover um estímulo no neurônio ou a sua inibição, dependendo do valor contido nestes pesos. O processamento consiste na soma das entradas multiplicadas pelos pesos correspondentes em cada neurônio. O resultado passa por uma função de transferência que determina, através de um limiar, o tipo de saída que o neurônio terá com aquelas entradas. A função de transferência pode ser do tipo *hard-limiter*, semi-linear ou *sigmoidal* [KEM 94a][KEM 94b]. O resultado da função é a saída do neurônio. Na figura 3.8 encontram-se as curvas correspondentes a cada função supra-citada.

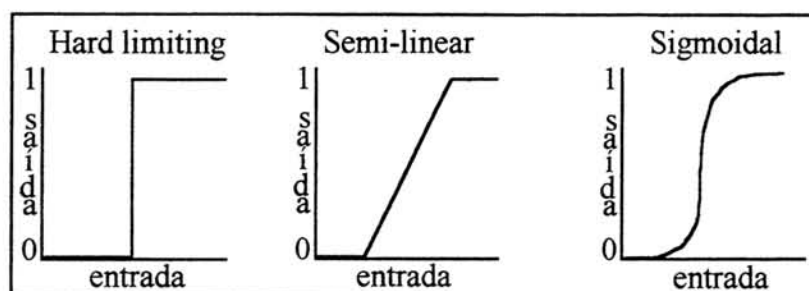


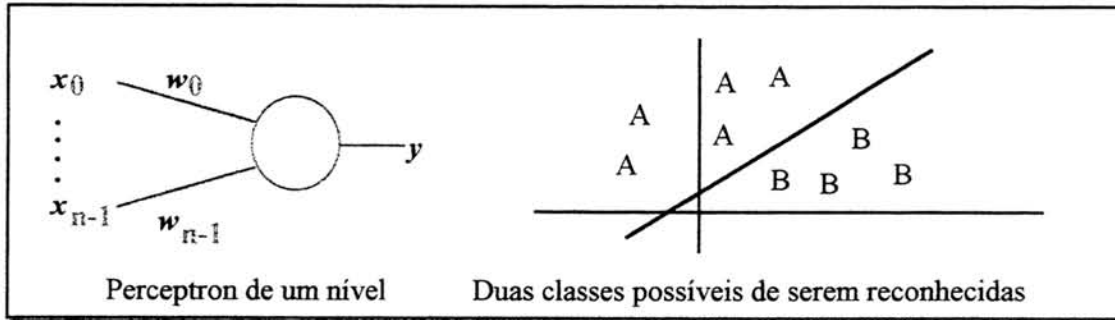
FIGURA 3.8- Tipos de funções de transferência

3.5 O Modelo Computacional Perceptron

Em 1959, Rosenblatt propôs uma rede de neurônios binários denominada Perceptron [KOH 88][WID 90]. Esta era uma rede de aprendizado supervisionado, possuía apenas um nível, sua memória era localizada e a função de transferência era do tipo *hard-limiter*. O funcionamento do perceptron introduz o conceito de plano de decisão e evidencia algumas limitações das redes neurais solucionáveis apenas com mudanças topológicas. Por plano de decisão, entende-se como sendo uma área onde as entradas são linearmente separadas. Cada área separada do plano representa a decisão tomada pelo neurônio de transmitir ou não a saída.

Sejam as entradas x_j , $j=(1...2)$ para o neurônio i , sendo que estas podem assumir os valores 0 ou 1. Os pesos (w_{ij}) foram configurados para que o neurônio reconheça as funções **and**, **or** e **xor**. A função de transferência é uma condição do tipo se $(w_{i1}x_1 + w_{i2}x_2) > 1$ então a saída $y_i = 1$ senão a saída $y_i = 0$. A saída do perceptron é comparada com a saída esperada e para verificar o reconhecimento. A figura 3.9 apresenta o perceptron, as tabelas de reconhecimento feitas para a função **and**, **or** e **xor** respectivamente, juntamente com o desenho do espaço de decisão para cada função reconhecida.

Uma publicação de 1969, intitulada Perceptrons, dos pesquisadores Minsky e Papert, explica a limitação deste modelo em reconhecer o padrão de entrada **xor** devido ao princípio da diferenciação de padrões linearmente separáveis, onde não se pode separar com uma linha reta o grupo das entradas do **xor** para todas as saídas possíveis do neurônio.



x_1	x_2	w_{i1}	$w_{i1}x_1$	w_{i2}	$w_{i2}x_2$	$w_{i1}x_1 + w_{i2}x_2$	$y_i = f(w_{i1}x_1 + w_{i2}x_2)$	e.
0	0	1	0	1	0	0	0	0
0	1	1	0	1	1	1	0	0
1	0	1	1	1	0	1	0	0
1	1	1	1	1	1	2	1	1

x_1	x_2	w_{i1}	$w_{i1}x_1$	w_{i2}	$w_{i2}x_2$	$w_{i1}x_1 + w_{i2}x_2$	$y_i = f(w_{i1}x_1 + w_{i2}x_2)$	e.
0	0	2	0	2	0	0	0	0
0	1	2	0	2	2	2	1	1
1	0	2	2	2	0	2	1	1
1	1	2	2	2	2	4	1	1

x_1	x_2	w_{i1}	$w_{i1}x_1$	w_{i2}	$w_{i2}x_2$	$w_{i1}x_1 + w_{i2}x_2$	$y_i = f(w_{i1}x_1 + w_{i2}x_2)$	e.
0	0	2	0	2	0	0	0	0
0	1	2	0	2	2	2	1	1
1	0	2	2	2	0	2	1	1
1	1	2	2	2	2	4	1 ???	0

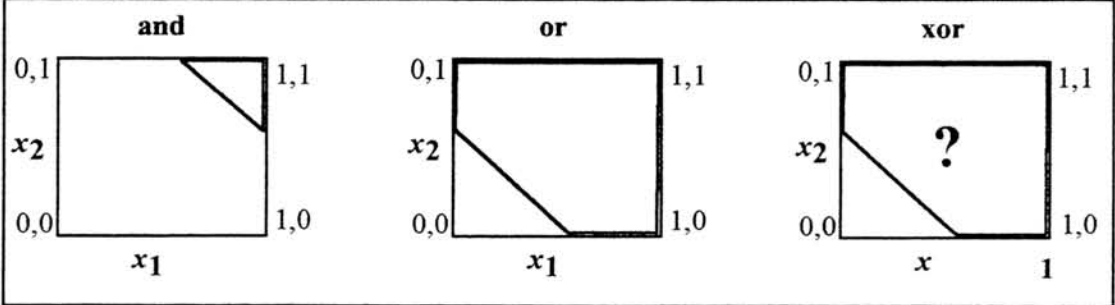


FIGURA 3.9- Reconhecimento com perceptron de um nível

A solução que possibilitou aos perceptrons resolverem este problema foi a criação de uma topologia multinível. A figura 3.10 apresenta a nova topologia da rede, a tabela de cada neurônio no reconhecimento do xor e o plano de decisão obtido com o processo.

A modificação dos pesos no perceptron é feita pela equação da fórmula 3.8, que é uma variante da equação da atualização supervisionada dos pesos (fórmula 3.7), onde γ é um número positivo menor que 1 e determina a velocidade da adaptação do neurônio. A modificação dos pesos é feita sempre em função do número da interação (t) da rede.

$$w_{ij}(t+1) = w_{ij}(t) + \gamma(d_i(t) - y_i(t))x_j(t)$$

FÓRMULA 3.8- Atualização supervisionada dos pesos no perceptron

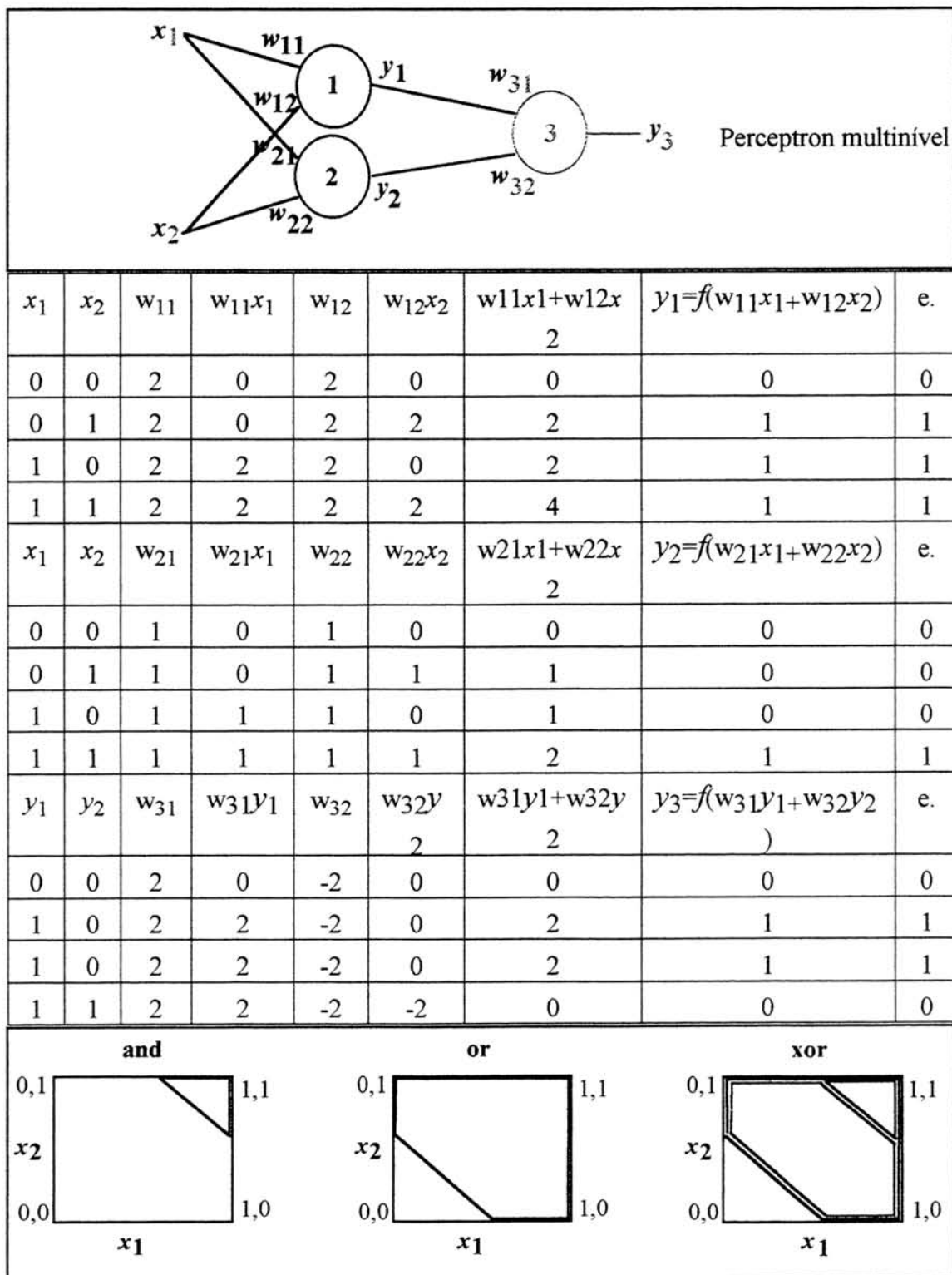


FIGURA 3.10- Reconhecimento com perceptron multinível

O Perceptron tornou-se importante por ser uma das primeiras redes neurais artificiais a utilizar as funções de aprendizado e reconhecimento inspiradas nos modelos

biológicos, dando margem ao desenvolvimento de vários modelos de redes neurais artificiais baseados em sua arquitetura, destacando os modelos Adaline e Back-propagation, largamente utilizados em diversas aplicações.

3.6 O Modelo de Kohonen

As funções ditas de baixo nível, como controles respiratórios, térmicos e metabólicos de um modo geral são atividades programadas geneticamente em muitas redes neurais naturais constituintes do sistema nervoso. Porém, existem as funções de alto nível, que são aquelas aprendidas pelo indivíduo durante a vida, como por exemplo, o reconhecimento de caracteres para leitura. Essas funções são, em sua maioria, configuradas nos neurônios piramidais do córtex cerebral (figura 3.11), cada uma numa área específica.

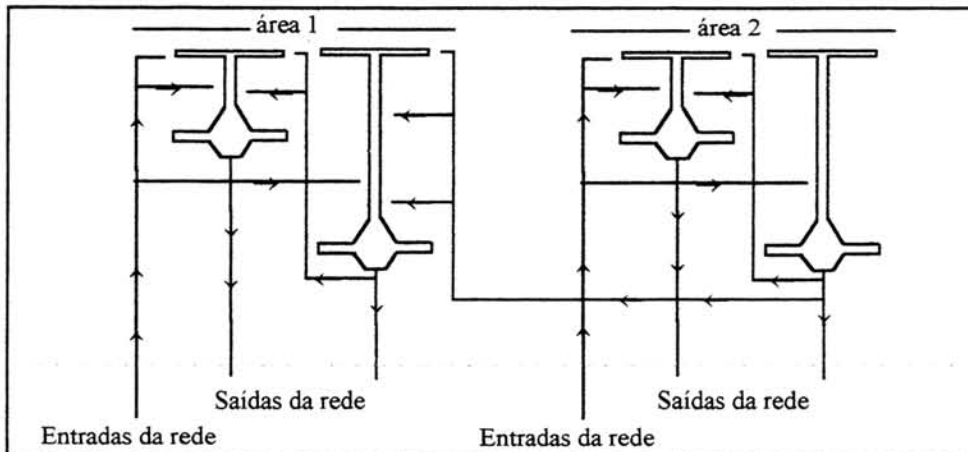


FIGURA 3.11- Topologia dos neurônios piramidais do córtex cerebral

Uma rede neural natural pode aprender de duas formas. A primeira é modificando o fator de transmissão sináptica, também conhecido como peso (w) através da variação na plasticidade sináptica (γ). As redes neurais artificiais baseadas no Perceptron utilizam este método de aprendizado. A outra forma está na modificação anatômica dos neurônios, onde cada neurônio pode prolongar seus dendritos, criar novos botões sinápticos ou expandir os botões existentes, tudo com a finalidade de fazer novas conexões, reforçar as conexões presentes ou substituir as conexões de um neurônio que já não funcionam mais [GUY 84].

Quando um padrão de entrada é apresentado à rede, os neurônios configuram sua anatomia e seus pesos para facilitar as entradas elementares x numa competição em que o neurônio ganhador armazenará o padrão de entrada específico. Por ganhador entende-se que é aquele neurônio que retorna mais potenciais de ação quando apresentado o padrão de entrada específico, ou seja, é o neurônio que melhor reconheceu aquele padrão de entrada.

Utilizando a alta conectividade da rede, o neurônio ganhador influencia os neurônios próximos, através de um processo denominado facilitação que emprega modificações anatômicas no neurônio. A medida em que os neurônios próximos vão sendo influenciados, forma-se uma vizinhança (*cluster*) em torno do ganhador representando o padrão aprendido. Ao apresentar o próximo padrão de entrada, o processo todo se repete e outro *cluster* é formado. Se os padrões possuem características semelhantes, os clusters serão próximos ou até mesmo ligados, caso

contrário, serão distantes. Entre os clusters existem os neurônios intermediários, que representam uma fronteira onde as características dos padrões de entrada se fundem. Isso ocorre porque os *clusters* vão expandindo até chegarem a um grau de proximidade onde os neurônios daquela área sejam influenciados de forma idêntica pelos *clusters* [KOH 88].

Caso alguma área da rede sofra algum dano, inviabilizando um *cluster*, os neurônios restantes se reconfigurarão constituindo um novo cluster, contanto que as entradas sejam novamente apresentadas. Todo o processo de aprendizado é feito simultaneamente com o reconhecimento, conferindo para a rede neural uma plasticidade que a mantém, em tempo real, atualizada com os padrões de entrada. Se os padrões forem mudando a medida em que forem apresentados, os *clusters* espelharão esta mudança criando um mapa na rede neural [KOH 88].

A rede neural de Teuvo Kohonen, também conhecida como mapa de auto-organização (*Self-Organization Feature Map - SOM*) [KOH 88][KOH 90][KOH 92][MAU 93] é uma rede de um único nível, com auto-aprendizado e com propagação lateral da saída, como apresentado na figura 3.12. A propagação lateral vem simular o processo de facilitação do modelo biológico.

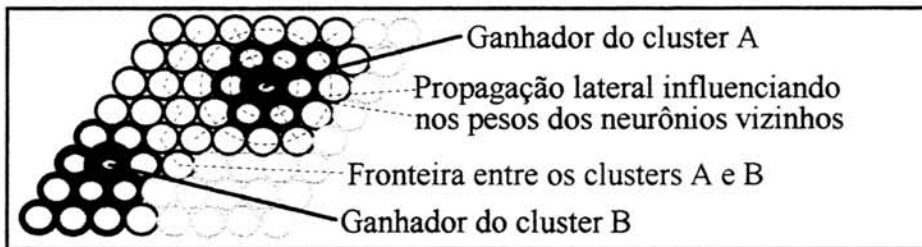


FIGURA 3.12- Rede neural de Kohonen

Seja uma rede neural de Kohonen recebendo um padrão de entrada com n entradas elementares x_j . Denomina-se neurônio ganhador o neurônio i que possui o menor erro, ou seja, a menor distância de seus pesos (w_{ij}) em relação ao padrão de entrada. A fórmula 3.9 calcula a distância d_i dentro de um espaço com dimensão definida por n . Como exemplo, a figura 3.13 apresenta o espaço de cálculo para $n=2$ e $n=3$.

$$d_i = \sqrt{\sum_{j=0}^{j < n} (x_j - w_{ij})^2}$$

FÓRMULA 3.9- Cálculo da distância para estimar o erro no neurônio

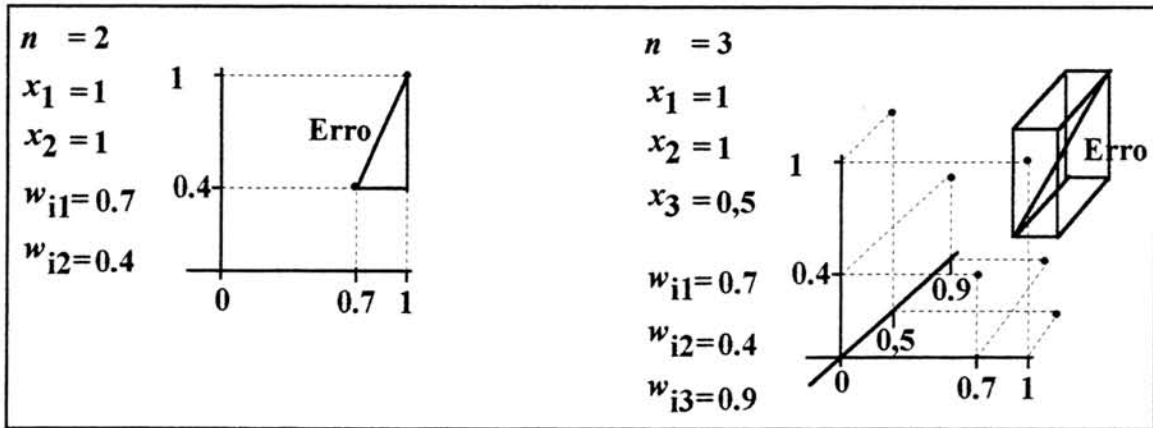


FIGURA 3.13- Espaço de cálculo do erro para rede de Kohonen

As etapas seguintes consistem, respectivamente, em ajustar os pesos do ganhador, visando reduzir ainda mais o seu erro, e ajustar os pesos dos neurônios vizinhos. Os pesos são ajustados de acordo com a fórmula 3.10, onde δ_i é o coeficiente de vizinhança e ε é uma variação do coeficiente de plasticidade sináptica (γ) que decresce a medida em que as interações vão se sucedendo.

$$w_{ij} = w_{ij} + \delta_i \varepsilon (x_j - w_{ij})$$

FÓRMULA 3.10- Ajuste de pesos dos neurônios

Num *cluster*, a influência do ganhador sobre os vizinhos cai exponencialmente em função da distância e do número de interações. Para calcular o valor de δ_i em cada neurônio, os mesmos são dispostos numa matriz bidimensional e suas distâncias em relação ao ganhador são calculadas com o teorema de Pitágoras. O resultado é aplicado na fórmula 3.11, onde d é uma constante que inicialmente define o tamanho médio inicial da vizinhança e vai decrescendo a medida em que aumenta o número de interações. A figura 3.14 apresenta a disposição matricial do neurônios, o cálculo da distância entre o neurônio ganhador e um dos neurônios vizinhos e a curva da influência do ganhador para com os vizinhos.

$$\delta_i = e^{\left(\frac{-distância^2}{2d^2} \right)}$$

FÓRMULA 3.11- Cálculo do coeficiente de vizinhança

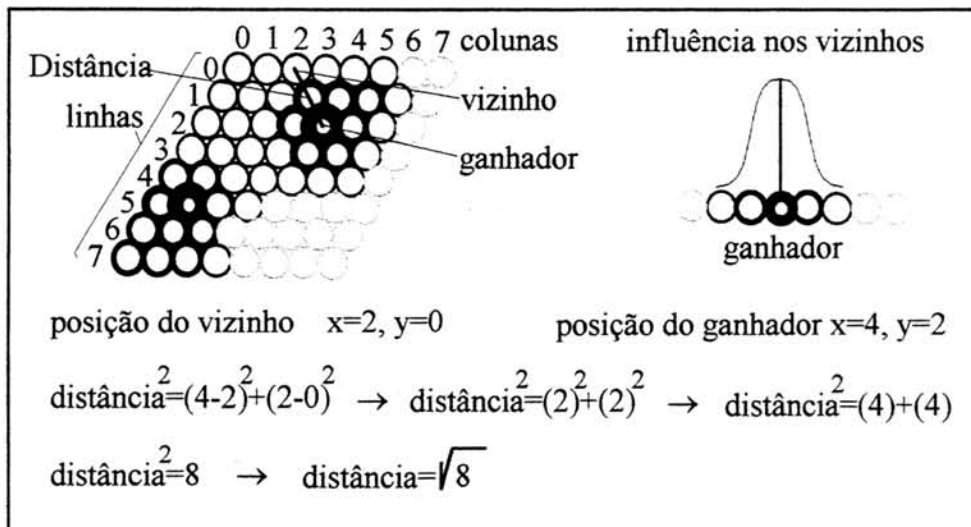


FIGURA 3.14- Disposição matricial dos neurônios

Com base nas equações discutidas, tem-se um algoritmo para produzir mapas auto-organizados constituído das seguintes etapas:

1. Inicialização dos pesos
 - com pequenos valores randômicos,
 - ou com padrões de entrada iniciais,
 - ou com zero.
2. Apresentar um padrão de entrada.
3. Calcular o erro dos neurônios (fórmula 3.9).
4. Selecionar o ganhador (neurônio com o menor erro).
5. Atualizar os pesos dos neurônios (fórmula 3.10).
6. Retornar ao passo ii.

Os mapas auto-organizados de Kohonen conferem às redes neurais grande semelhança do processamento computacional em relação ao processamento biológico, o que lhe confere a atribuição de modelo neural biologicamente plausível. Sua discussão é importante considerando sua destacada capacidade no reconhecimento de padrões, bem como a facilidade em otimizar sua topologia, dependendo dos padrões que ele reconheça. Por estes motivos, seu paradigma foi selecionado para aplicação no OCR deste trabalho, afinal, o modelo biológico do qual os mapas auto-organizados foram extraídos é ainda o melhor reconhecedor de padrões conhecido.

4 Algoritmos Genéticos

O paradigma de Algoritmo Genético pretende mimetizar o funcionamento dos mecanismos naturais de seleção natural e reprodução dos seres vivos. O interesse existente neste paradigma se justifica pela simples observação dos seres vivos no meio ambiente [RIB 94][LAN 94]. A medida em que este meio ambiente vai se modificando, os seres vivos tendem a adaptar-se à nova realidade utilizando um mecanismo descoberto por Darwin denominado seleção natural [KEN 93]. Esse mecanismo otimiza uma população de indivíduos, tornando-os mais adaptados às condições do meio. Esse processo de otimização atrai a pesquisa no intuito de aplicar essa tecnologia para otimização de sistemas, estruturas e procedimentos melhorando, desta forma, seus desempenhos [BRI 92][DAV 91].

Neste trabalho, o paradigma de Algoritmo Genético é empregado para otimizar o sistema de Reconhecimento Ótico de Caracteres - OCR desenvolvido com o paradigma de redes neurais artificiais, utilizando o modelo *Self-Organization Feature Map* de Teuvo Kohonen [KOH 88], descrito no capítulo 5, possibilitando um considerável ganho de desempenho. Os resultados desta hibridização estão descritos no capítulo 6 demonstrando assim a melhoria de desempenho obtido.

4.1 Características Básicas

O processo de seleção natural, princípio no qual repousa os fundamentos de Algoritmos Genéticos, atua na natureza garantindo que os seres vivos mais adaptados às modificações ocorridas no meio ambiente tenham melhores condições de sobrevivência e reprodução [BRI 92][LAN 94]. Dentro do meio ambiente, existem várias espécies de indivíduos, cada espécie contendo um fenótipo característico. Por fenótipo entende-se o conjunto de características de uma espécie que possibilita sua sobrevivência [ROC 93]. Isto significa que a adaptação ao meio não leva a um único resultado, mas sim a vários resultados alternativos que garantem uma boa interação dos indivíduos com o meio. Caso o meio ambiente venha a modificar-se, os fenótipos dos indivíduos das várias espécies também deverão modificar-se para adaptarem-se à nova realidade. Neste momento é que muitas espécies sucumbem, por não possuírem uma plasticidade genética tal que possibilite a mudança esperada em seus fenótipos. Esta seleção natural, promovida atualmente pelo progresso endógeno do homem é responsável pela extinção de várias espécies que não tiveram a capacidade ou tempo de se adaptarem [RIB 94][DAV 91].

Dentro de cada espécie existem populações de indivíduos. População é um conjunto de indivíduos da mesma espécie, que vivem numa área determinada [SRI 94]. Os fenótipos destes indivíduos são praticamente idênticos. Existem pequenas diferenças que ocorrem em grupos de indivíduos diferenciando-os por raças e existem ainda diferenças mais sutis que separam individualmente os membros da população. Este princípio denomina-se diversidade genética e garante uma melhor adaptação da população às mudanças do meio [DAV91]. A figura 4.1 apresenta uma possível diversidade genética dentro de uma espécie.

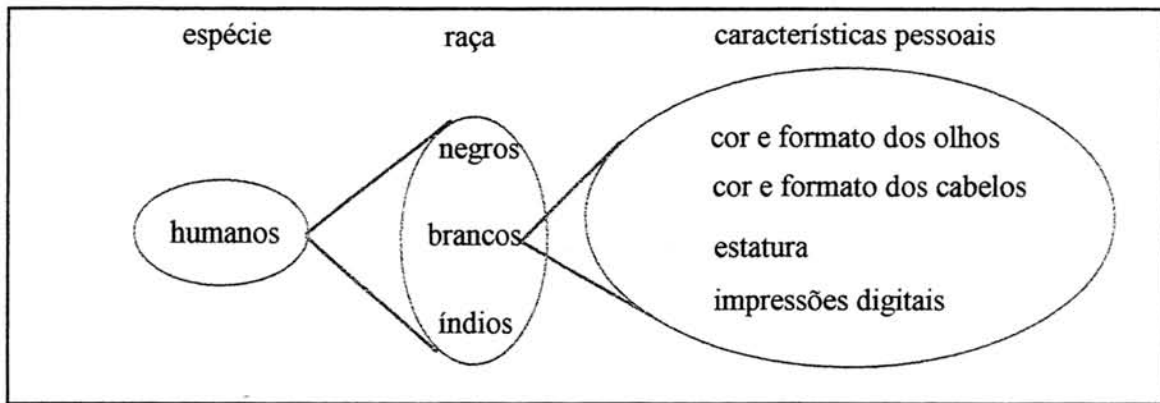


FIGURA 4.1- Exemplo de diversidade genética

O fenótipo de cada indivíduo é construído a partir de seu genótipo. Por genótipo entende-se como o conjunto de informações responsáveis pela construção do fenótipo [RAM 94][LAW 92]. Estas informações estão representadas nos códigos genéticos presentes nos cromossomos de cada indivíduo, que são constituídos por riboses do tipo ácido desoxirribonucleico (DNA) e ácido ribonucleico (RNA). Cada partícula elementar do DNA ou RNA é chamada de gene. A leitura dos genes possibilita a construção de proteínas que serão utilizadas na(s) célula(s) do indivíduo. O processo de síntese protéica segue a hipótese de Assunção, criada por Rodney Brooks. Segundo esta hipótese, a leitura do código genético leva a construção de substâncias elementares que constituirão outras mais complexas e assim sucessivamente até constituir um indivíduo completo. A figura 4.2 apresenta este comportamento emergente que serve de elo do genótipo para o fenótipo.

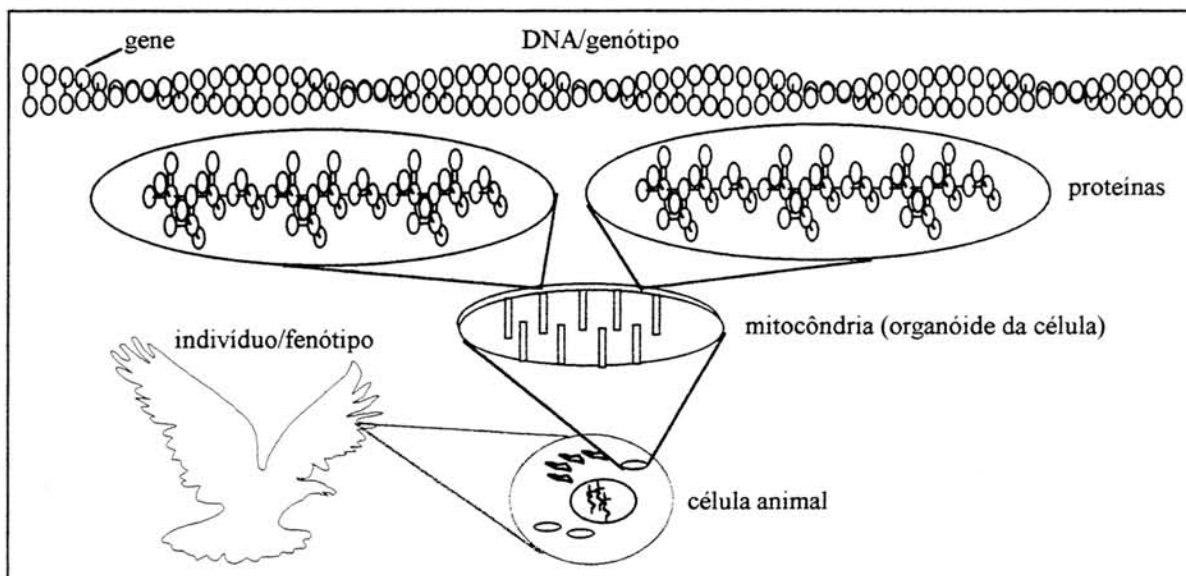


FIGURA 4.2- Exemplo de comportamento emergente

Caso ocorra alguma mudança no meio ambiente, os indivíduos não tem como se adaptar a ele diretamente. Porém, aquele indivíduo que possui alguma característica que o possibilite viver um pouco mais neste novo meio, reproduzirá mais e passará esta boa característica para a sua descendência. Os indivíduos que não possuem nada que os favoreçam no novo meio, viverão e reproduzirão cada vez menos, reduzindo a incidência de seus fenótipos na população. Desta forma, olhando de um

ponto de vista mais abstrato, ou seja, olhando a população como uma entidade, observa-se que esta entidade adapta-se às novas condições. Pode acontecer também que a população não tenha condições de adaptar-se resultando em sua extinção.

Cada característica do indivíduo é representada por um segmento do código genético, denominado esquemático. O ideal num processo evolutivo seria a transferência apenas dos bons esquemáticos dos indivíduos para suas descendências. A natureza, no entanto não trabalha de forma imediata, tendo como único recurso as gerações de indivíduos para transferir estes bons esquemáticos. Por geração entende-se cada ciclo de reprodução que ocorre numa população. Existem diversas formas de reprodução inerentes a cada espécie. Dependendo da forma, um indivíduo pode transmitir todo o seu genótipo ou simplesmente parte dele para a descendência [FOG94].

Uma baixa plasticidade genética pode ser um dos fatores que conduza uma espécie a extinção. Por exemplo, os dinossauros extingiram-se devido à mudanças bruscas no meio ambiente. A velocidade do meio foi maior que a velocidade com que esses animais podiam se adaptar. Talvez fosse necessário um número de gerações muito maior para que seus genótipos conseguissem levar a bons fenótipos. Resumindo, eles não foram rápidos o bastante. No caso dos vírus, a plasticidade é elevadíssima, tornando uma população de vírus imune a uma vacina em poucas gerações, o que dificulta o desenvolvimento de vacinas para certos tipos de infecções viróticas.

Outro fator decisivo para sobrevivência da espécie relaciona-se com os mínimos locais encontrados pela natureza, aos quais os indivíduos se prendem. Por exemplo, os insetos evoluíram para um sistema respiratório pouco eficiente para seres vivos de grandes dimensões. Isto impede que os insetos aumentem de tamanho e se tornem mais complexos. Para que a espécie possa evoluir ainda mais, necessitaria retornar pela escala evolutiva, optar por um outro sistema respiratório e então recomeçar o processo novamente. Como a natureza não trabalha desta forma, não pode retornar ou simplesmente abandonar uma tecnologia em detrimento de outra. O sistema respiratório dos insetos é um exemplo clássico de mínimo local, que impede a espécie de evoluir mais. A mutação é um fenômeno não controlado que pode tirar um indivíduo de um mínimo local. Como o processo mutagênico não é controlado, o fenótipo resultante é aleatório, e o indivíduo pode adquirir uma condição pior do que já tinha. De um modo geral, a mutação é responsável pelas grandes diversidades genéticas observadas atualmente nas várias espécies [DAV91][FOG94].

4.2 Versão Computacional do Algoritmo Genético

Com a abstração das características levantadas pela observação dos modelos biológicos de seleção natural e reprodução, John Holland, em 1975 [HOL 75], elaborou uma versão computacional destes modelos chegando a um modelo formal para algoritmo genético. Lawrence Davis, em 1991, propôs a implementação do algoritmo genético em três módulos principais que são o módulo de avaliação, o módulo de população e o módulo de reprodução [DAV 91].

4.2.1 Módulos de população e avaliação

O módulo de população compreende a definição do meio ambiente e da população de indivíduos. O meio ambiente deve ser o objeto da otimização [MAZ

94][RIB 94][DAV 91][AUS 90]. Por exemplo, ao otimizar o funcionamento de uma rede neural destinada ao reconhecimento de padrões, deve-se ter em mente o que se deve melhorar no modelo. O ambiente pode ser representado pelos padrões de entrada a serem apresentados para a rede.

Para a população, deve-se definir [AUS 90][MAR 91]:

- a técnica de representação dos indivíduos, seu fenótipo e genótipo;
- a técnica de inicialização dos indivíduos, ou seja, a primeira geração;
- a técnica de seleção de pais, seleção natural;
- a técnica de reprodução, deve-se fazer a escolha da técnica e parametrização do tamanho da descendência;
- a técnica de deleção dos indivíduos, substituição dos pais pelos filhos na população para troca de gerações.

A técnica de representação dos indivíduos define como será o fenótipo a partir da construção do genótipo. O genótipo (código genético) encontra-se nos cromossomos. Cada indivíduo pode ter de um a n cromossomos, dependendo de sua espécie, sendo que os indivíduos da mesma espécie possuem o mesmo número de cromossomos. Davis [DAV 91] utiliza em algumas de suas implementações indivíduos com apenas um cromossomo. Os genes do cromossomo assumem no modelo computacional uma configuração numérica que podem representar um modelo de DNA ou RNA, ou qualquer outro modelo, sendo que os indivíduos da mesma espécie usam a mesma configuração. Uma configuração muito utilizada em algoritmos genéticos é a binária com fenótipo implícito no gene onde as características do indivíduo não estão codificadas no cromossomo, porém elas são ativadas dependendo do estado do gene, que assume valores binários [SRI 94][DAV 91][FOG 94]. A diferença entre um cromossomo binário comum e um cromossomo binário com fenótipo implícito no gene é que o primeiro constrói o fenótipo e o segundo apenas ativa características de um fenótipo pré-estabelecido [RIB 94], como mostra a figura 4.3 que apresenta também um exemplo de meio ambiente baseado nos trabalhos de Kosko [KOS 91], [WIG 92] e Ramos [RAM 94][RAM 95c], onde o meio ambiente é representado pelas características de um veículo que se desloca sobre o plano cartesiano.

A inicialização dos indivíduos define qual o número inicial de indivíduos na população e quais serão os valores de seus genes na primeira geração. O procedimento usual é inicializar aleatoriamente cada gene dentro de seu domínio de valores. A inicialização aleatória dos genes possibilita uma grande diversidade genética que favorece a seleção de pais para reprodução [BRI 92][ROC 93][DAV 91].

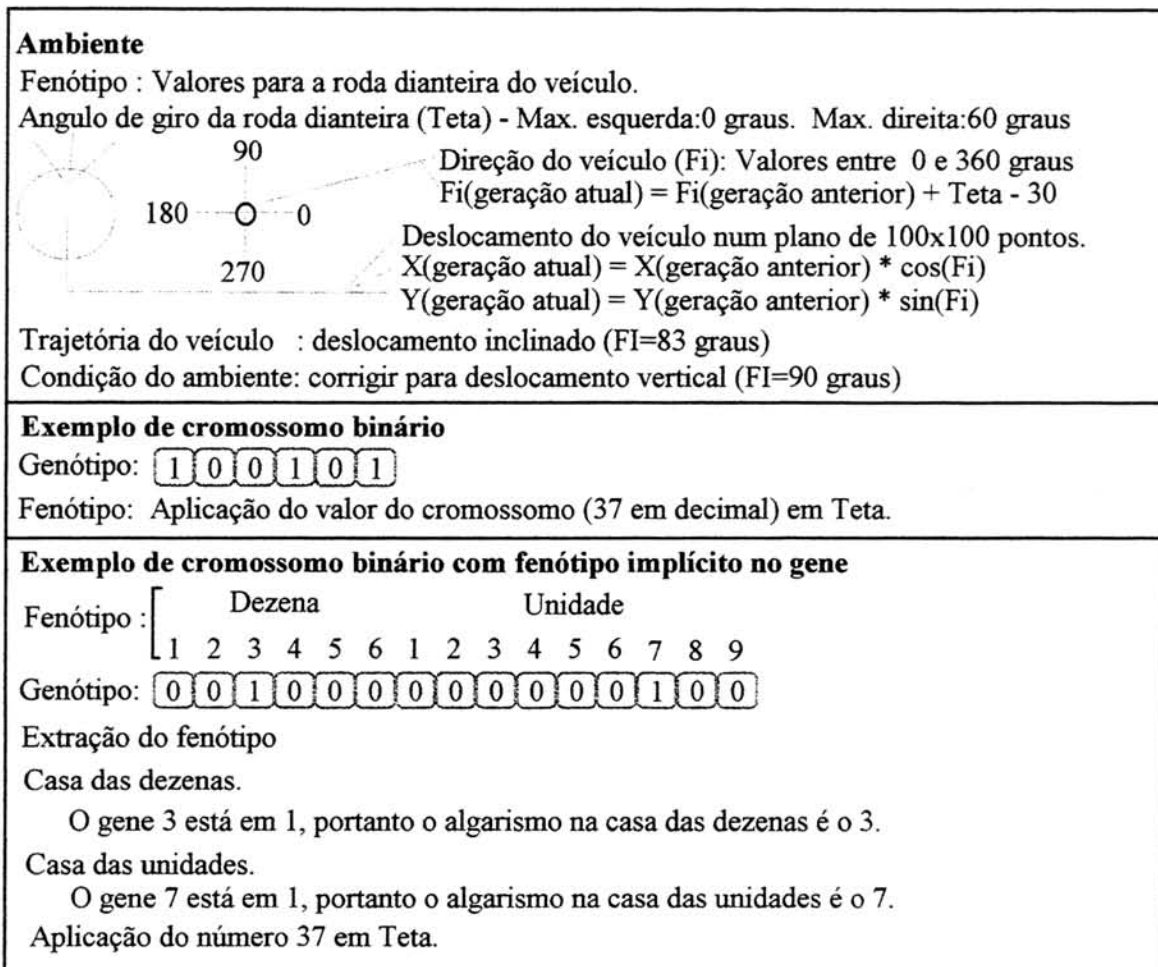


FIGURA 4.3- Exemplo de cromossomos binários

Para que os indivíduos sejam selecionados para reprodução é necessário que os seus respectivos graus de adaptação ao meio sejam quantizados. Este valor, o ajuste, serve de parâmetro para as técnicas de seleção de pais empregadas nos modelos computacionais. Para obtenção do ajuste, faz-se necessário a interação de cada indivíduo verificando o quanto seu fenótipo satisfaz as condições do meio. De acordo com o exemplo da figura 4.3, quanto mais para perto dos 90 graus o indivíduo conduzir o veículo (fenótipo próximo dos 37 graus), maior será o seu ajuste [DAV 91][ROC 93][SRI 94][BRI 92].

A seleção mais simples e menos plausível, do ponto de vista natural, é a classificação dos indivíduos por ordem de ajuste e consequente seleção dos indivíduos com ajuste mais elevado [WIG 92]. A técnica da roleta, empregada por Davis em seus trabalhos [DAV 91], utiliza a aleatoricidade na escolha dos pais, a exemplo da natureza que pode favorecer um indivíduo mais do que o outro, independente de seu ajuste [SRI 94]. Esta técnica sorteia os indivíduos da população para reproduzirem, com a ressalva de que as chances do indivíduo ser sorteado são proporcionais ao seu ajuste e, conforme o valor do ajuste, o indivíduo tem possibilidade de ser escolhido mais de uma vez. A figura 4.4 exemplifica o funcionamento desta técnica.

A técnica da reprodução compreende a definição do modelo de reprodução empregado e da quantidade de filhos gerada pelos indivíduos por geração, lembrando

que estes parâmetros servem para todos os indivíduos. As técnicas de reprodução dividem-se em reprodução assexuada e sexuada [FOG 94].

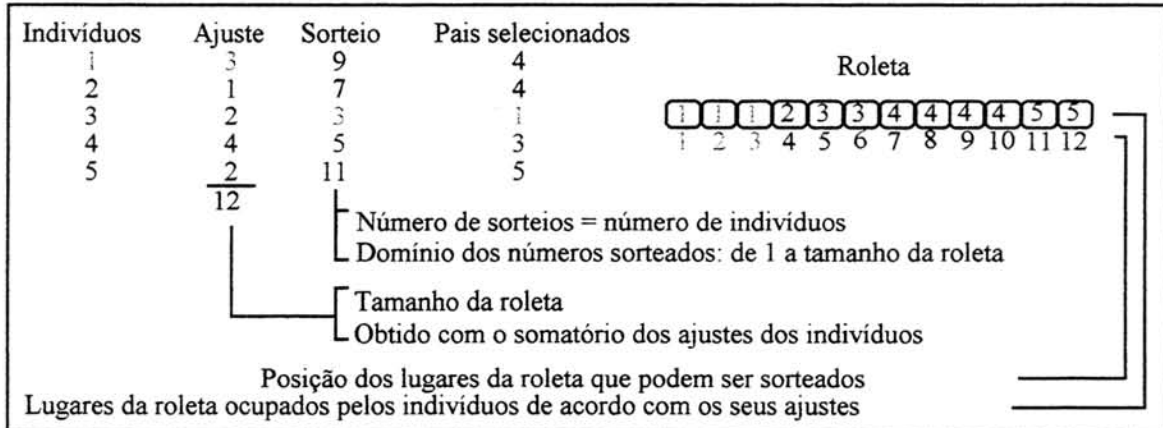


FIGURA 4.4- Técnica de seleção da roleta

Na forma assexuada os indivíduos simplesmente duplicam seus cromossomos e transferem esse material para seus filhos. Portanto, a descendência será composta de clones dos indivíduos escolhidos para reproduzirem. Apesar de manter na população os indivíduos com ajuste elevado, esse ajuste permanecerá estacionário, impedindo uma melhor adaptação da população ao meio ambiente. Outra clara desvantagem está na perda da diversidade genética da população, conduzindo a mesma, com o passar das gerações, a possuir seus indivíduos com um único fenótipo. Caso haja uma mudança no meio ambiente, esta população não terá plasticidade genética para se readaptar às novas condições.

No caso da reprodução sexuada, existe uma mistura do material genético de dois indivíduos para gerar um novo cromossomo que possui características comuns aos dois. Esta técnica possui muitas variações, mas todas elas baseiam-se na divisão dos genes dos pais, duplicação dos segmentos resultantes da divisão e posterior reagrupamento misturando os segmentos dos pais. A grande vantagem desta técnica está na diversidade genética mantida ao criarem-se novos fenótipos a cada geração, garantindo uma alta plasticidade genética em caso de mudança nas condições do meio.

A quantidade de filhos gerada por reprodução garante o controle populacional [RIB 94][BRI 92]. Este fator deve ser considerado levando em conta as condições de memória, velocidade do sistema e facilidades na implementação do modelo. O procedimento mais comum é a manutenção de uma taxa constante de indivíduos na população promovendo um número de clones equivalente a quantidade de vezes que o indivíduo foi selecionado, (reprodução assexuada) ou 2 filhos por casal (reprodução sexuada), conforme exemplos apresentados na figura 4.5.

Para um número ímpar de indivíduos, aquele que possui o ajuste mais alto é preservado para a próxima geração e, além de sua preservação, existe ainda a possibilidade de cruzá-lo com outro indivíduo, caso ele tenha sido selecionado mais de uma vez. Esse procedimento garante um ajuste cada vez mais alto no decorrer das gerações, impedindo que o melhor indivíduo se perca por cruzamentos mal sucedidos.

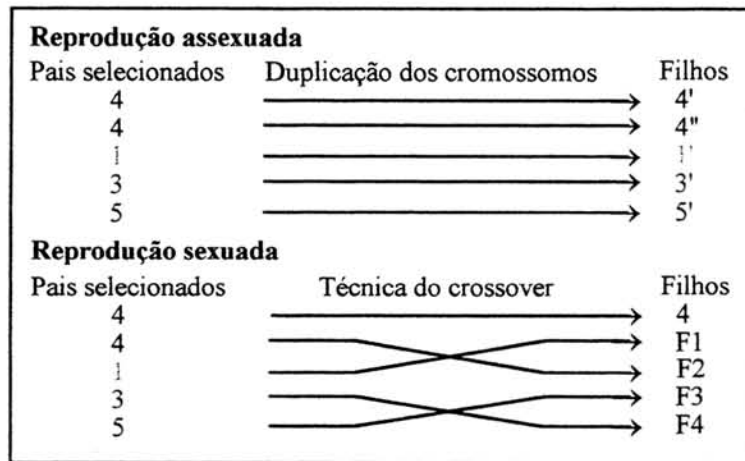


FIGURA 4.5- Técnicas de reprodução e número de filhos

As duas técnicas de reprodução podem vir acompanhadas de um processo mutagênico, para tirar uma população de um mínimo local. A mutação pode criar novos cromossomos mesmo com reprodução assexuada. A desvantagem está na impossibilidade de se saber como serão os cromossomos dos filhos, uma vez que possuem características novas não herdadas por nenhum indivíduo [DAV 91][RIB 94][SRI 94].

A deleção dos indivíduos corresponde na sua substituição pelos seus filhos na população. Os indivíduos podem ser eliminados de acordo com o seu ajuste, quanto maior o ajuste mais tempo se vive, ou podem ser todos eliminados ao término de uma geração, sobrando apenas os filhos. A segunda forma de deleção é empregada quando se deseja manter o número de indivíduos constante ao passar das gerações. Quando empregado o crossover, geralmente emprega-se as duas formas, especialmente em caso de população com número ímpar de indivíduos [DAV 91], segundo exemplo na figura 4.5.

4.2.2 Módulo de reprodução

Neste módulo são parametrizados os modelos de reprodução escolhidos para a população. Para reprodução assexuada é definido o número de filhos de acordo com o ajuste do indivíduo. No caso de reprodução sexuada é definido o tipo de *crossover* e, quando a reprodução comporta mutação, é definido o tipo de mutação empregado e a taxa de mutação [SRI 94][RIB 94][BRI 92].

Os tipos de *crossover* são classificados quanto a dimensão, unidimensionais e bidimensionais, e quanto a quantidade de segmentos divididos em *one point crossover*, *two point crossover* e *multi point crossover*, ou seja, *crossover* de um ponto, *crossover* de dois pontos e *crossover* de múltiplos pontos [FOG 94][DAV 91]. Quanto à escolha do segmento a ser dividido, o *crossover* pode ser aleatório, qualquer segmento pode ser escolhido, ou inteligente, separa os bons esquemáticos na divisão. Os esquemáticos são obtidos através do *template matching* nos indivíduos selecionados que possuam ajuste elevado. Os modelos *one point crossover* e *two point crossover* são apresentados nas figuras 4.6 e 4.7, respectivamente.

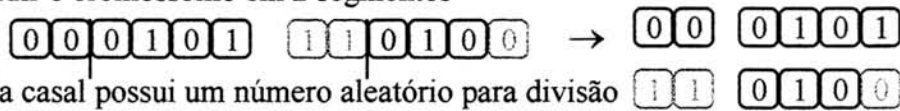
Identificação dos bons esquemáticos por template matching

Indivíduo/Cromossomo	Ajuste
4 $\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$	4
1 $\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$	3

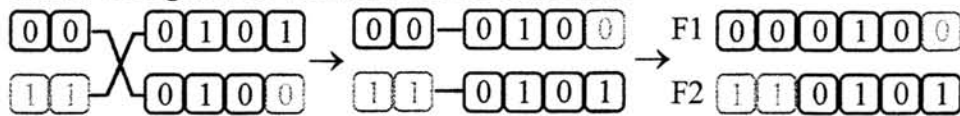
São iguais no primeiro e segundo indivíduos (responsáveis pelo ajuste elevado).
Modelo para um ajuste elevado: ##010#, onde # pode ser qualquer valor.
A descendência, para garantir um ajuste elevado, precisa seguir o modelo.

Etapas de um crossover de um ponto unidimensional

- 1) Dividir o cromossomo em 2 segmentos



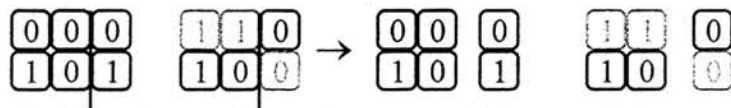
- 2) Trocar os segmentos de um indivíduo com outro



Etapas de um crossover de um ponto bidimensional

- 1) Colocar os cromossomos numa matriz 3x2 e dividir em 2 segmentos

A matriz deve ser a mesma para os dois indivíduos do casal



- 2) Trocar os segmentos de um indivíduo com outro

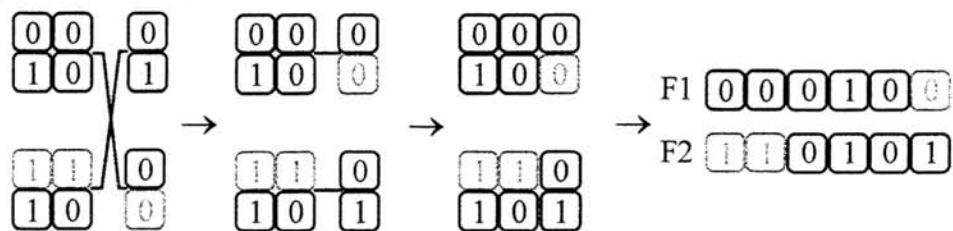


FIGURA 4.6- Identificação dos esquemáticos e crossover de um ponto

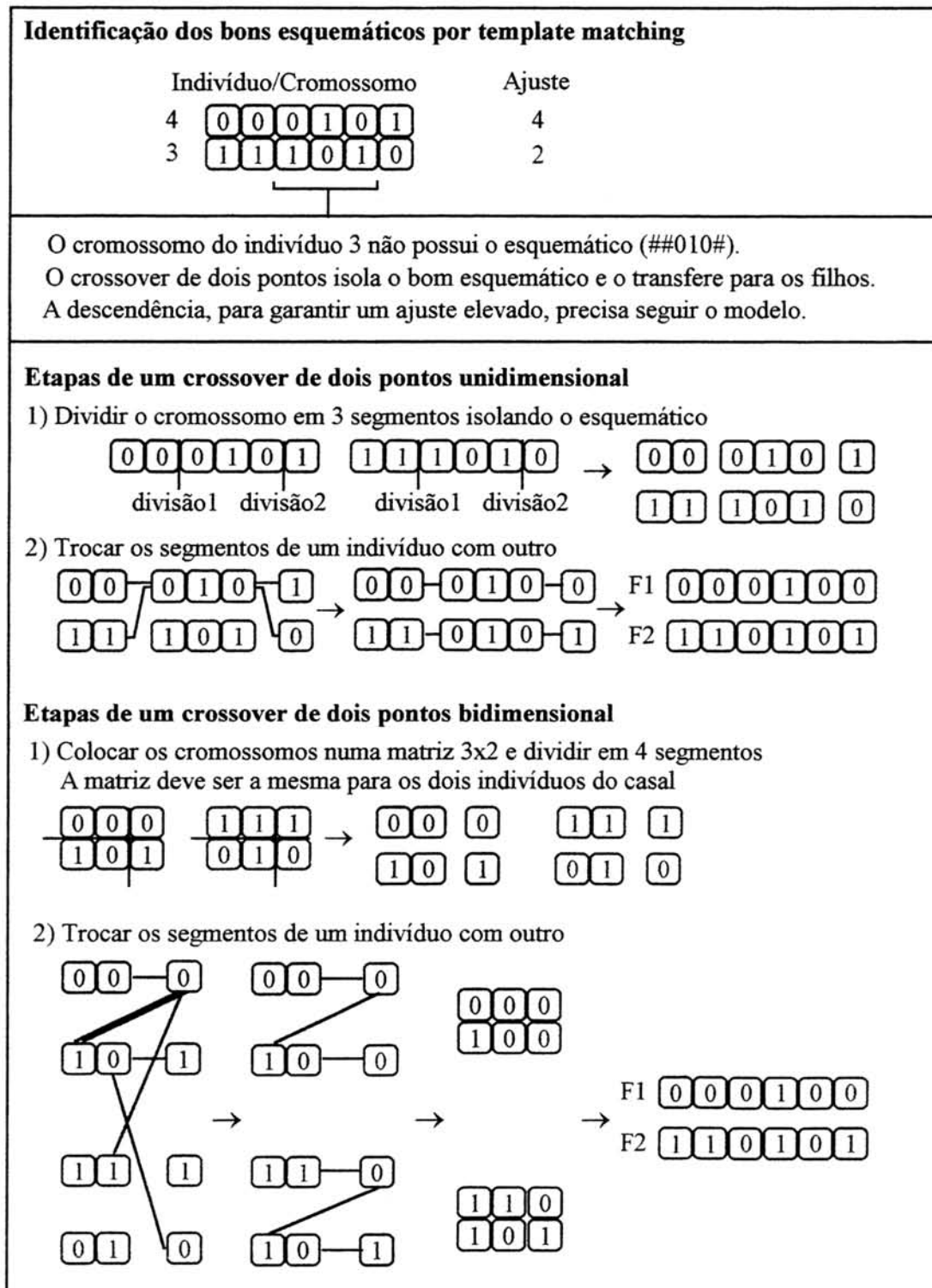


FIGURA 4.7- Funcionamento do crossover de dois pontos

A medida em que as gerações passam, a flexibilidade da população em adaptar-se às condições do meio ambiente diminui devido à redução da plasticidade genética. Isto ocorre porque a medida em que as gerações se sucedem, mais e mais indivíduos da população vão adquirindo os bons esquemáticos, responsáveis pelo fenótipo que melhor se adapta ao meio, tornando os indivíduos cada vez mais parecidos e acabando com a diversidade genética. Dependendo das condições do meio ambiente, do tamanho da população e, principalmente, do tipo de genótipo empregado, é bem

provável que a plasticidade genética se esgote antes mesmo que a população se adapte completamente.

Foi desenvolvido um sistema em Algoritmos Genéticos para o controle de trajetórias do veículo [RAM 94][RAM 95c] exemplificado na figura 4.8, objetivando interceptar um alvo parado num ponto do plano cartesiano. Quando a população era muito pequena (abaixo de 30 indivíduos), o veículo não chegava até o alvo porque os indivíduos rapidamente adquiriam o mesmo valor, este insuficiente para redirecionar o veículo. A medida em que se aumentava a população, os sucessos se confirmavam. O *crossover* tinha agora subsídios, cromossomos, para obter um ajuste melhor. Outro problema observado foi quando o alvo moveu-se pelo plano. A medida em que as gerações passavam, o veículo tinha uma dificuldade crescente em perseguir o alvo, afinal, as condições do meio estavam mudando e a plasticidade genética estava se esgotando. Concluiu-se que o *crossover* era muito eficaz para convergir um algoritmo genético visando uma condição específica do meio ambiente. Quando esta condição muda, o *crossover* precisa de muitas gerações e uma grande população com elevada diversidade genética. Infelizmente estes recursos não são desejados em sistemas computacionais que possuem memória e tempo de execução restritos às necessidades da aplicação.

A mutação é uma técnica que veio solucionar parte dos problemas levantados pelo *crossover*. Esta técnica deve ser empregada em algoritmos que convergem rapidamente para um valor específico e em casos de constantes mudanças nas condições do meio. No caso da rápida convergência, é possível que o melhor ajuste não seja ótimo, mas sim um mínimo local causado por uma baixa diversidade genética. O processo mutagênico aumenta a diversidade, colocando novos fenótipos para serem testados no ambiente. Para as condições instáveis de um meio-ambiente, a mutação mantém a flexibilidade do algoritmo em adaptar-se com o passar do tempo [RIB 94][SRI 94][RAM 94][RAM 95c].

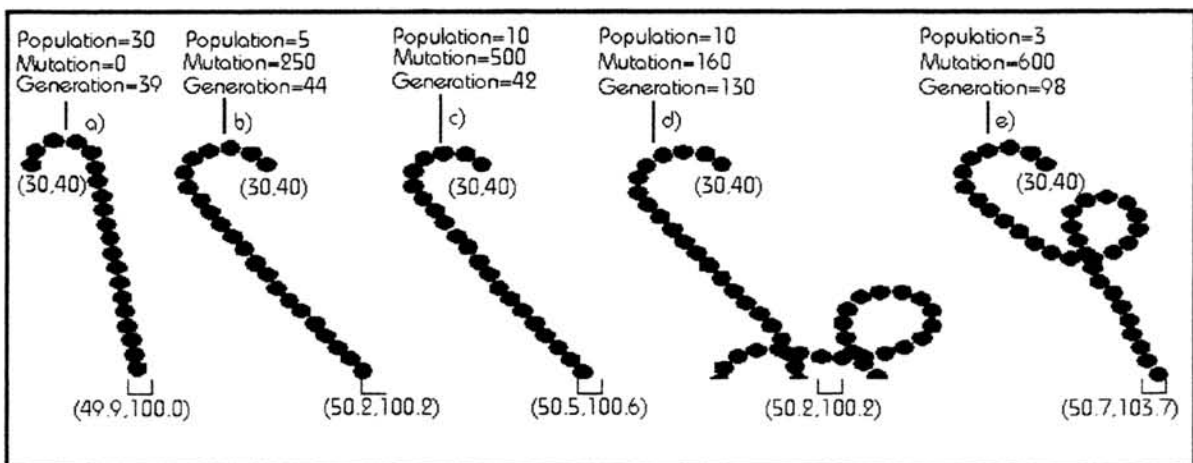


FIGURA 4.8- Comparação do tamanho da população

O processo mutagênico mais comum, empregado em algoritmos genéticos, é o de troca aleatória do valor do gene, onde este troca de 0 para 1 ou de 1 para 0. Este processo possui uma taxa de mutação que determina a probabilidade de um gene ser modificado no cromossomo. Cada gene recebe um número aleatório que, se for menor ou igual a taxa, indica mutação [DAV 91], como demonstra a figura 4.9.

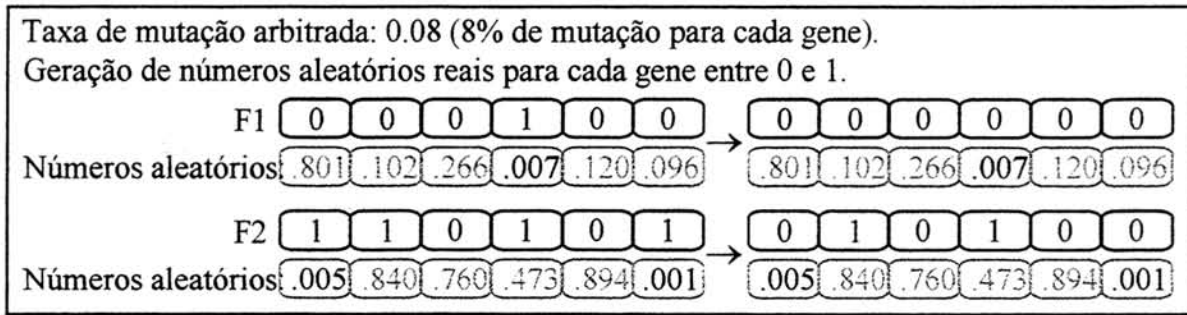


FIGURA 4.9- Mutação nos novos indivíduos

4.3 Aplicações do Paradigma de Algoritmo Genético em Problemas de Otimização

O emprego de Algoritmos Genéticos em problemas de otimização encontra aplicação em diversas áreas, tais como:

- áreas industriais [ROC 93];
- sistemas de controle híbridos visando melhorar o desempenho dos sistemas existentes [RAM 94][RAM 95a][RAM 95b];
- área financeira e sistemas preditivos para análise de capitais, flutuação de títulos no mercado, aplicação no problema do caixeiro viajante [TRE 92] [SAN 94];
- área educacional com sistemas que contribuem para construção de regras para sistemas especialistas [RIB 94];
- na área de produção, com auxílio de controle de estoque e em projetos de engenharia [LAN 93];
- em funções de criptografia [LAN 94].

Existe inclusive um sistema projetado para otimização da análise de estruturas do DNA. Neste caso, sua aplicação ajudaria o homem a aprender mais sobre o paradigma que gerou os próprios Algoritmos Genéticos.

5 Implementação do Modelo de Kohonen

A implementação do modelo neural de Kohonen em um sistema de OCR foi aplicada sobre a base de dados do correio britânico para reconhecimento de caracteres [GUA 92]. Nesta base de dados, constituída de um arquivo texto, estão digitalizados 3000 algarismos. São 300 tipos de cada algarismo dispostos em ordem crescente (0,1,2...), repetindo esta sequência por 300 vezes. Os algarismos digitalizados foram escritos a mão e estão dispostos numa matriz binária de 16 colunas por 24 linhas. Muitos algarismos apresentam diversas falhas decorrentes de um mal posicionamento dos caracteres na matriz, ruídos decorrentes de falhas na digitalização e má formação dos caracteres inerentes à sua natureza, ou seja, porque foram escritos a mão. A figura 5.1 apresenta os 10 primeiros algarismos desta base de dados.

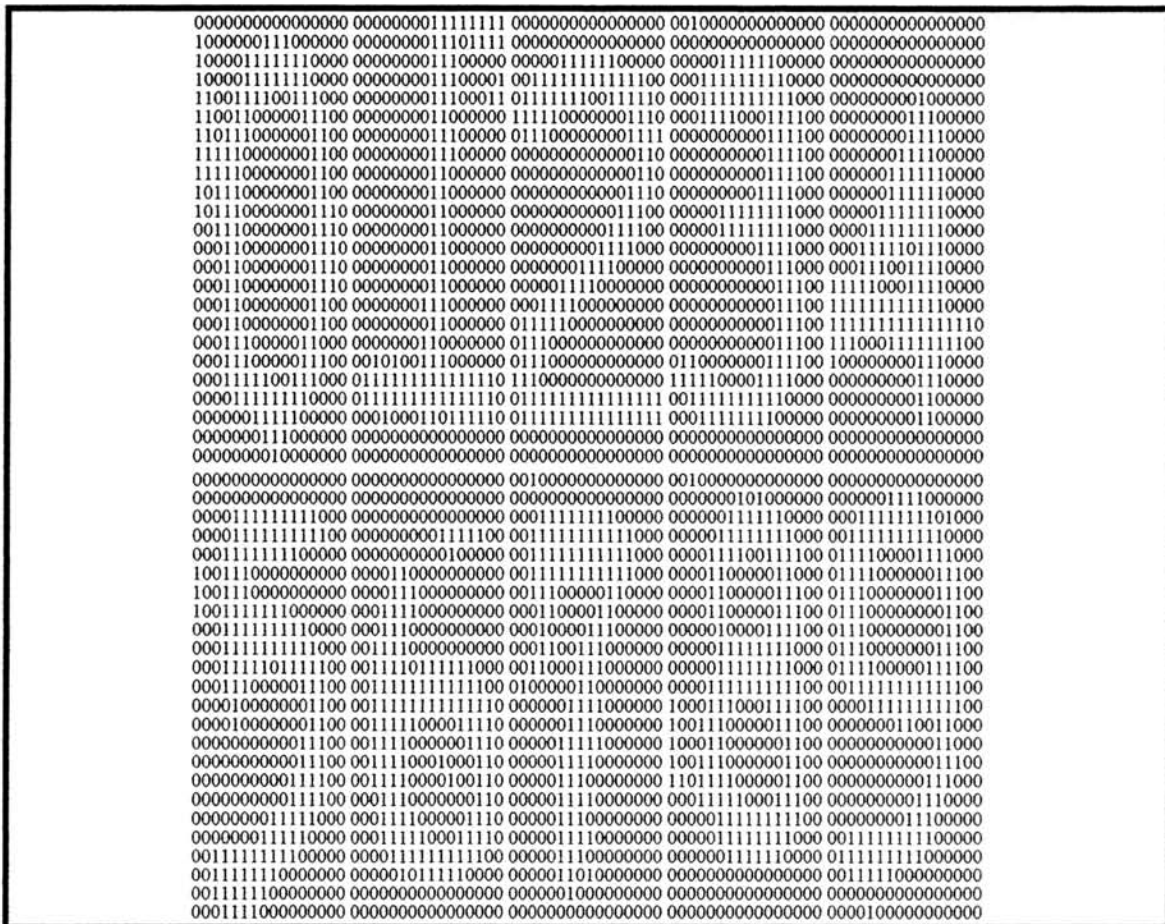


FIGURA 5.1- Exemplo de algarismos do correio britânico

Parte do arquivo correio britânico será utilizada para o treinamento da rede neural e a outra parte será empregada para o reconhecimento. O sistema desenvolvido aceita padrões de entrada sem a necessidade de pré-processamento para melhorar a qualidade dos algarismos [RAM 95a][RAM 95b]. Foi observado, inclusive, que o ruído dos padrões de entrada aumentavam a diversidade das características de cada algarismo aprendido facilitando a etapa de reconhecimento.

O modelo neural de Kohonen gera agrupamentos de neurônios responsáveis por cada algarismo aprendido. Considerando que esta rede é do tipo auto-aprendizado, não se tem o controle sobre a forma como será realizado tal agrupamento e, inclusive,

desconhece-se os neurônios que aprenderam um algarismo específico, dificultando a etapa de reconhecimento.

Para solucionar o problema do reconhecimento, desenvolveu-se um algoritmo de classificação dos agrupamentos. Este algoritmo utiliza os mesmos padrões de entrada para classificar os grupos de neurônios e determinar as regiões da rede responsáveis pelo aprendizado de cada algarismo. Pode-se considerar esta técnica como um processo de pós-supervisionamento onde, ao invés de induzir o aprendizado da rede com uma saída desejada, busca-se a saída desejada dentro da rede configurada.

O modelo de Kohonen proposto para o reconhecimento ótico de caracteres é constituído de 25 neurônios, cada um contendo 384 entradas [RAM 95a][RAM 95b]. Cada entrada corresponde a um bit da matriz onde encontra-se digitalizado cada algarismo (a matriz possui dimensão de 16x24, ou 384 bits). A memória é localizada, ou seja, cada neurônio da rede armazena as características de um algarismo inteiro [KOH 88][KOH 90].

O arquivo de caracteres do correio britânico é dividido em duas partes. A primeira parte é utilizada para treinamento da rede e a segunda é empregada no processo de reconhecimento. Como não existem dois padrões de entrada iguais em todo o arquivo, exclui-se a possibilidade de um mesmo padrão ser apresentado no treinamento e também no reconhecimento.

5.1 Treinamento da Rede

O treinamento da rede de Kohonen é composto das seguintes etapas, sendo discutidas a seguir [LIP 87][KOH 88].

1. Inicialização dos pesos
 - com pequenos valores randômicos
 - ou com padrões de entrada iniciais
 - ou com zero.
2. Apresentar um padrão de entrada.
3. Calcular o erro dos neurônios (fórmula 3.9).
4. Selecionar o ganhador (neurônio com o menor erro).
5. Atualizar os pesos dos neurônios (fórmulas 3.10 e 3.11).
6. Retornar ao passo 2 até o fim dos padrões de entrada.

Na etapa 1, foi estipulado a inicialização com zero para o vetor de pesos w_{mn} , onde m equivale aos 25 neurônios da rede e n representa o número de entradas elementares de cada padrão, ou seja, os 384 bits da matriz onde se encontra digitalizado o algarismo. Todas as inicializações com valores diferentes de 0 resultaram em desempenho inferior ao desempenho obtido utilizando a inicialização com valores nulos.

A etapa 2 refere-se à leitura do padrão de entrada usado para o treinamento. O padrão de entrada é movido para o vetor x_n , onde n é o número de entradas elementares de cada padrão. A visualização do padrão de entrada é dada pela matriz de dimensão 16x24. Para a rede neural, no entanto, o padrão apresenta-se como um vetor composto de 384 entradas. Este vetor é que será movido para x_n . Depois da

apresentação do padrão de entrada, o algoritmo avança para o próximo padrão (o próximo caractere do arquivo correio britânico). Desta forma, quando a etapa 2 for novamente acionada, o padrão de treinamento será outro.

A etapa 3 consiste no cálculo dos erros dos neurônios sobre o padrão de entrada armazenado em x_n para identificar o neurônio ganhador através da fórmula descrita na fórmula 3.9 e, na etapa 4, o neurônio ganhador é selecionado.

Na etapa 5, os pesos dos neurônios são atualizados pelas fórmulas 3.10 e 3.11, respectivamente, alterando os valores da matriz w_{mn} e, na etapa 6, repete-se a partir do passo 2 para ler o próximo padrão de entrada.

5.2 Reconhecimento da Rede

Para o reconhecimento da rede foi desenvolvido um classificador de clusters [RAM 95A][RAM 95b]. Por cluster entende-se o agrupamento de neurônios responsável pelo armazenamento das características de uma classe. Este classificador identifica o cluster que aprendeu as características de determinado algarismo. O classificador tem como entrada a segunda parte do arquivo de caracteres do correio britânico para que, à medida em que os padrões forem apresentados, um controle estatístico determine a taxa de reconhecimento de cada neurônio para cada algarismo, gerando uma tabela de classificação de toda a rede. A seguir, as etapas do processo de classificação dos clusters.

1. Apresentar um padrão de entrada.
2. Calcular o erro dos neurônios (fórmula 3.9).
3. Selecionar o ganhador (neurônio com o menor erro).
4. Incrementar tabela de classificação.
5. Retornar ao passo 2 até o fim dos padrões de entrada.
6. Montar o diagrama com os clusters.

As etapas 1, 2 e 3 são idênticas às etapas de treinamento 2, 3 e 4, respectivamente [KOH 88][KOH 90]. A etapa 4 consiste no incremento de uma tabela de classificação. Esta tabela, representada pela matriz bidimensional σ_{mt} , onde t equivale ao número de classes diferentes de algarismos (10 classes), armazena, para cada neurônio, o número de padrões reconhecidos por ele. Considerando que os algarismos são apresentados sempre na mesma ordem por toda a base de dados, torna-se fácil a identificação de cada um deles para incrementar σ_{mt} . A seguir, são apresentados os passos para o incremento desta matriz.

- a. Calcular o resto da divisão i/t atribuindo o valor à r .
 - onde i é o contador dos padrões de entrada apresentados.
- b. Incrementar σ_{gr} .
 - onde g é a posição do neurônio ganhador na rede de 25 neurônios (posições de 0 a 24).

A etapa 5 monta o diagrama dos clusters, representado pelo vetor c_m , de acordo com a taxa de reconhecimento de cada neurônio. Para montar este diagrama, verifica-se em cada neurônio, através de uma leitura na matriz σ_{mt} , qual foi o algarismo mais reconhecido. Este valor é então atribuído ao vetor do diagrama,

proporcionando uma clara visualização dos clusters na rede. A seguir, apresenta-se os passos para montagem do diagrama dos clusters.

- a. Verificar o algarismo mais reconhecido em σ_{ij} .
 - onde i é um dos neurônios a serem analisados e j é uma das t categorias de algarismos da base de dados.
- b. Atribuir a c_i o valor de t .

5.3 Otimização da Rede Após o Reconhecimento

Observando os pesos de cada neurônio, após a fase de treinamento, constatou-se que muitos deles armazenaram características comuns a dois ou mais algarismos. Isto acontece principalmente nos neurônios que ficam à mesma distância entre dois neurônios ganhadores, cada um responsável pelo aprendizado de um tipo específico de algarismo. Os dois neurônios ganhadores acabam influenciando da mesma forma estes neurônios que acabam ficando na fronteira entre os dois clusters.

Outro fator prejudicial ao aprendizado é a baixa ortogonalidade de alguns algarismos [KOH 88]. Por exemplo, o algarismo 5 é parecido com o 3, que é parecido com o 8, que é parecido com o 9. Essas semelhanças podem induzir um neurônio a convergir para as características comuns a duas ou mais classes distintas, prejudicando o posterior reconhecimento.

A facilidade do aprendizado é proporcional à quantidade de neurônios que o modelo de Kohonen venha a possuir. Isto se deve ao fato de que grandes clusters, com numerosos neurônios serão formados. A partir dos testes realizados com o OCR utilizando redes neurais, foi observado que, dentro de cada cluster, que representa uma classe aprendida, poderão existir pequenos clusters representando subclasses, que podem estar individualizados pelas características do caractere. A divisão pode ir até a cada neurônio do cluster, responsável pelo armazenamento das características de um grupo restrito de padrões de entrada. A figura 5.2 exemplifica esta característica.

Considerando que o modelo de Kohonen em questão possui apenas 25 neurônios, acarretando desta forma em clusters pequenos para cada algarismo aprendido, optou-se por minimizar este problema, aumentando a definição entre os clusters através da redução destes neurônios espúrios. Com esta redução, aumenta-se a definição dos clusters, oferecendo uma fronteira bem definida para cada um deles. A rotina de otimização é composta das seguintes etapas:

1. Apresentar um padrão de entrada.
2. Calcular o erro dos neurônios.
3. Selecionar o neurônio ganhador, aquele com o menor erro.
4. Calcular a taxa de acerto dos neurônios.
5. Atualizar os pesos dos neurônios proporcionalmente.
6. Retornar ao passo 2 até o fim dos padrões de entrada.

As etapas 1, 2 e 3 são idênticas às etapas iniciais para o reconhecimento [KOH 88][KOH 90]. A etapa 4 consiste em verificar qual algarismo k , onde $0 > k \leq t$,

representa o padrão de entrada apresentado. Em seguida, verifica-se em σ_{mk} a quantidade de algarismos k acertados pelos m neurônios. O cálculo da taxa de acerto de cada neurônio i (α_i), apresentado na fórmula 5.1, nada mais é do que o percentual de acertos de cada neurônio em relação ao total de algarismos k apresentados à rede. A variável n_int representa o número de interações feitas pela rede durante o treinamento (número de padrões de entrada apresentados à rede durante a fase de treinamento).

$$\alpha_i = \frac{\sigma_{ik}}{\left(\frac{n_int}{t} \right)}$$

FÓRMULA 5.1- Cálculo da taxa de acerto

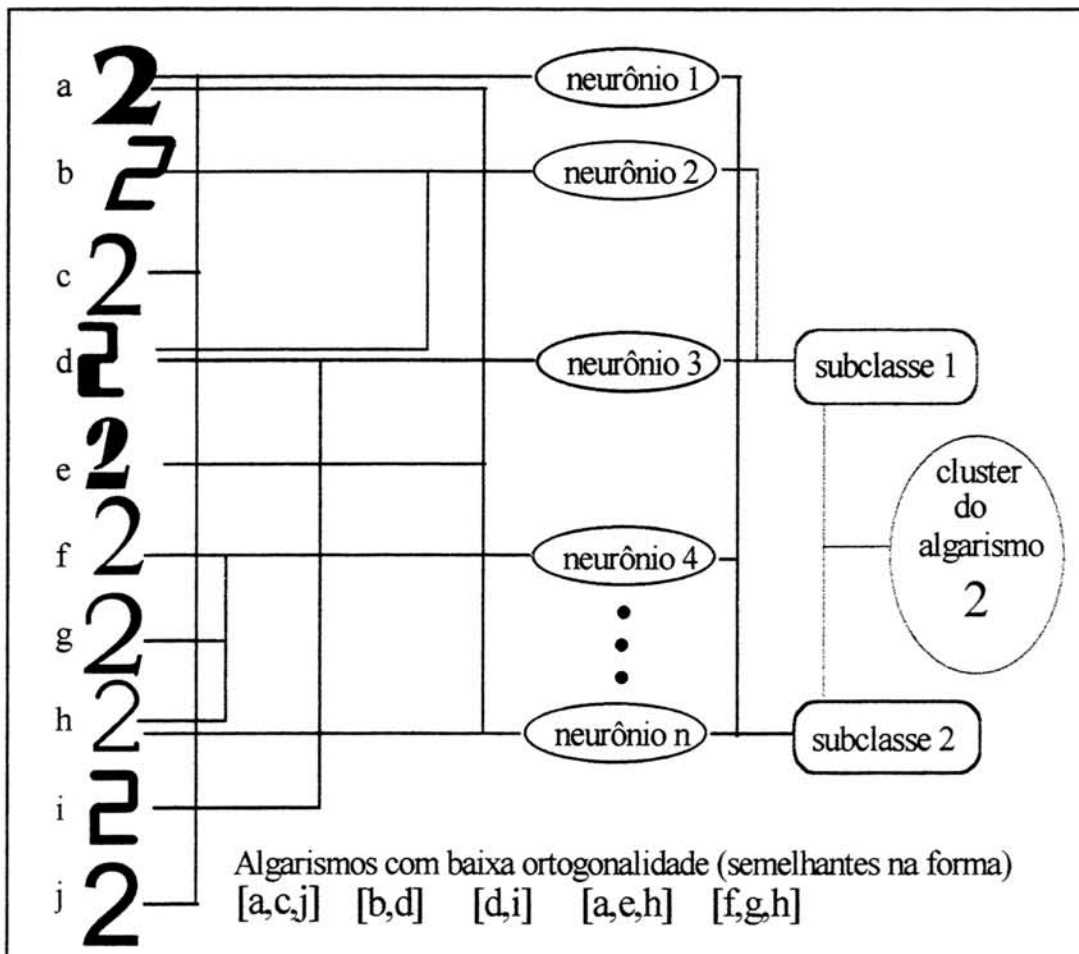


FIGURA 5.2- Divisão das classes em subclasses

A etapa 5 atualiza os pesos dos neurônios da rede em função de suas taxas de reconhecimento. A fórmula 5.2 é aplicada caso o neurônio i possua uma taxa de reconhecimento maior para o algarismo k apresentado em relação aos outros algarismos. Caso a taxa de reconhecimento do neurônio i seja maior para outro algarismo, a fórmula empregada será a 5.3, onde $0 > i \leq n$.

$$w_{ij} = w_{ij} + \alpha_i \delta_i \varepsilon (x_j - w_{ij})$$

FÓRMULA 5.2- Ajuste de pesos do neurônio que reconheceu

$$w_{ij} = w_{ij} - \alpha_i \delta_i \varepsilon (x_j - w_{ij})$$

FÓRMULA 5.3- Ajuste de pesos do neurônio que não reconheceu

Dois redes neurais foram treinadas com 2000 padrões de entrada do correio britânico, perfazendo um total de 200 padrões de cada algarismo. Os valores iniciais estipulados para d e ε foram 2.5 e 2, respectivamente. A rede *a* não possui o módulo de otimização. Para esta rede, os padrões de entrada foram apresentados duas vezes, resultando em 4000 interações de aprendizado. A rede *b* possui o módulo de otimização e, para ela, os padrões foram apresentados uma vez, resultando em 2000 interações de aprendizado. Para o módulo de otimização, os mesmos padrões foram apresentados uma vez, resultando em mais 2000 interações.

O resultado desta técnica pode ser observado ao se comparar o mapa dos pesos de cada neurônio da rede *a* (figura 5.3) com os pesos dos neurônios da rede *b*, (figura 5.4). O mapa dos pesos é obtido a partir da leitura do vetor de pesos de cada neurônio e sua disposição segue o mesmo formato da matriz dos padrões de entrada, ou seja, as dimensões 16x24, para facilitar a visualização e comparar os valores dos pesos com o algarismo aprendido. É bom salientar que o modelo neural proposto neste trabalho trabalha com um vetor de 384 posições e não com a matriz apresentada.

Os números que aparecem na matriz correspondem à primeira casa depois da vírgula do valor do peso. Por exemplo, seja o peso $w_{ij} = 0,876666453$, o valor correspondente na matriz será 8. Se o peso for menor que 0,1 o valor na matriz será 0. Caso o peso seja igual a 1, o valor na matriz será representado pelo símbolo #. Aqueles neurônios que apresentam seus vetores de pesos semelhantes aprenderam a mesma classe e fazem parte do mesmo cluster. Geralmente andam sempre juntos com o ganhador possuindo um vetor de pesos mais definido e o vizinho mais distante dele, um vetor menos definido.

Neste mapa se observa uma definição melhor de cada neurônio da rede otimizada frente às características armazenadas dos algarismos. Ao observar o mapa dos pesos da rede *a*, verifica-se que os neurônios que fazem limite entre dois clusters possuem características de duas classes distintas, atenuando a fronteira entre os clusters. Na rede *b* estes limites são mais evidentes, pois a função reforça uma característica enquanto atenua outra.

Nas tabelas de classificação das redes *a* (tabela 5.1) e *b* (tabela 5.2) observa-se o número de padrões de entrada reconhecidos pelos neurônios e constata-se claramente uma taxa de reconhecimento mais elevada feita pelos neurônios da rede otimizada. Estas tabelas são tiradas da matriz σ_{mk} , onde m equivale ao número de neurônios da rede (25) e k corresponde ao número de classes ou algarismos aprendidos (10). O valor destacado em cada coluna representa o algarismo mais reconhecido por determinado neurônio e é através deste valor que são montados os clusters. Por exemplo, segundo a tabela 5.2, o cluster que representa o algarismo 5 é formado pelos neurônios 8,13 e 18, fato este que pode ser observado nos mapas da figura 5.4.

0	1	2	3	4
4	1	2	1	1
3799772	2357776411	13788832	232212121	1 1311
589999992	34688889421	11578888831	1474423322	111 2474 1
48998889991	24898876862	1899874421	248552221	1 111237751 1
2699421 6895	147865413453	178971	1 26852111	1 11137885 1
148972 37072	369731 23541	59841	1 3741111	1 12687751 1
17981 6891	1 43842 3451	39953	1 48731 111	1 1365875
28351 894	597411 1552	16972	21158811 132	1 36774575
3884 1674	1588422222331	178711 2111	2 25872113531	1 137743486
4882 18831	15876555565621	3885335442	1 48881125831	1 379421696
4985 15371	2688777777651	589999999731	149952237961	5973 2597
13894 16961	27886444467763	6899654568862	369821136983	4884 15981
15995 17951	29974111125775	79941 1347761	4795 268632	3688412279711
13993 17941	29951 47861	6982 16763	49963224577332	58987445799542
28971 27921	49851 17862	2794311 57952	378866667787842	1489978778899732
117962 58821	39852 26962	27852111 148963	428955666778971	1276466769978852
4985 279421	37853 46841	6862 158952	112212113677532	11122224887322
27984 16794	159751 35793	27742 369742	2 221 168622 1	1 17963 1
5788446672 1	3888643563942	68863115797232	3 12211137973	1 1127973 1
7899999731 1	28999999963	158999999731 2	3 12344689732	1 1234676432
48999621 2	16899741	2688982	2 12226631	1 1125431
111	112	1	1 11	1 111
5	2	6	7	8
2	1	1	1	1
6899882	67898841	112556666541	11 1111	1 134444444431 1
389999999	17899999851	236788999863	26788888877422	134565567875412
279865468992	1168663345993	257886677752	368877766642	1234445587752 1
58941 37971	38841 15981	11367521 2343	3674213387681	12211246862 1
38963 15883	14982 3883	137731 12431	37531 111	1212135864
37851 4893	6971 3694	148621 2331	3773211211	122455552
37941 2894	4964 2883	15863 1112242	57774544531	1236776774
377621 14883	259531 27851	147743233541	1478876678741	125886679831
35654333345873	137988889881	2467767777541	578844468753	1367532578531
3486434459972	1588556678741	257887788863	3674213387681	13542 1357551
125563333346883	4795322359772	3677432346884	12441 243563	144 243563
13663111 124894	17952 24795	3763 136882	1241 126773	13431 345453
13651 138842	4992 16982	4761 4885	2332 125785	24443222345674
3663 147741	5961 4991	5741 47961	134442111126895	244433444479841
3663 1256632	697 2883	5741 37951	1254211112488831	1332333356888832
156431 128931	498251 3972	5751 257741	2331 178851	1 1111 12589731
56641123557421	28971 13896	37841 146973	2332 1258963	1 379831 1
267765556663 1	6997321157981	26875322578841	245445568852	1 11224568852 1
148999887531	69999899993	2379999999861	135678887542	1 3455565554211
11268877621	27899843	12488887721	133456633442	1344443112221
111	122	111 1	1	1
10	11	12	13	14
12	1	1	2	1
789988	117788661	2367764831 2	5676666767111	1 11111111
599999992	24999999921 1	47999999974322	115899999999835	136999999999812
48995545993	148985556899311	16898769997521	13599868789975 2	2566645598986
18973 38991	16974 137981	1247431 125653 1	135862 11121	1341 46775
15994 2893	25851 25874	13531 4442	34842 11	1121 15674
15974 4893	2563 4782	12421 1121442	3486211121	1111 25775
16992 2895	14631 34873	332224457541	247664786621	121215799873
3896 69941	134422 237852	233356888852	1258888999852	11135799999963
698431128993	236856678741	23367788863	2374213387681	123444355962
89355888994	468877899741	1222224488751	36421 36883	11 17952
17788788991	1254555668861	11 1236874	12 38971	1896
2442357972	2221211146872	24784	36962	2797
2895	2321 1 4885	11 17971	17971	1894
15981	232 147741	111 1 3794	58961	16933
28961	343 1245641	121 148711	1479821	1579711
38973	2441 12345641	121 246862	489931	1111589921
48989	152 1234462	122 1339974	12 2689921 2	111368992 1
133678861	25665434666641	2443444888751	13357998821 1	123589862
124677541	2478999776652	578999988641	26659999761	1579998864
135455421	14688765521	24588886531	289987711	27899661
1 131	222	121	112	2221
15	16	17	18	19
4	2	1	1	2
1231	11 111	1357887664531	124567786644122	236 111 112
4889999921	146899999996421	278999999998842	25788999877733	1368899998555511
279987579952	358986667899631	369965779898852	1357887878664322	12567778785542
48851 58971	268531 135882	368421 1135773	13575322222221	123223466432
25971 17992	36831 14873	24521 12575	1357431222211	1212247641
12686 6892	3653 14772	1322 13663	1246621333211 1	121137872
2684 14821	14821 34763	221 36751	12454787531 1	111358971
1476 5993	332 146752	1111134874	24555678863 1	12347985531
4761 126981	1221112457641	11112378721	113455467764311	1123566335531
675112359961	222224677521	1377421	122 1144445421	1245313552
36765449874	12236777532	25762211	134325432	2343 1353
14688688772	1256765342	12765111	233224432	2443 1353
477665673	1456532321	147521121	1233225422	2443 1452
1233 445671	12342321232	2575 11	1234235421	244313442
111221 56741	122332222322	12665 111	1335354321	244423431
11121 26662	1222232222211	137631121	1445655211	35554441
112211 367751	13323344443222	147632111	3677641112	1114678651
11224346988321	233456566543321	11136783221	23356897521	12455689987532
23465588842221	13566676544332	12223474311	145667786511	268999997644443
2444345332211	134444322121	11223442	25666653211	157887744433332
121	11 11	112	11 1	111 1111
20	21	22	23	24
3	3	3	2	1
38888884	12457888774321	2466677755341	1233366642 1	155777761
5999999954	268999999997631	248899999998842	124466799753221	4688899986
3799764589953	46887676889742	157876879999853	23567999854111	2157899996
268861 1389832	377431 1469821	158621111167871	2554566775111	22248998
15753 148962	36631 248731	3411 35873	145545677411 1	111368971
455 38851	2522 25852	1311 136831	245542357851	1 1125998
2321 48851	1111 146741	36852 1	233311257852	14897
22 26984	11 2367631	14883 2	122 1 238961	24897
248872	267642	37861	14995	15796
1 358842	2577531	3885	14985	25896
297799942	2787631	16973	159873	15796
458999721	1268663	27952	25984	16996
4798421	1358962	15983	126971	15897
4687721 1	1245854	16982	36973	25985
258752 1 21	1245642	26952	36872	14897
467411 111	12344631	46941	137873	16995
1587432221 11	13336641 1	56642	147872	1 111899721
15998888577662	13368864223221	157641	1222347885311	24888989999755
68999999999973	13468764433333	246531	122233578743222	6899999999999732
157765778988761	2246533333322	15622 1	11224367642321	1579988899888761
1 152	1 1	1111	11	1 11121
11 221	1 1	1111	131	1 11 122

FIGURA 5.3- Mapa dos pesos dos neurônios da rede a - não otimizada

0	1	2	3	4
3	2	2	2	2
11	1	21	331	231
4799872	458943	13477732	1113675	136321321
1699999992	148898852	2689887631	11135752	146432332
5998878992	2698863	2789864411	12478861	146433332
27974116995	278862	1689631	124787761	157642232
149861	178951	68842	137756761	178521343
17981	6892	38963	378636861	47841365
3895	1895	1698411	1487325871	1698215951
4884	16951	2797332321	3895216971	279612972
5882	8951	389675776631	1597716981	588316982
5984	16971	4999999999843	498415981	279612972
4984	16961	59998545589941	279731289721	4997535798321
5984	17961	6996211137983	4997546799542	169997779998743
13883	27941	169841	1499978889998753	137767788998873
2897	28841	16963	1121224897322	1223338984321
17982	58821	16972	27962	27962
4985	379521	48841	127972	27962
379821	25994	49984224799421	1124687321	1236963211
68985467981	1	49999999952	1156421	15642
799999972	1	15999999952	11	11
5899871	1	269998731	21	21
1111	1	111	11	11
12	1	1	1	1
5	6	7	8	9
3	1	1	1	1
1176997831	6788773	688876	1357777777731	1221122333211
4888999872	11799999941	18888998	479999999862	1122234678643
2888711225871	1169864457933	888852588831	1579788888651	11235576631
5885	4883	6886	158841	124477752
887	789	9891	15763	234566641
1895	3891	687	2587321	133655552
67841	12886	4884	25877556551	123555353
449841	19883	5884	2589888887741	234663446621
288884458887	39988879872	1888999882	15798656788841	2556432466221
1	69999999971	588888886	1254222237972	358634567331
48999999921	4795333468851	38887669887	11	5875222366632
1698899884	18961	888421237951	47941	36662111456653
122235884	4992	68851	38951	48854222346621
3	5961	4884	4894	48763343468951
31	6971	8892	1111	365333324567952
2	7897	78841	27982	2343112456653
3	12235884	68851	38951	48854222346621
31	5961	4884	4894	48763343468951
3	5971	8892	1111	365333324567952
2	7897	78841	27982	2343112456653
3	12235884	68851	38951	48854222346621
2	16998321258971	3888421278883	245556679864	13333336864211
4889886	6999999992	39889998894	379999987541	1225566665421
498864	28999952	7888853	145788754321	211134334532
2	111	1221	1211	111
10	11	12	13	14
3	1	2	1	1
11	11	125676431	3576677777321	1111
17899771	3328988844861	136899999721	379999999953	237777756661
14999999921	5588884999882	136899999952	13799999999821	13799999999821
38996545982	23846688798891	11468877589973	1389867878874	15566363889961
17973	46886	1344211248886	147751	57896
4795	15893	1311	14873	26895
15884	4793	12248881	138731	26995
14883	1795	11268741	148886767731	4799973
3895	1899	1136982	14899999974	469999983
178831	26994	25799952	15898653579964	2455549962
1997532469894	14885	257888852	1552	25993
389976798983	28881	123568841	26983	2796
3677666873	888	47973	25983	2696
13333347371	1886	15994	16973	2895
14899999941	1489999952	4895	38961	1893
36963	2	15887	279831	1469711
26884	5	2221	279941	1478931
378861	31	2222	2489952	113579831
122479971	654	23685544789862	12346898841	12357899731
1246779862	488889974	158999999652	157999998621	6899999652
12565431	288835	347888641	17998722	27997621
1	1	1211	231	131
111	113	111	112	111
15	16	17	18	19
2	2	2	2	2
11	1111	13677877664632	11357888788833	2367777743331
17899771	278997731	389999999952	1147999999931	1368999995551
14899999941	1489999952	489766767789963	1257987788864	1256778785542
159853358994	3898765789941	4873111135984	1258741	123223466432
29972	27992	3661	26962	1212247641
15983	3895	2411	157841	12137872
16972	27951	11	36862	111359882
16932	16971	14884	25488999952	12347988553
15995	38971	121	27971	1123566635531
1798421	1479961	159831	1331	1245313552
12899766789996	258872	48951	148951	2343
27889888993	14699931	25984	38951	2443
13454345983	2688721	1378841	16983	2443
12788641	1478753	14874	259971	244313442
15893	378632	25764	12589811	244423431
47961	158731	36752	11125799821	35554441
47873	27864211	146751	1147999821	1114678651
1258973	11221	15751	124568998621	1245689987532
13479964	4899776666521	145631	15899986411	26899997644443
137898843	4899999999862	12432	36886751	157887744433332
24676521	14666778888631	11	331	111
1	11211	11	221	111
1	111	1111	1	111
20	21	22	23	24
2	1	3	2	2
111	111	2577777564521	336666662	1122245631
2779883	5988865457873	15999999999952	14777889961	1223469972
48999999953	2888888988882	1589776777899963	11467899972	2345899841
3899653479962	288875698898883	15762111257973	112379982	1223579994
168741	169931	3541	112359871	12345678993
36732	16971	13	11124987	1244558983
1553	16961	37862	1	13342347992
34	26861	168731	148971	1222126983
3895	15887	57852	3897	126992
127982	28881	37973	49861	26983
13389921	5885	48851	139941	26982
226798922	687	27873	4897	26982
249999611	585	48952	4897	27983
368841	787	25983	4897	37881
3787611	48861	27872	149951	27983
16982	8886	37852	4896	37982
3894	8882	15751	48951	2872
5983211	58891	157741	111799721	47882
169998766553421	2883	26864	12367679998654	11222338885322
49999999999983	55	25762	377888899888872	12233379843332
35455567877741	32	14532	347777888777641	112233676433321
2	33	11	11211	11
1	1431	121	1	1211

FIGURA 5.4- Mapa dos pesos dos neurônios da rede b - otimizada

TABELA 5.1- Classificação de reconhecimento da rede a

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0	96					03					01															096
1																								34	66	100
2																04				09	06					094
3		01										11	32		55				01							098
4				39	60																			01		099
5								05	14	01			06	64	02				08							086
6		27	67					04		02																096
7																		42					13	44	01	099
8		08					73	14				05														087
9	01					04					75					19			01							098

TABELA 5.2- Classificação de reconhecimento da rede b

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0	99															01										099
1																								66	34	100
2																	65				35					100
3							03						37		59				01							096
4				60	38																				02	098
5			01						28					28					43							099
6		49	47				01			03																099
7																			57					43		100
8							97	01		02																098
9	02										45					53										098

5.4 Execução do Sistema e Avaliação do Desempenho

A rede neural de Kohonen foi submetida a uma bateria de 55 testes onde variavam-se o número de padrões de entrada para o treinamento, o número de padrões para o reconhecimento e o valor dos fatores d e ϵ , onde constatou-se que os melhores valores para estas variáveis são 2.5 e 2, respectivamente. Após, realizou-se mais 16 testes com os valores acima estipulados.

A seguir, são apresentadas 16 tabelas, com seus respectivos gráficos que mostram as classificações referentes aos treinamentos com variação na quantidade de padrões de entrada. Os valores para d e ϵ mantêm-se em 2.5 e 2. Foram utilizados 1000 padrões de entrada para o reconhecimento da rede.

As tabelas de classificação foram obtidas da matriz bidimensional σ_{mt} , onde cada linha m da tabela representa um neurônio da rede, perfazendo um total de 25 linhas. Cada coluna t representa um dos 10 algarismos (classes) a serem reconhecidos pela rede. Os gráficos foram obtidos através da proporção entre o total de padrões de entrada por algarismo (100) e total de padrões de entrada reconhecidos pelo sistema.

TABELA 5.3- Reconhecimento da rede - treinamento com 100 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0													100													100
1																				083		017				100
2											001								099							099
3									001			001		004		093				001						004
4											097											003				097
5							001			001	001						097									097
6									019	002	027		008				034					010				002
7																				001		001	098			098
8									077	002			012				009									077
9							086						014													086

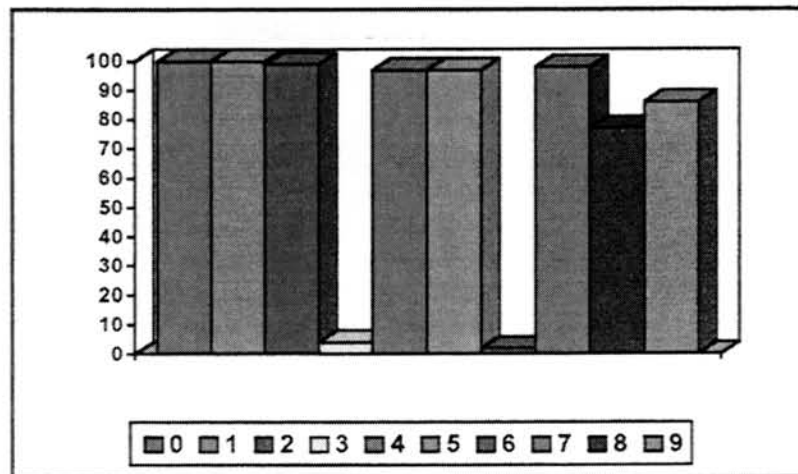


FIGURA 5.5- Gráfico do reconhecimento - treinamento com 100 padrões de entrada

TABELA 5.4- Reconhecimento da rede - treinamento com 200 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0								100																		100
1									004	095														001		095
2								003	002	009		034				014	026						012			000
3									001			004				015			042	038						053
4				033	034				033																	100
5									001		001	009				003			059	027						059
6	020	062					002		003		005	008														082
7																							084	016		100
8	001					004	003				033	059														099
9								002				001					039						058			097

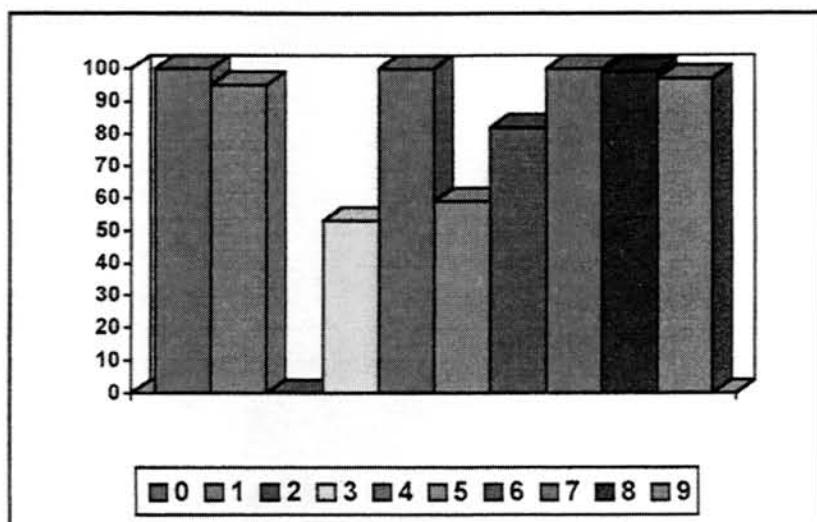


FIGURA 5.6- Gráfico do reconhecimento - treinamento com 200 padrões de entrada

TABELA 5.5- Reconhecimento da rede - treinamento com 300 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0			098															001	001							098
1					082						011														006	093
2		004			001		078				002	015														097
3		001		001	005											039	051						003			090
4																	001			084					015	099
5				002	002					001							019								076	076
6				003	006					051					039								001			090
7	096	003									001															096
8				068	028					002				002												098
9													048					014	038							100

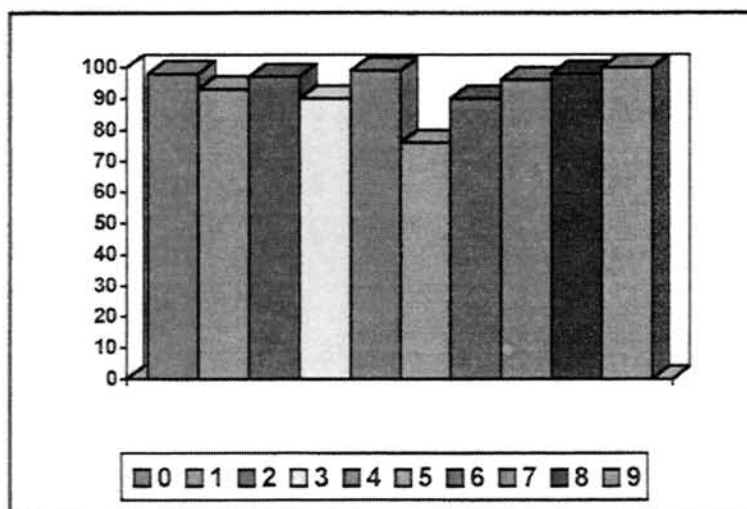


FIGURA 5.7- Gráfico do reconhecimento - treinamento com 300 padrões de entrada

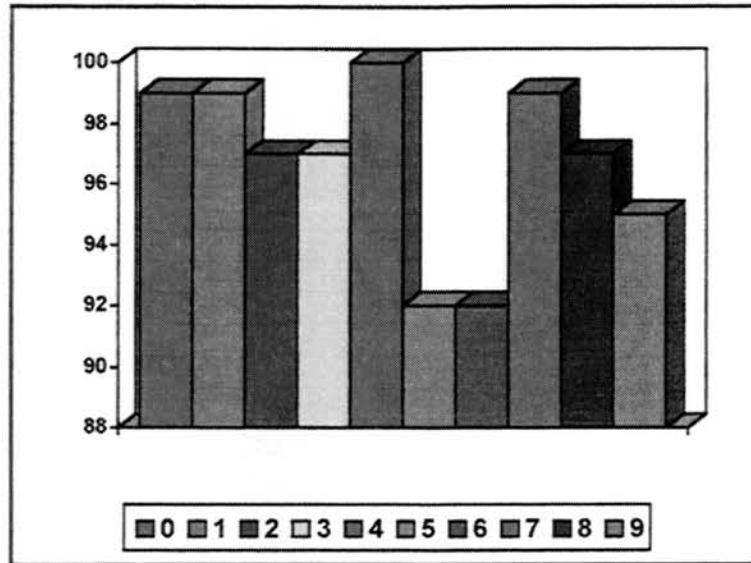


FIGURA 5.9- Gráfico do reconhecimento - treinamento com 500 padrões de entrada

TABELA 5.8- Reconhecimento da rede - treinamento com 600 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0			098								001															098
1															047	001			001	051						098
2	024				074				002																	098
3														010							009	042	039			090
4																001			064					035	099	
5											029		001		001	026	041					002				096
6				092					002					004			001							001		092
7										098							002									100
8									019					080		001										099
9		034	002				032	025	003		004															095

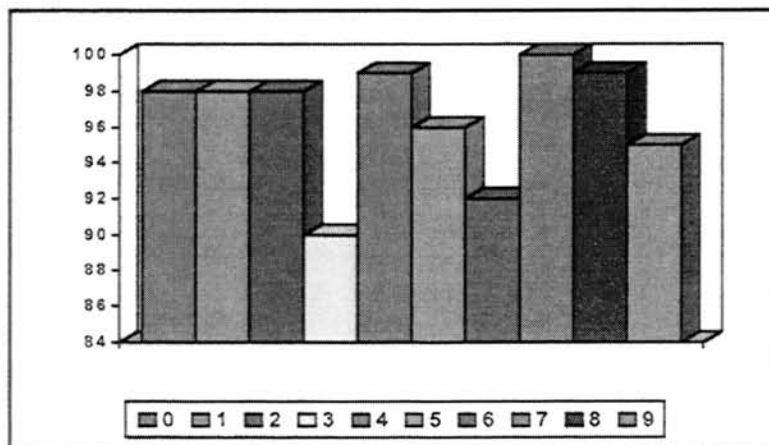


FIGURA 5.10- Gráfico do reconhecimento - treinamento com 600 padrões de entrada

TABELA 5.9- Reconhecimento da rede - treinamento com 700 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0						001													001	098						098
1		060	038			001													001							098
2											001								005	004				068	022	099
3				062				031	001	004							001	001								062
4	043					057																				100
5				045	003			040	009	001							002									052
6											058	001				039	002									097
7													066	033				001								099
8									001	036							063									099
9											002							001	001	056	040					096

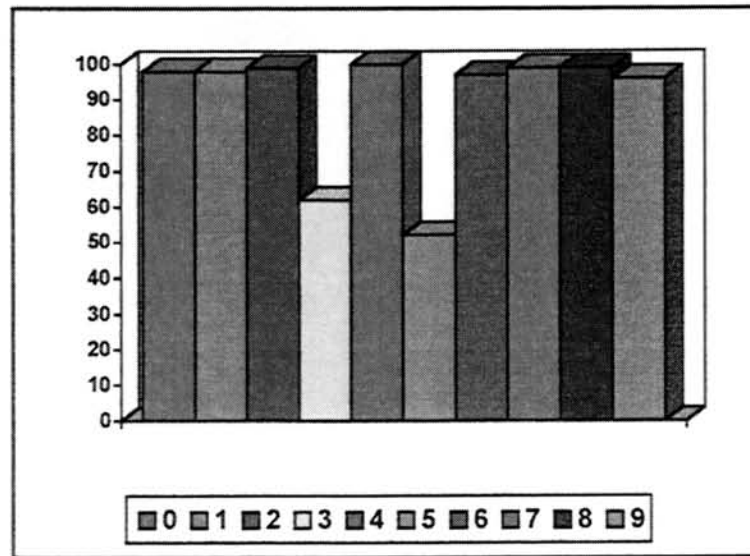


FIGURA 5.11- Gráfico do reconhecimento - treinamento com 700 padrões de entrada

TABELA 5.10- Reconhecimento da rede - treinamento com 800 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0							030				068				001						001					098
1										042	001			016	040							001				098
2	002		026					071	001																	097
3	013																		008	079						087
4															001						043	056				099
5	003															001				002				094	094	
6	008						001				042					048							001			090
7				042				058																		100
8	098													001				001								098
9	002						001					001						040					056			096

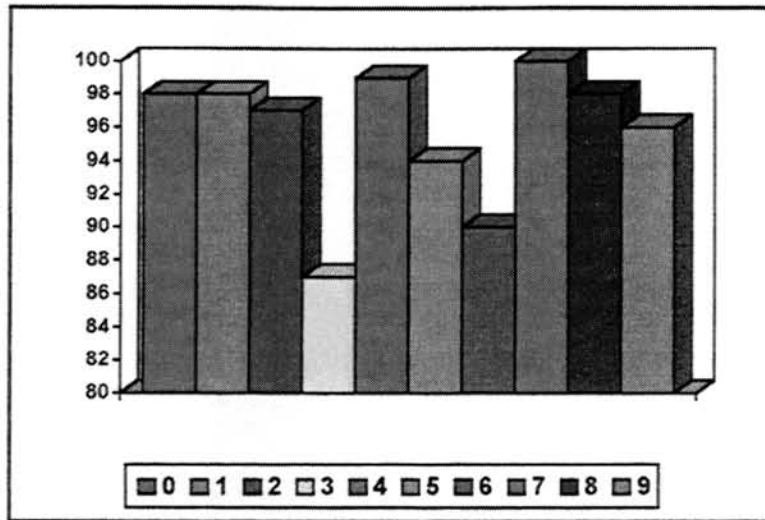


FIGURA 5.12- Gráfico do reconhecimento - treinamento com 800 padrões de entrada

TABELA 5.11- Reconhecimento da rede - treinamento com 900 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot	
0				001		063					035				001											098	
1										058			002	040													100
2							002													074					024	098	
3						004		021				029	001								045					095	
4				052	046								001	001												098	
5	002							003				002	001								001	091				091	
6	046	049					002	002														001				095	
7																			035					065		100	
8						040	058						001										001			098	
9						001	001	001			001						044	023					029			096	

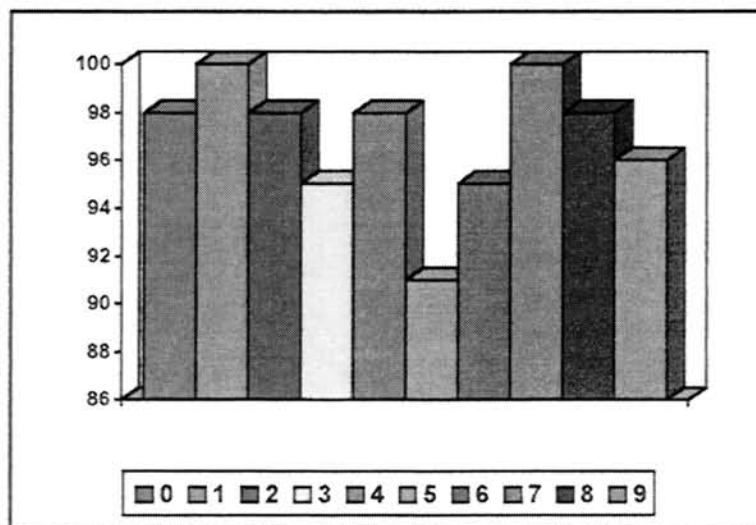


FIGURA 5.13- Gráfico do reconhecimento - treinamento com 900 padrões de entrada

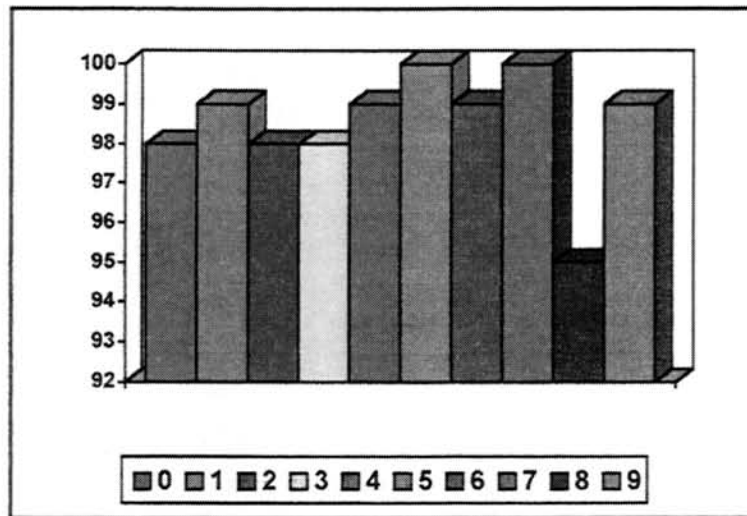


FIGURA 5.15- Gráfico do reconhecimento- treinamento com 1100 padrões de entrada

TABELA 5.14- Reconhecimento da rede - treinamento com 1200 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot	
0		001												098		001											098
1					001											002					045	052					100
2			003					001			021	074				001											095
3			009			001	042	002					012					034									099
4	038	061																				001					099
5																	001	046						053			099
6			007		091			001											001								091
7															060						040						100
8			044						053								001										097
9																				046					054		100

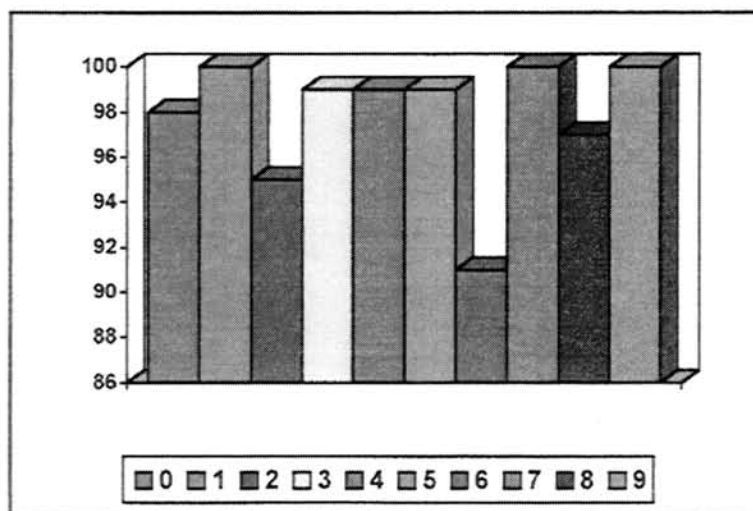


FIGURA 5.16- Gráfico do reconhecimento - treinamento com 1200 padrões de entrada

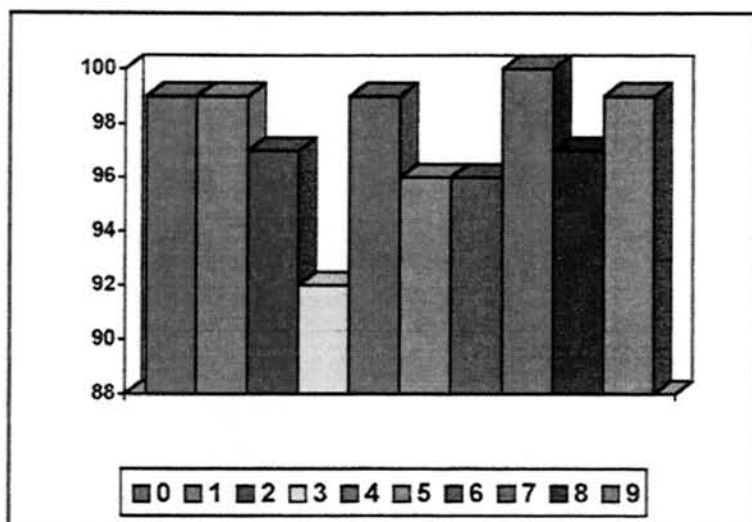


FIGURA 5.18- Gráfico do reconhecimento- treinamento com 1400 padrões de entrada

TABELA 5.17- Reconhecimento da rede - treinamento com 1500 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot	
0			002		097																	001				097	
1											001							001			035	028	035			099	
2										001		098						001								098	
3	048						043			007					002											091	
4																			037				001	062		099	
5		099								001																099	
6		001								003					001						037					058	095
7											001					039	060									099	
8			001							034			001		064											098	
9			031	053	001			015																		099	

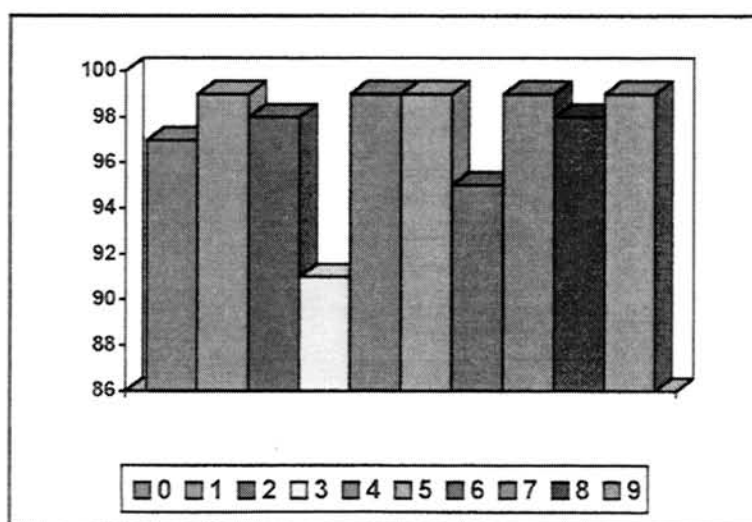


FIGURA 5.19- Gráfico do reconhecimento- treinamento com 1500 padrões de entrada

TABELA 5.18- Reconhecimento da rede - treinamento com 2000 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Tot
0	099															001										099
1																								066	034	100
2																065				035						100
3							003						037	059					001							096
4				060	038																			002		098
5			001						028				028						043							099
6		049	047				001			003																099
7																		057					043			100
8							097	001		002																098
9	002										045					053										098

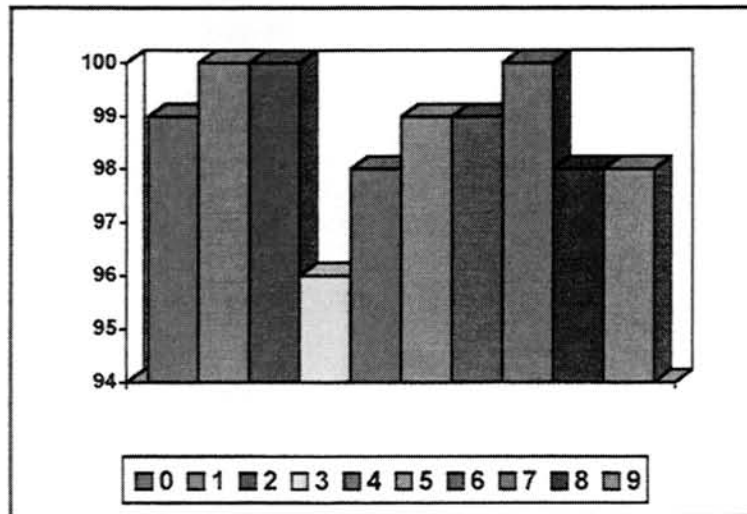


FIGURA 5.20- Gráfico do reconhecimento- treinamento com 2000 padrões de entrada

Analisando o primeiro treinamento feito com 100 padrões de entrada, observa-se, de acordo com a tabela 5.3, que a clusterização foi feita de forma muito superficial, armazenando em um único *cluster* vários algarismos distintos. A constatação parte do reconhecimento feito pelo *cluster* que reconheceu o algarismo 5.

Este *cluster*, constituído pelo neurônio 16, reconheceu erroneamente os algarismos 3 e 6, reduzindo o reconhecimento destes algarismos pelos respectivos *clusters*. Este fato se deve à modelagem dos pesos feitos neste neurônio que foi influenciado pela baixa ortogonalidade dos algarismos 3,5 6, ou seja, para este neurônio, estes 3 algarismos são parecidos o suficiente para pertencerem a mesma classe. Na figura 5.21 apresenta-se o vetor de pesos deste neurônio que, devido ao baixo número de interações para o treinamento, não pode configurar-se corretamente. Ao observar este vetor, verifica-se que os pesos mais altos formam a silueta do número 3 misturado com o número 5.

```

Neuronio[16]
1.2.....
.....1..1.....
..1.33777755331..
..3688888788421.
..368787678882..
..36542.347953..
..13431.12466441.
..13331224564432.
..23222244555442.
..11244444667542.
..12245553458852.
..2333443467862.
..1232122557764.
..2321132245864.
..12232132225784.
..12231222125874.
..12211.11235871.
12221.1.1357731.
11221111256752..
21455443688641..
..33788887764111.
..1356676443111..
.....
.....1..1.....

```

FIGURA 5.21- Vetor de pesos do neurônio 16 - treinamento com 100 padrões de entrada

O segundo teste, tabela 5.4, apresentou um número maior de *clusters* reconhecendo algarismos errados, porém a quantidade de reconhecimento erroneos foi inferior a taxa observada no primeiro teste. A média do reconhecimento elevou-se devido a um maior número de interações para o treinamento, que foi o dobro do número realizado no primeiro teste. Os algarismos 3, 6, 8 e 9 foram mais reconhecidos, sendo que seus *clusters* reconheceram erroneamente muitos algarismos 2.

Neuronio[11]	Neuronio[15]	Neuronio[16]	Neuronio[18]	Neuronio[19]
..3.....	..4.....	..4.....	..3.....	..3.....
.....2.....1..1.....
....2568877....	..333388#98615...2379#97....	...56###866842..	..1156###855432..
...4899#99982...	..138#9999999532..	..25#999999931..	1157###999#9861.	1.7#9999#8996321
..3898766699#2..	..167986698#951.	..2599966678993..	1145###88999#97..	..569999889#931.
11696421114996.	...54.1...457613	..2#98.....8992.	..2.121189997...	..3321..57#86...
..18853....28981.34573.	1499.....2996.36897....	..11...157#721..
..17971....1799.	..23.....337633	1599.....2697.	..1...58981....	..2...1578932...
..6973....16981.	1.33.....24583.	49974.....2798.4599974....237787631..
..38965...13884.	..122.111353241.	15995.....37961.3569999951..2367799953..
..269##8679981..5686772422.	..29#93....2799512368788996..4896559851.
..79#####6...224###87#542.	..3997544457995.111247991.443345783.
..577755568#95..	..122.134799973.	..2499#99799995.37#92.....2158761
1277431...147#72.	..33.....1359#84.224587575975.16##2.....13995.
26864....13993.	2232.....37#61	1....2222234883.	1.....159#1.	...11.....38951
17862....2894.	123.....1..16#52	1.....112971.	2.....138##1.	11111....4994.
267611....12795.	112.....15#62	1.....22877..	2.....1112288#8..	111111....35991.
178721....1127951	1232.....7941	1.....225771.	1..112233889#1..	11111..1245894..
..38852112236982.	24632.....149#31225874....	..11233688991...	11211113366942..
..16977233379951.	327995...177#74.244578751.	1.36668898811...	214554589943...
..289999999952..	316#9##8##9751.	..599999774....	..67999888631....	175999998842...
...348999984....	2..2568#99#62...	..479885111....	..36898711.....	..2599874311....
.....1211.....	2....143.5.31...11.....1.....	..11.....1.....
.....11.....	2.....11.1.....1.....1.....	..1.....1.....

FIGURA 5.22- Vetor de pesos dos neurônios 11, 15, 16, 18 e 19 - treinamento com 200 padrões de entrada

A figura 5.22 apresenta os vetores de pesos do neurônio 11, responsável pelo reconhecimento dos algarismos 2 e 8, do neurônio 15, que reconheceu os algarismos 2 e 3, do neurônio 16, que reconheceu os algarismos 2 e 9, dos neurônios 18 e 19, que reconheceram os algarismos 3 e 5. Os neurônios 18 e 19 formam um *cluster* responsável por uma classe híbrida constituída dos algarismos 3 e 5. Importante

observar o formato das classes misturadas no vetor de pesos, lembrando que a influência do peso é proporcional ao seu valor, ou seja, o formato das classes é evidenciado pelos valores mais altos dos vetores.

No terceiro e quarto testes, efetuado com 300 e 400 interações de entrada, respectivamente, observou-se uma definição maior dos *clusters* na rede neural, possibilitando uma sensível redução nos reconhecimentos errôneos. Destaca-se o reconhecimento feito pelos neurônios 1 e 24, no terceiro teste, tabela 5.5, e o reconhecimento efetuado pelo neurônio 16, do quarto teste, tabela 5.6. Apesar de terem reconhecidos mais de um algarismo, seus desempenhos não foram significativos dado o número de reconhecimento geral efetuado por eles.

O neurônio 17, do quinto teste, tabela 5.7, apresentou uma taxa de reconhecimento muito baixa. Pode-se dizer que este neurônio é responsável por uma classe híbrida que engloba os algarismos 3 e 8. Os demais neurônios apresentaram uma taxa menor de reconhecimentos errôneos em relação aos testes anteriores devido a um treinamento maior.

As taxas de reconhecimento observadas a partir do sexto até o último teste são progressivas e apresentam uma especialização cada vez maior de cada neurônios em um algarismo específico, gerando classes bem definidas com baixos reconhecimentos errôneos. Exceção à regra é feito pelo neurônio 3 do teste 7, tabela 5.9, neurônio 15 no décimo teste, tabela 5.12, e neurônio 16 no teste número 12, tabela 5.14.

Com a análise dos testes conclui-se que:

- A definição dos *clusters* é proporcional ao tempo de treinamento.
- Que as classes de baixa ortogonalidade tendem a aglutinar-se em uma única classe quando o número de interações para o treinamento é baixo.
- Uma classe que possui características comuns a outras classes acaba por constituir um *cluster* que reconhece os padrões de entrada da sua e das classes comuns, aumentando o índice de reconhecimentos errôneos.
- Os algarismos de alta ortogonalidade, muito diferentes dos demais, acabam gerando *clusters* bem definidos logo nas primeiras interações do treinamento da rede.
- A medida em que o treinamento da rede vai aumentando os neurônios que reconhecem mais de uma classe não fazem parte de nenhum *cluster* definido e possuem uma taxa total de reconhecimento muito baixa. Possivelmente foram influenciados com a mesma intensidade por dois *clusters* distintos. A função de otimização atenua este problema na proporção em que aumenta o número de treinamentos.
- Os padrões de entrada de uma classe que são reconhecidos pelo *cluster* de outra, enfraquecem esta classe até que a mesma seja absorvida pelas demais, ou tornem-se neurônios isolados que reconhecem uma ou mais classes com baixa ortogonalidade.

As figuras 5.23 a 5.32 apresentam os gráficos com a taxa de reconhecimento dos algarismos durante as etapas de treinamento da rede. Este valor é obtido através do somatório, dentro de cada coluna, de todos os valores em destaque na tabela de classificação.

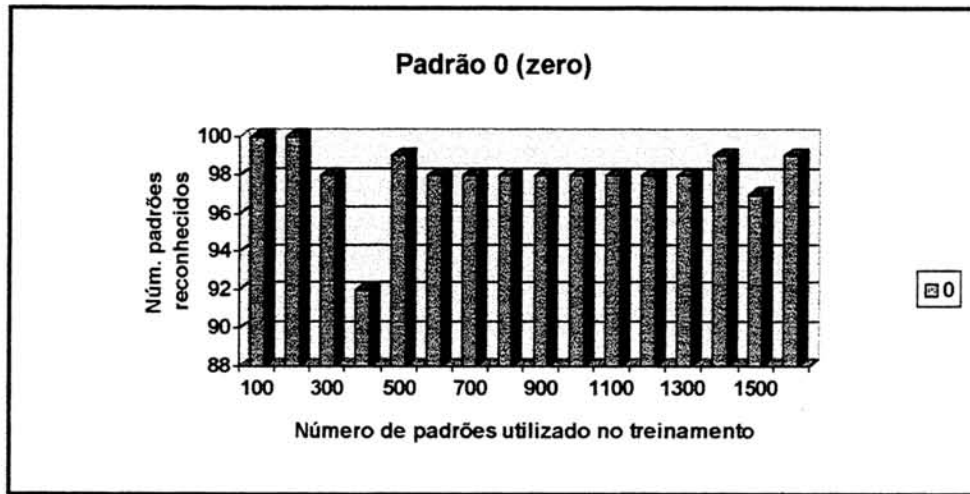


FIGURA 5.23- Gráfico com as taxas de reconhecimento do padrão 0

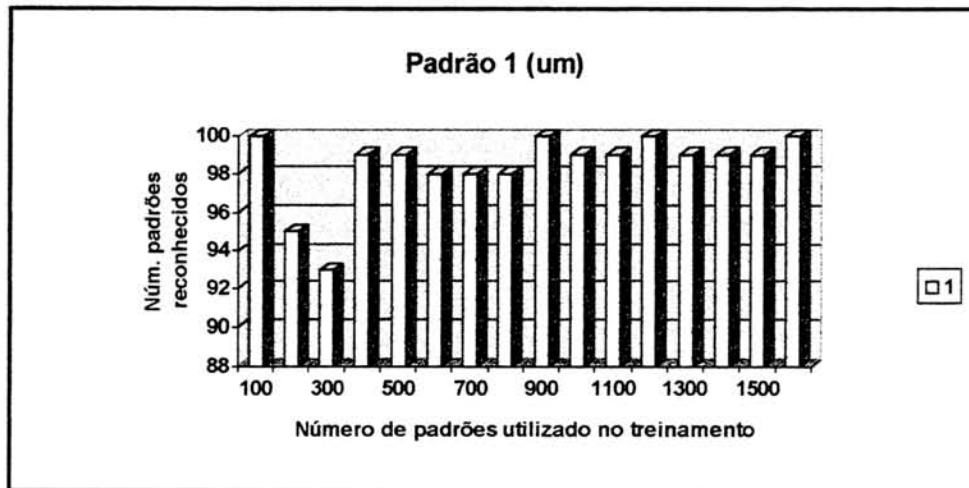


FIGURA 5.24- Gráfico com as taxas de reconhecimento do padrão 1

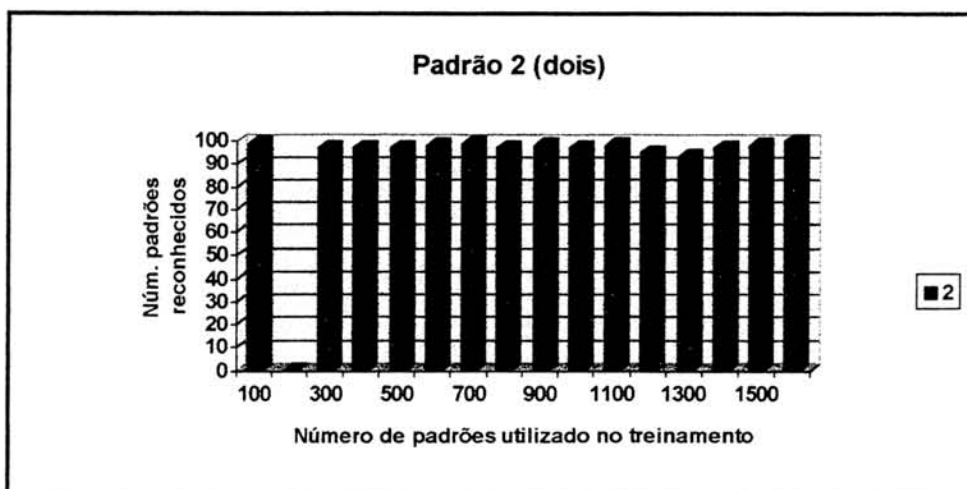


FIGURA 5.25- Gráfico com as taxas de reconhecimento do padrão 2

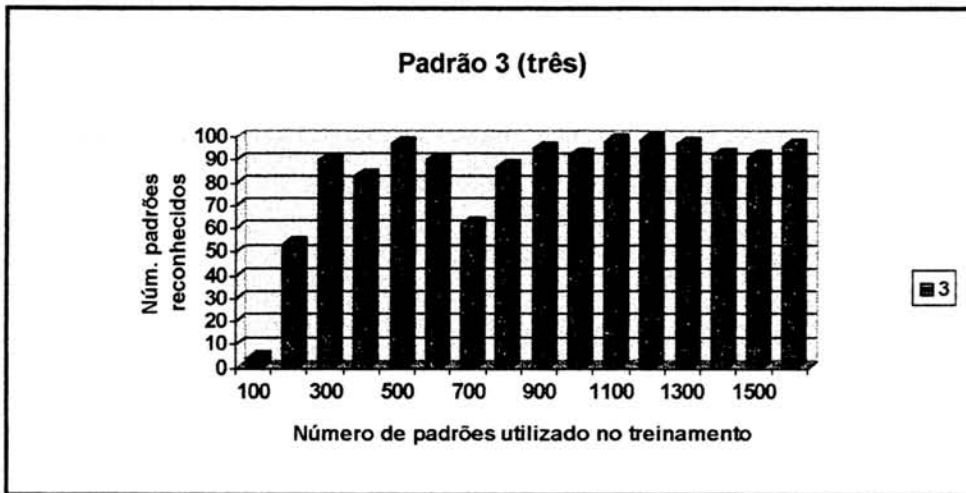


FIGURA 5.26- Gráfico com as taxas de reconhecimento do padrão 3

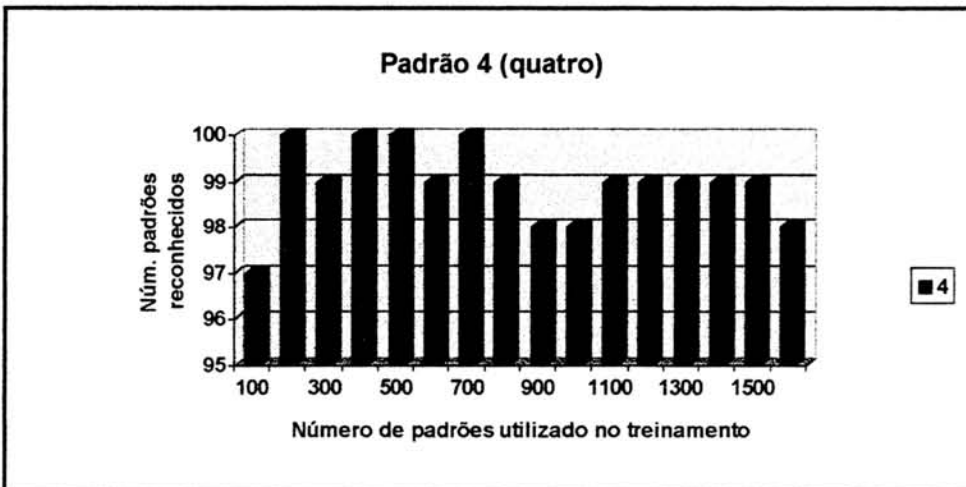


FIGURA 5.27- Gráfico com as taxas de reconhecimento do padrão 4

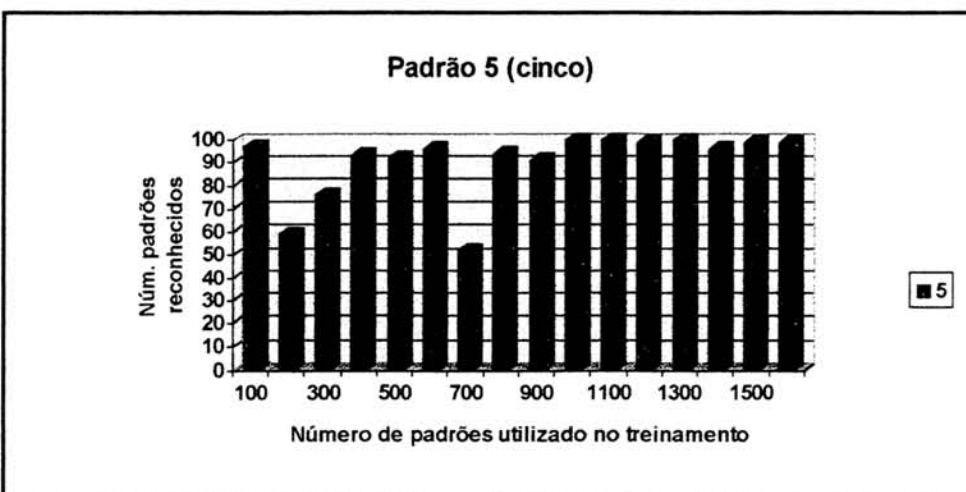


FIGURA 5.28- Gráfico com as taxas de reconhecimento do padrão 5

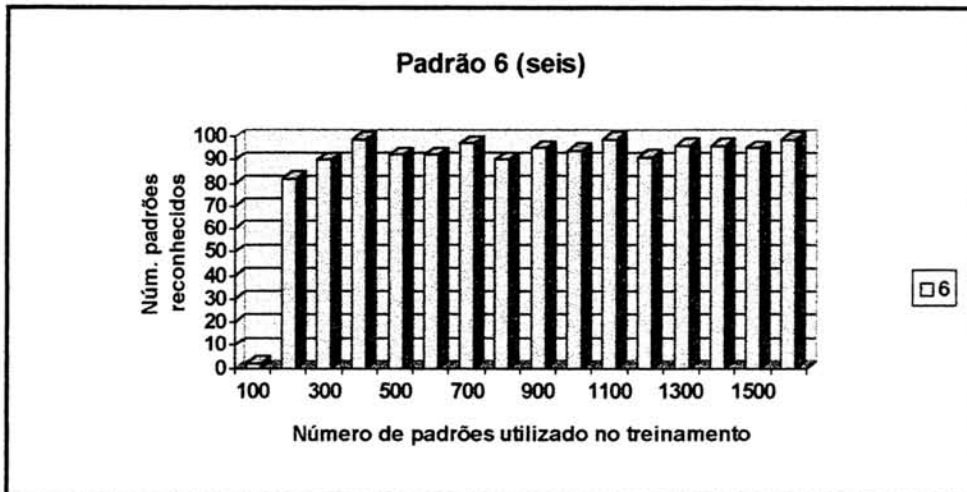


FIGURA 5.29- Gráfico com as taxas de reconhecimento do padrão 6

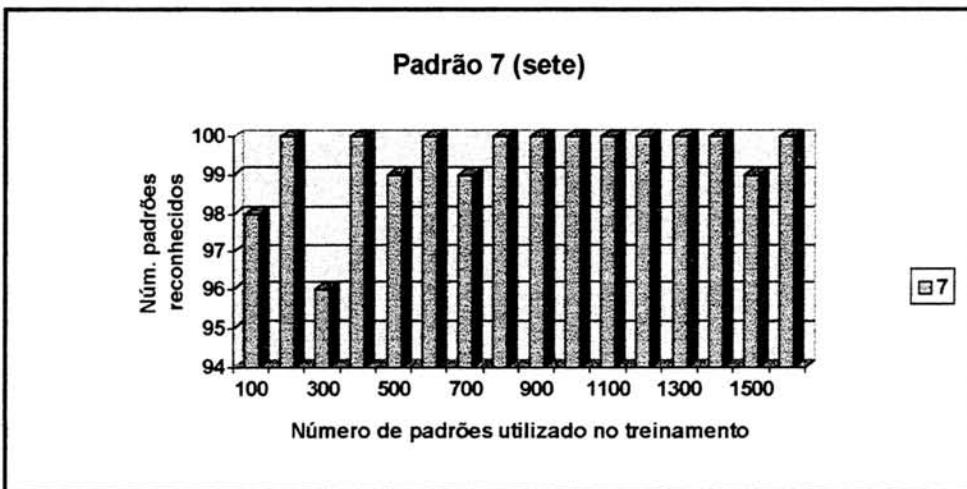


FIGURA 5.30- Gráfico com as taxas de reconhecimento do padrão 7

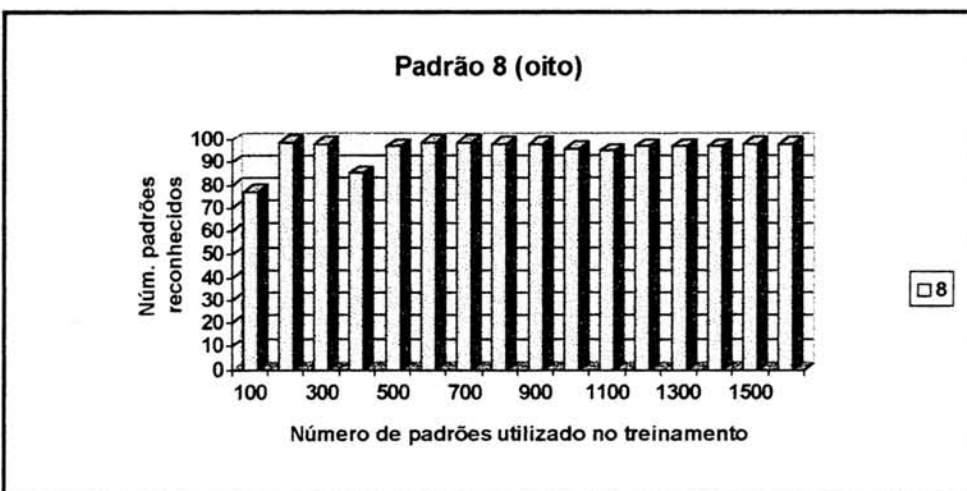


FIGURA 5.31- Gráfico com as taxas de reconhecimento do padrão 8

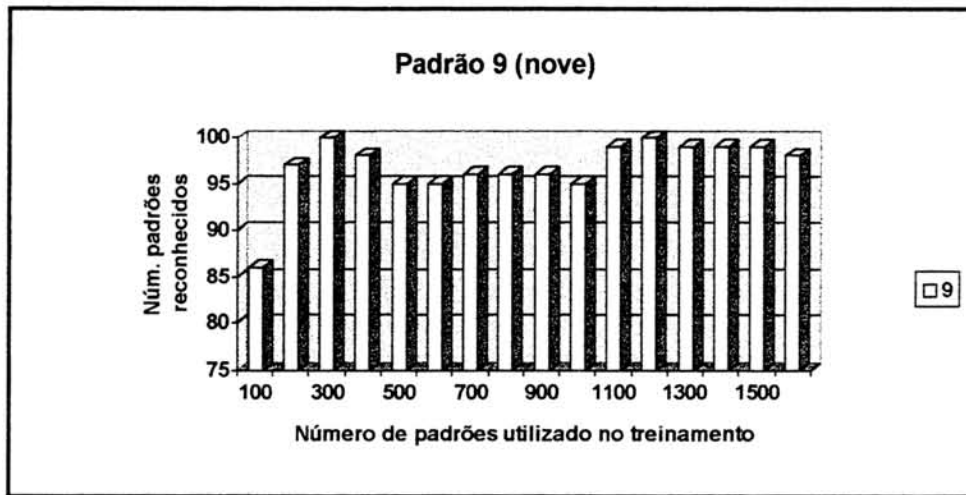


FIGURA 5.32- Gráfico com as taxas de reconhecimento do padrão 9

Avaliando o desempenho do sistema, conclui-se que a configuração ótima de treinamento é de, no mínimo, 2000 interações, para que o reconhecedor consiga identificar acima de 95% todos os algarismos, sendo que 30% destes, reconhecidos em sua totalidade.

A rede neural de Kohonen provou ser eficaz no reconhecimento ótico de caracteres. Possui a grande vantagem de aceitar qualquer padrão de entrada, estando ou não com ruído. Esta característica possibilita que se treine a rede com padrões de entrada do mundo real, sem ter a necessidade de modelá-los. Outra vantagem expressiva está na velocidade do treinamento e da otimização, conseguindo uma boa convergência já na primeira apresentação dos padrões de entrada.

Como desvantagem, existe a complexidade do algoritmo e o tamanho da rede, que podem complicar a velocidade do reconhecimento, especialmente quando se precisa de processamento em tempo real. Para o modelo de Kohonen, a precisão do sistema é inversamente proporcional à velocidade de execução e diretamente proporcional a quantidade de neurônios.

Faz-se necessário um sistema que treine com a melhor qualidade e quantidade de neurônios possível, mas que reconheça apenas com os neurônios essenciais para a atividade, reduzindo desta forma o seu tamanho e aumentando sua velocidade. Além disso, o modelo utilizado para o reconhecimento deve manter as características de plasticidade para o caso de surgir uma nova classe durante a fase de reconhecimento. O paradigma de algoritmos genéticos surge como uma opção aplicada ao modelo neural de Kohonen [RAM 95a][RAM 95b] [MCI 94] visando a otimização de seus *clusters*, para aumentar a velocidade, reduzir o tamanho e conservar a eficiência.

6 Hibridização do Modelo de Kohonen

O objetivo desta hibridização é o de reduzir o tamanho da rede neural de Kohonen utilizada no reconhecimento de caracteres [RAM 95a][RAM 95b]. A necessidade desta hibridização se justifica pelo aumento de velocidade no reconhecimento dos padrões utilizando menos neurônios e, conseqüentemente, menos recursos computacionais. Seria interessante que esta rede reduzida mantivesse as mesmas taxas de acerto que o modelo utilizado para a função de aprendizado. É neste momento que entra o processo de hibridização por algoritmos genéticos, já utilizados em trabalhos de projeto [TOD 89] e definição de pesos em redes neurais artificiais [WHI 93].

Existem alguns argumentos contra a otimização do modelo de Kohonen baseados na perda de algumas propriedades básicas desta rede como, por exemplo, a plasticidade sináptica, que possibilita a criação de novos *clusters* quando uma nova classe de padrões é apresentada. Outra propriedade importante que estaria sendo perdida é relacionada com a funcionalidade do modelo de Kohonen como sendo uma rede de auto aprendizado [KOH 88], onde o mesmo módulo é usado para treinamento e reconhecimento, pois as duas funções ocorrem simultaneamente.

Como defesa, argumenta-se a favor da otimização pela redução da rede, basicamente de duas maneiras. Primeiramente é importante que se diga que o modelo reduzido continua sendo um modelo de Kohonen e, portanto, também possui o mecanismo de plasticidade sináptica que possibilitará uma adaptação de seus neurônios a medida em que os padrões de entrada forem se modificando [KOH 90]. No caso de surgir uma nova classe de padrões, faltarão neurônios no modelo hibridizado para formar um novo *cluster*.

É bom lembrar que a grande maioria dos OCR existentes prescinde de uma modelagem das classes dos padrões a serem reconhecidos ou de um treinamento prévio para o aprendizado destas classes, não tendo nada a ver com o processo de reconhecimento, ou seja, o OCR colocado à disposição do usuário final já vem previamente configurado para reconhecer os padrões de entrada solicitados. Esta observação torna desnecessária a plasticidade sináptica numa rede que já está reconhecendo.

O segundo argumento em favor da hibridização é referente à divisão da rede em dois módulos, que são o de aprendizado e o de reconhecimento. Se a rede utilizada para o processo de reconhecimento pode ficar menor com o processo de hibridização, tornar-se tão eficiente quanto a original e ainda mais rápida, torna-se imprescindível sua aplicação, deixando a rede original apenas para o processo de treinamento.

6.1 Implementação do Modelo Hibridizado

O modelo hibridizado, feito dentro do paradigma de algoritmos genéticos, tem por objetivo transformar cada *cluster* da rede neural artificial em uma população [RAM 95a][RAM 95c]. A partir daí, utiliza-se o processo de reprodução para transportar as melhores características de cada neurônio do *cluster* para um único descendente. Este descendente, que é um neurônio, substituirá um *cluster* inteiro no reconhecimento dos padrões da classe aprendida. Desta forma, a quantidade de

neurônios da rede será a mesma do número de classes t . O algoritmo é constituído das seguintes fases:

1. Identificar o *cluster* k da rede neural, sendo $0 > k \leq t$;
2. Transformar cada *cluster* k numa população;
3. Identificar o grau de adaptação dos indivíduos ao meio;
4. Reproduzir com todos os indivíduos da população;
5. Substituir o *cluster* pelo neurônio gerado;
6. Incrementar k e retornar ao passo 1.

A etapa 1 identifica, através do vetor c_m , quais os neurônios pertencentes ao *cluster* da classe k , onde $0 > k \leq t$. Cada neurônio i pertencente a esse cluster possui $c_i = k$, onde $0 > i \leq m$.

Na etapa 2, considera-se cada neurônio em um indivíduo de uma população representada pelo *cluster* identificado, e o cromossomo deste indivíduo é o vetor de pesos do neurônio i , w_{in} .

A identificação do grau de adaptação de cada indivíduo, feita na etapa 3, é responsável pelo ajuste deste indivíduo. O ajuste é um valor numérico que informa o quanto o indivíduo está apto a sobreviver no meio ambiente (algarismos contidos no arquivo correio britânico). Sendo assim, quanto mais padrões de entrada forem reconhecidos pelo indivíduo, maior fica o seu ajuste. O ajuste é obtido através de σ_{mk} , que informa a taxa de reconhecimento de cada neurônio. Considerando que σ_{mk} é calculado na fase de reconhecimento da rede, pode-se dizer que a função de ajuste é realizada pela rotina de reconhecimento da rede neural.

A etapa 4 reproduz todos os indivíduos da população através de uma função de crossover, apresentada na fórmula 6.1, onde W'_{ij} é o novo vetor de pesos gerado pela função (cromossomo do indivíduo gerado), j é uma das entradas elementares do padrão apresentado, sendo que $0 > j \leq n$, n_int é o número de interações efetuadas no treinamento da rede neural e t é o número de classes apresentadas à rede. Neste *crossover*, os esquemáticos dos pais são transferidos a um único filho na direta proporção de seus ajustes. A figura 6.1 apresenta um exemplo da aplicação desta função sobre uma versão simplificada de um *cluster*, onde o mesmo é reduzido a apenas um neurônio que passará a reconhecer a classe inteira.

$$W'_{ij} = \sum_{j=0}^{j < n} w_{ij} \left(\frac{\sigma_{ik}}{\left(\frac{n_int}{t} \right)} \right)$$

FÓRMULA 6.1- Fórmula do crossover

Na etapa 5, é feita a substituição do *cluster* pelo novo neurônio gerado, reduzindo, segundo exemplo apresentado na figura 6.1, o tamanho deste *cluster* em 5 vezes. Ao fazer o procedimento em todos os *clusters* gerados, se terá uma rede com o número de neurônios referente ao número de classes aprendidas.

Para finalizar, a etapa 6 incrementa k e retorna à etapa 1 para buscar o próximo *cluster*.

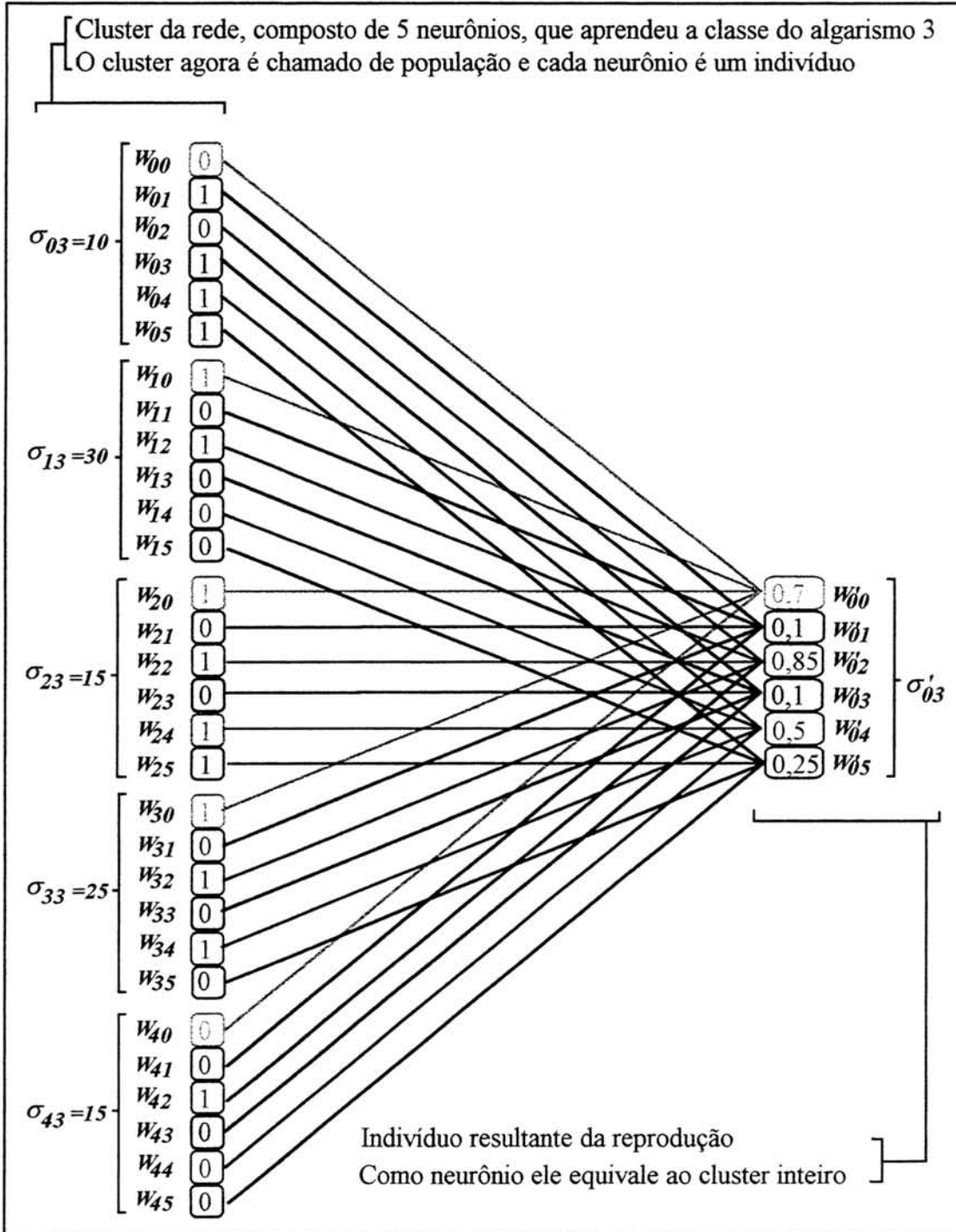


FIGURA 6.1- Uso do crossover para a redução da rede

TABELA 6.5- Reconhecimento da rede
hibridizada - treinamento com 500
padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	099				001						
1		099			001						
2			097						003		
3				088	001				011		
4					100						
5				004	001	090	005				
6	001				001		097		001		
7								100			
8				001			002		097		
9	002				001					002	095

TABELA 6.6- Reconhecimento da rede
hibridizada - treinamento com 600
padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	099				001						
1		097			001			001			001
2			099						001		
3				090					010		
4					100						
5				001		096	001		002		
6							094		006		
7								100			
8				001					099		
9	002									002	096

TABELA 6.7- Reconhecimento da rede
hibridizada - treinamento com 700
padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	099				001						
1		097			001			001			001
2			099						001		
3				062	026				012		
4					100						
5				001		093	001		005		
6						001	091		008		
7								100			
8					001				099		
9	002									002	096

TABELA 6.8- Reconhecimento da rede
hibridizada - treinamento com 800
padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	098	001			001						
1	001	097			001			001			
2			097						001	002	
3				089	001				010		
4					100						
5				003		093	001		003		
6	001					001	091		007		
7								100			
8				001			001		097	001	
9	002									002	096

TABELA 6.9- Reconhecimento da rede
hibridizada - treinamento com 900
padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	098	001			001						
1	001	099									
2			098						002		
3				085	002				013		
4					100						
5				001		096	001		002		
6	001					001	095		003		
7								100			
8						001			098	001	
9	002									002	096

TABELA 6.10- Reconhecimento da
rede hibridizada - treinamento com
1000 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	098	001			001						
1	001	099									
2			097						003		
3				083	007				010		
4				001	099						
5				002		096	001		001		
6	001					001	095		003		
7								100			
8				001			001		098		
9	002									002	096

TABELA 6.11- Reconhecimento da rede hibridizada - treinamento com 1100 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	098	001			001						
1		099									001
2			098						002		
3				088	001			011			
4					100						
5						100					
6					001	094		005			
7							100				
8			001			001		097	001		
9	001									099	

TABELA 6.12- Reconhecimento da rede hibridizada - treinamento com 1200 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	098	001			001						
1	001	099									
2			096						004		
3				088	001			011			
4					100						
5						098	001	001			
6					001	094		005			
7							100				
8			001			003		095	001		
9	001							001	098		

TABELA 6.13- Reconhecimento da rede hibridizada - treinamento com 1300 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	098	001			001						
1		099									001
2			095					001	004		
3				090					010		
4					100						
5				001		097	001	001			
6	001					001	096		002		
7								100			
8			001			002		096	001		
9										100	

TABELA 6.14- Reconhecimento da rede hibridizada - treinamento com 1400 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	099	001									
1	001	099									
2			097					001	002		
3				090					010		
4					100						
5						098	001	001			
6						001	096		003		
7								100			
8						001	001		097	001	
9	001										099

TABELA 6.15- Reconhecimento da rede hibridizada - treinamento com 1500 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	099	001									
1		099									001
2			100								
3				090					010		
4					100						
5						099		001			
6						001	096		003		
7								100			
8			001				001		097	001	
9	001										099

TABELA 6.16- Reconhecimento da rede hibridizada - treinamento com 2000 padrões de entrada

N→ K↓	00	01	02	03	04	05	06	07	08	09	10
0	099									001	
1		100									
2			099					001			
3				090	003				007		
4					099						001
5				002		095	001		002		
6				001			097		002		
7								100			
8							001		099		
9	003										097

O neurônio 10, que tem o valor de seus pesos iguais a 0, foi mantido para receber aqueles padrões de entrada que não foram reconhecidos por nenhum outro neurônio da rede. As tabelas de 6.1 a 6.16 (modelo neural hibridizado) correspondem,

respectivamente, aos treinamentos feitos nas tabelas de 5.3 a 5.18 (modelo neural não hibridizado). Estas tabelas serão comparadas na análise a seguir.

No treinamento realizado com 100 padrões de entrada, observa-se um número de reconhecimento do modelo hibridizado frente ao modelo não hibridizado superior em cinco algarismos (3, 4, 6, 8 e 9), dois reconhecimentos iguais para os algarismos 3 e 7 e três algarismos com taxa de reconhecimento menor para 0, 1 e 5. Importante observar que os algarismos 3 e 6 tiveram uma taxa de reconhecimento muito baixa para o modelo não hibridizado ao passo que no modelo hibridizado este reconhecimento foi significativamente maior, porém mais baixo do que observado em outros algarismos. Isto ocorreu porque o neurônio responsável pelo reconhecimento do algarismo 5 possui parte das características do algarismo 3. O mesmo ocorre com o neurônio responsável pelo reconhecimento do algarismo 3 em relação ao algarismo 5, como pode ser observado na tabela 6.1. O motivo desta mistura de classes, responsável pelo baixo reconhecimento dos dois algarismos está na informação contida nos clusters que aprenderam estes algarismos no modelo não hibridizado, onde o cluster do algarismo 5 reconheceu vários algarismos 3. Como as características do cluster são transferidas para um único neurônio no modelo hibridizado, é natural que os problemas, apesar de atenuados também se repitam. O algarismo 6 obteve uma taxa de reconhecimento mais elevada, apesar do neurônio 6, responsável pelo seu reconhecimento, ter reconhecido também alguns algarismo 5. Mesmo com estes problemas, constata-se que a média do reconhecimento foi maior no modelo hibridizado em relação ao modelo não hibridizado.

No reconhecimento com 200 padrões de entrada, apresentado na tabela 6.2 em comparação com a tabela 5.4, observa-se um reconhecimento maior para os algarismos 1, 2, 5 e 6, a mesma taxa de reconhecimento para os algarismos 0,4 e 7, e um reconhecimento menor para os algarismos 3,8 e 9. Novamente os algarismo 3 e 5 apresentaram confusão no reconhecimento feito pelos respectivos neurônios acarretado pela herança transmitida dos clusters vindos do modelo não hibridizado que apresentam o mesmo problema, como pode ser verificado na tabela 5.4. No modelo não hibridizado, o algarismo 2 foi reconhecido erroneamente pelos clusters que aprenderam os algarismos 5,8 e 9. No modelo hibridizado, apesar de ter ocorrido o mesmo problema, o neurônio 2 conseguiu uma taxa significativamente maior de reconhecimento para este algarismo.

Assim como no modelo não hibridizado, observa-se que, proporcionalmente ao número de padrões apresentados para o treinamento, os neurônios vão se especializando, aumentando a taxa de reconhecimento. O importante é que o modelo hibridizado proporciona um aumento da taxa de reconhecimento com um número menor de padrões de entrada apresentados para o treinamento, em comparação com o modelo não hibridizado, conseguindo uma taxa de reconhecimento mais homogênea logo nas primeiras interações.

O treinamento realizado com 300 algarismos apresentou um desempenho muito superior para o modelo hibridizado, onde seis algarismos foram reconhecidos com uma taxa maior (0, 1, 2, 5, 6 e 7), destacando a elevação na taxa de reconhecimento do algarismo 5. O único algarismo que apresentou uma taxa de reconhecimento menor foi o 3 e a perda foi de apenas dois padrões de entrada.

A partir do treinamento com 400 Algarismos até o treinamento com 2000 Algarismos, o modelo hibridizado vai mantendo seu desempenho enquanto o modelo não hibridizado continua elevando sua taxa de reconhecimento, vindo a ultrapassar o desempenho somente com 2000 Algarismos apresentados para o treinamento. Nota-se que a comparação para desempenho refere-se apenas aos Algarismos que apresentaram **maior taxa de reconhecimento** no modelo hibridizado. Se forem considerados os caracteres que apresentaram **de igual para maior taxa de reconhecimento** nas classes apresentadas, constata-se que o modelo hibridizado supera em todos os testes, vindo a empatar somente no último, onde ocorre dois Algarismos mais reconhecidos, três Algarismos reconhecidos na mesma quantidade ($3+2=5$) e cinco Algarismos menos reconhecidos. A média destes reconhecimentos pode ser verificada na tabela 6.17, onde é feita a comparação do modelo hibridizado em relação ao modelo não hibridizado.

TABELA 6.17- Taxas de reconhecimento para cada treinamento do modelo hibridizado em comparação ao modelo não hibridizado

	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	2000
Número de padrões de entrada para treinamento	5	4	6	6	2	5	3	3	2	4	2	3	3	2	5	2
Número de classes mais reconhecidas	2	3	3	2	6	4	5	3	6	4	6	2	4	7	3	3
Número de classes reconhecidas na mesma quantidade	3	3	1	2	2	1	2	4	2	2	2	5	3	1	2	5
Número de classes menos reconhecidas																

As figuras 6.3 a 6.12 apresentam os gráficos com a taxa de reconhecimento dos Algarismos durante as etapas de treinamento da rede. Este valor é obtido através do somatório, dentro de cada coluna, de todos os valores em destaque na tabela de classificação.

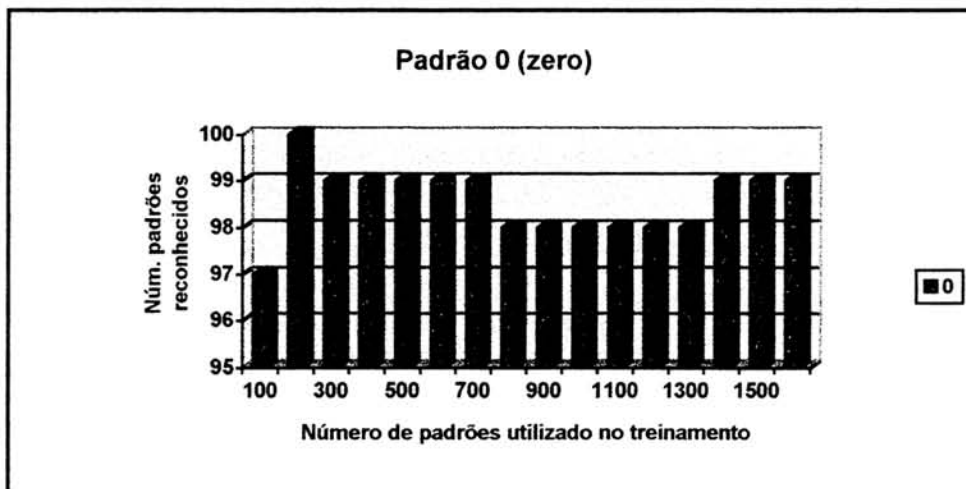


FIGURA 6.2- Gráfico com as taxas de reconhecimento do padrão 0 - rede hibridizada

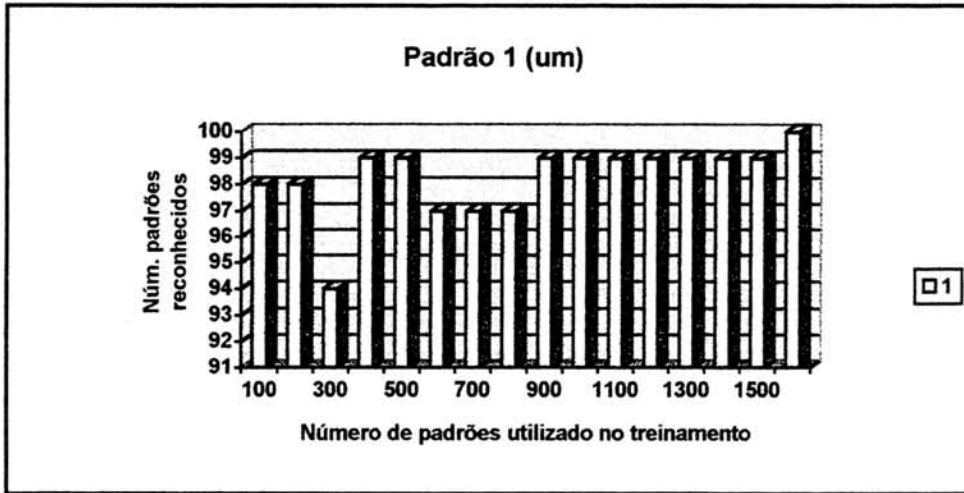


FIGURA 6.3- Gráfico com as taxas de reconhecimento do padrao 1 - rede hibridizada

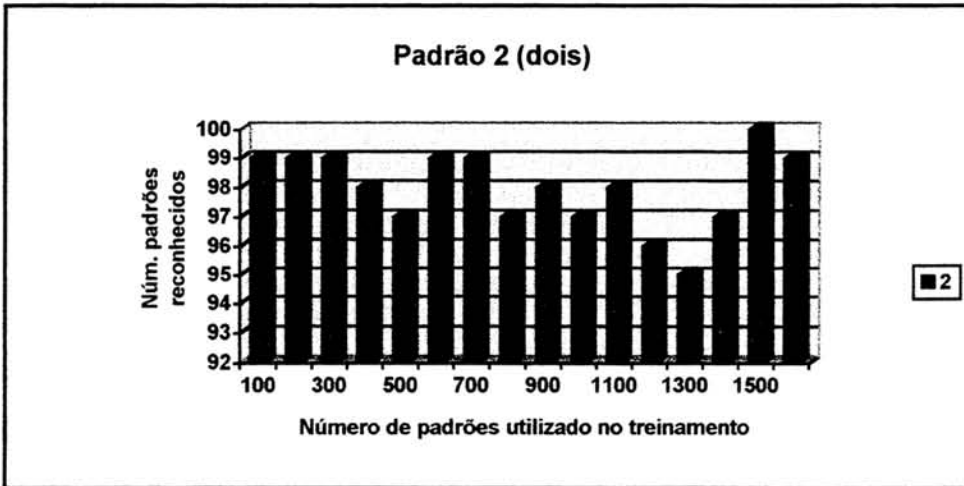


FIGURA 6.4- Gráfico com as taxas de reconhecimento do padrao 2 - rede hibridizada

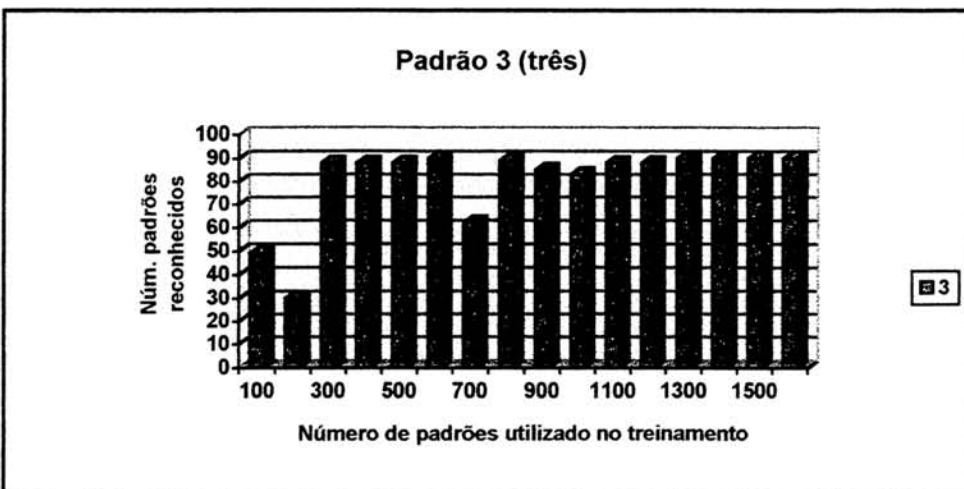


FIGURA 6.5- Gráfico com as taxas de reconhecimento do padrao 3 - rede hibridizada

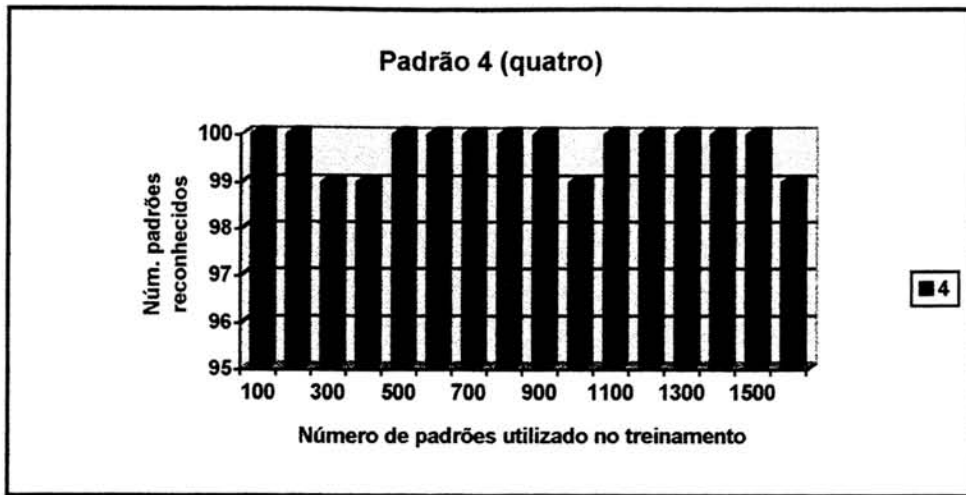


FIGURA 6.6- Gráfico com as taxas de reconhecimento do padrao 4 - rede hibridizada

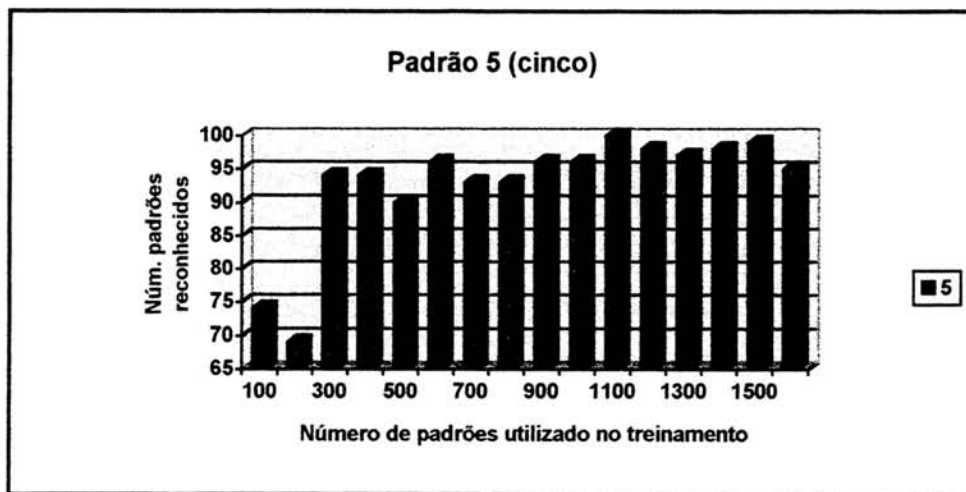


FIGURA 6.7- Gráfico com as taxas de reconhecimento do padrao 5 - rede hibridizada

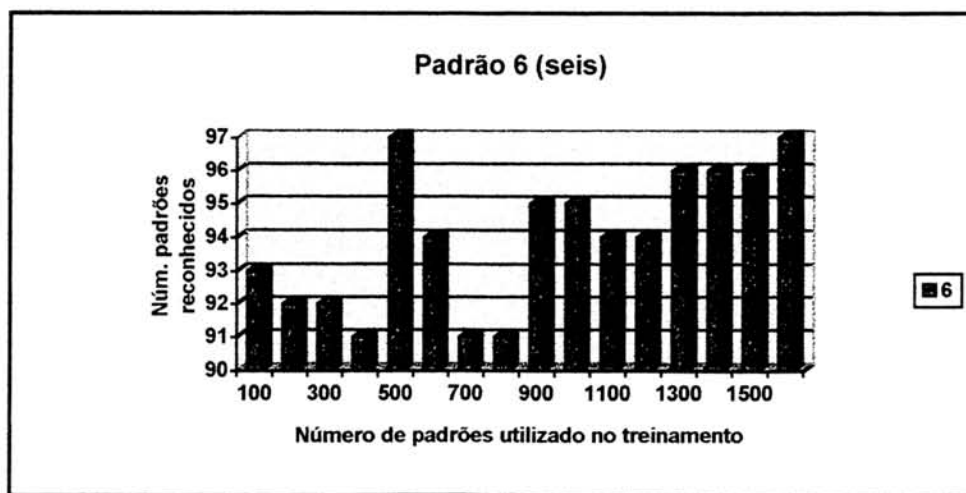


FIGURA 6.8- Gráfico com as taxas de reconhecimento do padrao 6 - rede hibridizada

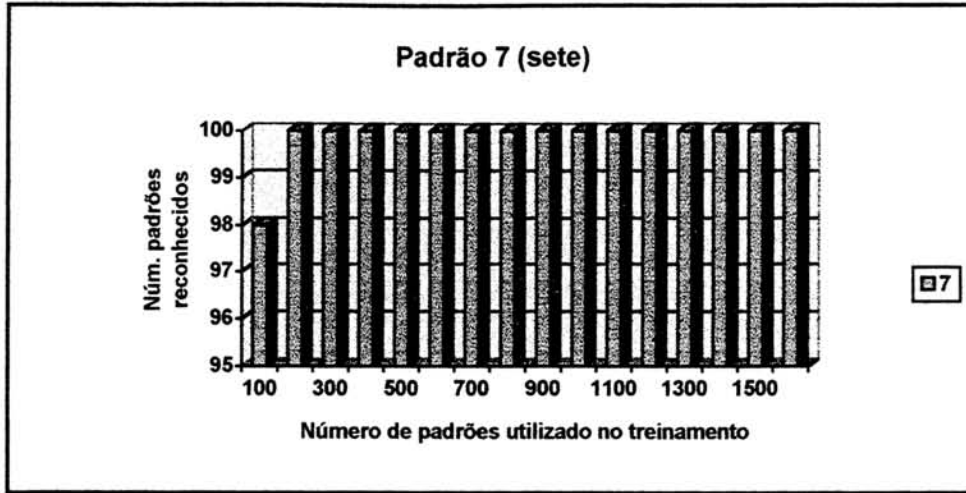


FIGURA 6.9- Gráfico com as taxas de reconhecimento do padrao 7 - rede hibridizada

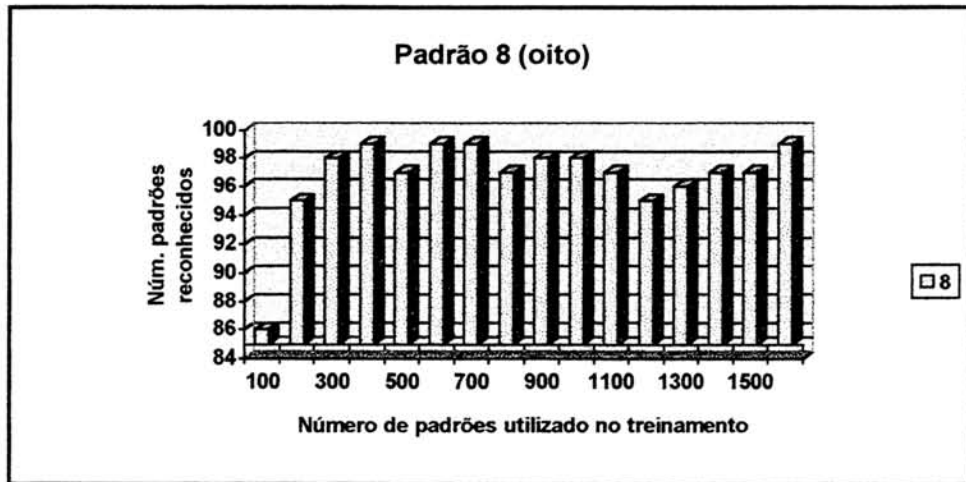


FIGURA 6.10- Gráfico com as taxas de reconhecimento do padrao 8 - rede hibridizada

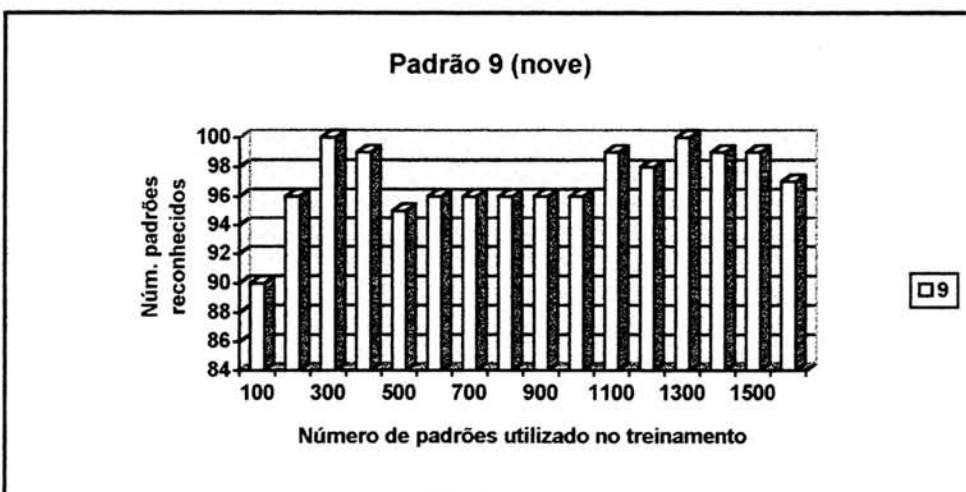


FIGURA 6.11- Gráfico com as taxas de reconhecimento do padrao 9 - rede hibridizada

Comparando as tabelas com as taxas de reconhecimento dos modelos nas etapas antes e depois da hibridização, conclui-se que.

- Nos dois modelos a taxa de reconhecimentos errôneos decresce com o número de padrões de entrada apresentados na fase de treinamento.
- Ao contrário do que acontece com o modelo não hibridizado, no modelo hibridizado não existe muita diferença nas taxas de reconhecimento entre as classes.
- A taxa de reconhecimento no modelo hibridizado é maior com treinamentos contendo poucos padrões de entrada. Isto ocorre devido à função de crossover que transfere os esquemáticos da classe aprendida numa proporção maior do que a transferência dos esquemáticos de outras classes.

O sistema foi escrito em Turbo C++ for Windows, versão 4.0, e rodou num 486 DX/2 66Mhz com acelerador gráfico da Trident, de 80Mhz, dentro do ambiente gráfico Windows 3.1, da Microsoft. O tempo gasto para o treinamento de 100 padrões de entrada foi de **2,78 segundos**. Para o reconhecimento de 100 padrões de entrada foram gastos mais **5,94 segundos**. A rotina de otimização levou **8,41 segundos** para completar seu trabalho e o reconhecimento da rede otimizada levou **5,87 segundos**.

Depois do processo de hibridização, onde a rede reduziu mais de 50% o seu tamanho, o tempo gasto para o reconhecimento dos 100 padrões de entrada caiu para **2,47 segundos**. O tempo gasto para o reconhecimento do modelo hibridizado foi **2,38 vezes menor** do que o tempo gasto pelo modelo normal. Em termos de memória, o modelo hibridizado ocupa **60% menos** no tamanho das matrizes, devido ao seu tamanho reduzido.

7 Conclusão

O objetivo deste trabalho foi o desenvolvimento de um sistema híbrido para otimizar o funcionamento de um OCR baseado na tecnologia de redes neurais artificiais, mais precisamente o modelo neural de Teuvo Kohonen.

A rede neural de Kohonen empregada no OCR é uma variação do SOM. A diferença encontra-se no desenvolvimento de dois módulos adicionais que visam, respectivamente, a classificação dos *clusters* para identificar os padrões aprendidos pela rede e a otimização destes *clusters* pelo reforço dos neurônios utilizando os padrões de entrada mais reconhecidos por eles. Com isso, consegue-se neurônios com o vetor de pesos mais configurados para um padrão específico, eliminando os estados espúrios onde um neurônio aprende partes de várias classes, não reconhecendo bem nenhuma delas.

Os caracteres reconhecidos pelo OCR correspondem a um conjunto de algarismos de 0 a 9 pertencentes ao arquivo de testes do correio britânico. Estes algarismos foram digitalizados pelo correio britânico e armazenados em um arquivo ASCII sob a forma de matrizes binárias de tamanho 16x24 numa sequência de 0 a 9 repetindo 300 vezes, perfazendo um total de 3000 algarismos digitalizados.

Avaliando o desempenho do sistema, conclui-se que a configuração ótima de treinamento é de, no mínimo, 2000 interações, para que o reconhecedor consiga identificar acima de 95% todos os algarismos, sendo que 30% destes, reconhecidos em sua totalidade. O módulo de otimização fortalece o peso dos neurônios de um *cluster* na proporção em que este aprendeu sobre o algarismo. Desta forma, reduz-se o reconhecimento errôneo e facilita a distinção entre algarismos de baixa ortogonalidade, como se pode verificar na figura 2.2.

Pela comparação das tabelas verifica-se que existem alguns algarismos que, mesmo com um baixo treinamento, formaram seus *clusters* nos neurônios da rede. São eles o 0, 2, 8 e o 9. De acordo com a figura 5.3, os mapas dos pesos dos neurônios denunciam uma semelhança entre os 4 algarismos anteriormente citados. Isto significa que os fatores responsáveis pela clusterização de um daqueles algarismos podem ter influenciado a clusterização dos demais.

A rede neural de Kohonen provou ser eficaz no reconhecimento ótico de caracteres, conseguindo uma média de 98,5 acertos para os padrões apresentados. Possui a grande vantagem de aceitar qualquer padrão de entrada, estando ou não com ruído. Esta característica possibilita que se treine a rede com padrões de entrada do mundo real, sem ter a necessidade de modelá-los. Outra vantagem expressiva está na velocidade do treinamento e da otimização, conseguindo uma boa convergência já na primeira apresentação dos padrões de entrada.

Como desvantagem, existe a complexidade do algoritmo e o tamanho da rede, que podem reduzir a velocidade do reconhecimento, especialmente quando se precisa de processamento em tempo real. Para o modelo de Kohonen, a precisão do sistema é inversamente proporcional à velocidade de execução e diretamente proporcional a quantidade de neurônios.

Objetivando a otimização do sistema de OCR desenvolvido com o modelo de rede neural de Kohonen, foi empregado o paradigma de Algoritmos Genéticos, mais

precisamente uma versão dos módulos de seleção natural e reprodução, para reduzir o tamanho da rede neural artificial.

Comparando o resultado dos 2 modelos, percebe-se que o modelo hibridizado pouco deriva o seu resultado daquele que foi obtido com o modelo neural original, obtendo **64** resultados de reconhecimento de classe iguais. Existem pequenas flutuações que podem denunciar uma perda de desempenho no reconhecimento para algumas classes em alguns testes específicos onde, em **40** reconhecimentos, o modelo hibridizado perdeu para o modelo neural original. Porém, o modelo hibridizado superou em **56** reconhecimentos o modelo original, destacando-se o total de números 2 reconhecidos pelas redes no segundo teste, e o total de números "6" reconhecidos no primeiro teste.

Outro fator importante a favor do modelo hibridizado está na média dos padrões de entrada reconhecidos. Pelas observações do gráfico, fica evidente que a diferença na taxa de reconhecimento dos padrões de entrada mantém-se no mínimo à **30%** para todas as classes e em todos os testes. Não existem flutuações abruptas como as encontradas no modelo neural não hibridizado para os Algarismos **9,7,5,4** e **0**, como demonstrado nos gráficos da figura 5.3. Também não se observou no modelo hibridizado, nenhum reconhecimento a nível tão baixo como em alguns testes efetuados com o modelo não hibridizado, a exemplo dos Algarismos **2**, no segundo teste, o **3**, no primeiro teste e o **6**, também no primeiro teste.

Conclui-se, através dos testes efetuados, que o modelo hibridizado, ao eliminar os neurônios espúrios da rede, induzindo a cada neurônio o armazenamento de uma classe completa, é mais estável que o modelo neural não hibridizado necessitando de menos interações de treinamento para obter o mesmo resultado. Com isso economiza-se tempo na mais demorada fase do sistema de reconhecimento, que é a modelagem das informações utilizadas.

Como resultado final deste processo de hibridização, observou-se um aumento na ordem de **60%** na velocidade de execução, reduzindo, também em **60%**, a quantidade de memória utilizada para o processo de reconhecimento, conseguindo o desempenho médio de **97,5** acertos para os padrões apresentados. Com estes resultados, atesta-se a viabilidade do emprego de sistemas híbridos para execução de tarefas de forma mais otimizada do que se fossem executadas por paradigmas tradicionais.

As perspectivas futuras relacionadas com esta tecnologia visam a otimização de redes neurais artificiais do tipo auto-aprendizado, onde se faz presente a definição clara dos *clusters* representando as classes dos diversos padrões de entrada apresentados. Como exemplo prático deste novo trabalho, destaca-se um reconhecedor de sinal-radar, a ser desenvolvido para o Instituto de Pesquisas da Marinha, utilizando novamente a rede neural de Kohonen, hibridizada com algoritmos genéticos. O objetivo é a redução da rede a ponto de poder rodar em tempo real na identificação de alvos captados pelo radar.

Bibliografia

- [ARB 94] ARBIB, M.A.; ROBINSON, J. A. **Natural and Artificial Parallel Computation**. Cambridge, Massachussets: The MIT Press, 1994. 345p.
- [AUS 90] AUSTIN, Scott. An Introduction to Genetic Algorithms. **AI Expert**, San Francisco, v. 5, n. 3, p. 48-53, Mar. 1990.
- [BRI 92] BRILL, Frank; BROWN, Donald; MARTIN, Worthy. Fast Genetic Selection of Features for Neural Network Classifiers. **IEEE Transactions on Neural Networks**, New York, v. 3, n. 2, p. 324-328, Mar. 1992.
- [CHE 94] CHEN, C. Y.; HWANG, C. J. **A Multi-level Backpropagation Network for Pattern Recognition Systems**. Chung-Li, Taiwan: Department of Computer Engineering and Science, Yuan-Ze Institute of Technology, R.O.C., 1994.
- [CAV 93] CAVALIERI, S. **Hopfield Neural Network for Routing**. Italy: Istituto di Informatica e Telecomunicazioni, Facolta di Ingegneria, Università di Catania, 1993.
- [DAV 91] DAVIS, Lawrence. **Handbook of Genetic Algorithms**. New York: Van Nostrand Reinhold, 1991. 385p.
- [DOY 60] DOYLE, W. Recognition of Sloppy Hand-Printed Characters. In: WESTERN JOINT COMPUTER CONFERENCE, 1960. **Proceedings...** [S.l.:s.n.], 1960. p. 133-142.
- [ERA 57] ERA. An Electronic Reading Automation. **Eletronic Engineering**, [S.l.], p.189-190, Apr. 1957.
- [FER 94] FERREIRA, Ana Paula Ludke. **Reconhecimento de Caracteres**. Porto Alegre: CPGCC - UFRGS, 1994.
- [FOG 94] FOGEL, David B. An Introduction to Simulated Evolutionary Optimization. **IEEE Transactions on Neural Networks**, New York, v. 5, n.1, p. 1-11, Jan. 1994.
- [FRE 92] FREEMAN, James A.; SKAPURA David M. **Neural Networks: Algorithms Applications and Programming Techniques**. Reading, PA: Addison-Wesley Publishing Company, 1992. p. 401.
- [GLA 56] GLAUBERMAN, M. H. Character Recognition for Busines Machines. **Eletronic**, New York, v. 29, n. 2, p.132-136, Feb. 1956.
- [GRO 87] GROSSBERG, Stephen; LEVINE, Daniel. **Neural Dynamics of Attentionally Modulated Pavlovian Conditioning: Blocking, Inter-Stimulus Interval, and Secondary Reinforcement**. [S.l.]: Applied Optics, The Optical Society of America, 1987. p. 5015-5030.
- [GRO 93] GROTHOR, Patrick J; CANDELA, Gerald T. **Comparison of Handprinted Digit Classifiers**. [S.l.]; Image Recognition Group - National Institute of Standards and Technology, 1993.

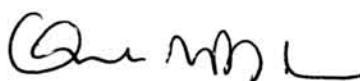
- [GUA 92] GUAZELLI, Alex. **Fundamentação de Modelos de Redes Neurais e seus Métodos de Aplicação no Reconhecimento**. Porto Alegre: CPGCC-UFRGS, 1992.
- [GUA 93] GUAZELLI, Alex; BARONE, Dante. A Simplified Art Map Architecture For Real-Time Learning. In: MIRA, J.; CABESTANI, J.; PRIETO, A. (Eds.) **New Trends in Neural Computation**. Berlin: Springer-Verlag, 1993. (Lecture Notes in Computer Science, 686).
- [GUY 84] GUYTON, Artur C. **Textbook of Medical Physiology**. [S.l.:s.n.], 1984.
- [HAN 62] HANNAN, W.J. R.C.A. Multifont Reading Machine. In: FICHER, G.L. et al. **A Optical Characterer Recognition**. [S.l.]: Ed. Mac-Gregor & Werner, 1962. p 3-14.
- [HOL 75] HOLLAND, J. **Adaptation in Natural and Artificial Systems**. Ann Arbor:Univ. of Michigan Press, 1975.
- [IJI 63] IJIMA, T.; OKUMURA, Y.; KUEBARA, K. New Process of Character Recognition using Sieving Method. **Information and Control Research**, [S.l.], v. 1, n. 1, p. 30-35, 1963.
- [KEM 94a] KEMPKA, Anthony. A. Activating Neural Networks. **A.I. Expert**, San Francisco, v. 9, n. 6, p. 33-41, June 1994.
- [KEM 94b] KEMPKA, Anthony. A. Activating Neural Networks. **A.I. Expert**, San Francisco, v. 9, n. 8, p. 42-50, Aug. 1994.
- [KER 88] KERRICK, David D.; BOVIK, Alan C. Microprocessor-based Recognition of Handprinted Characters From a Tablet Input. **Pattern Recognition**, Oxford, v. 21, n. 5, p. 525-537, 1988.
- [KEN 93] KENNEDY, Scott A. Five Ways to a Smarter Genetic Algorithm. **AI Expert**, San Francisco, v. 8, n. 12, p. 35-40, Dec. 1993.
- [KOH 88] KOHONEN, T. **Self-Organization and Associative Memory**. Heidelberg: Springer-Verlag, 1988. 312p.
- [KOH 90] KOHONEN, T. The Self-Organizing Map. **IEEE Neural Networks**, New York, v.78, n. 9, p. 74-87, 1990.
- [KOH 92] KOHONEN, T. **Som_Pak The Self-Organizing Map Program Package**. Helsinki: Helsinki University of Technology, Laboratory of Computer and Information Science, 1992.
- [KOS 91] KOSKO, Bart. **Neural Networks and Fuzzy Systems: A Dynamical Approach to Machine Intelligence**. Englewood Cliffs: Prentice Hall, 1991.
- [LAN 93] LANE, Alex. Programming with Genes. **AI Expert**, San Francisco, v. 8, n. 12, p. 16-20, Dec. 1993.
- [LAN 94] LANE, Alex. Genetic Cryptography. **AI Expert**, San Francisco, v. 8, n. 9, p.13-19, Sept. 1994.

- [LAW 92] LAWTON, George. Genetic Algorithms for Schedule Optimization. **AI Expert**, San Francisco, v. 7, n. 5, p. 23-27, May 1992.
- [LIP 87] LIPPMANN, Richard P. An Introduction to Computing with Neural Nets. **IEEE ASSP Magazine**, [S.l.], p. 4-22, 1987.
- [LIS 92] LISA, F.; CARRABINA, et al. **Feed Foward Network For Vehicle Licence Character Recognition**. Spain: Centre Nacional de Microeletrônica, Universitat Autònoma de Barcelona, 1992.
- [LYN 93] LYNCH, Gary; AMBROS-INGERSON, José. **Memory's Highway: Brain Rules for Encoding and Organizing Information**. California: University of California, 1993.
- [MAU 93] MAUREEN Caudill. A little Knowledge is a Dangerous Think. **AI Expert**, San Francisco, v. 8, n. 6, p. 16-23, June 1993.
- [MAR 91] MARGARITA, Sergio. Neural Networks, Genetic Algorithms and Stock Trading. In: KOHONEN, T. et al. (Eds.). **Artificial Neural Networks**. Amsterdam: North-Holland, 1991.
- [MAY 94] MAYYA, N.; RAJAN, V.T. **Recognition of Handwritten Characters by Voronoi Representations**. Gainesville, Fl: Departament of Computer & Information Sciences. University of Florida, 1994.
- [MAZ 94] MAZA, Michael de la; YURET, Deniz. Dynamic Hill Climbing Combine Genetic Algorithms, Hill Climbing, and Conjugate Gradient Methods to Become a New Optimization Algorithm. **AI Expert**, San Francisco, v. 9, n. 3, p. 26-31, Mar. 1994.
- [MCI 94] McINERNEY, M.; DHAWAN, A. Training the Self-Organizing Feature Map using Hybrids of Genetic and Kohonen methods. **IEEE Neural Networks**, Louisville, p. 641-644, 1994.
- [MIL 94] MILLAR, D.; CLARICI, E. Investigation of Back Propagation and Artificial Neural Networks in Modelling the Stress-Strain Behavior of Sandstone Rock. In: IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE, 1994. **Proceedings...** [S.l.:s.n.], 1994. v.5.
- [MIR 94] MIRANDA, Carlos Alberto; BOTELHO, Silvia S. da Costa. **Reconhecimento de Caracteres Numéricos Utilizando Redes Neurais Backpropagation**. Porto Alegre: CPGCC-UFRGS, 1994.
- [NEV 92] NEVES, Luíz Antônio da Costa Lima; CARVALHO FILHO, Edson Costa de Barros. Arquitetura Neural Integrada para Extração e Reconhecimento de Caracteres Alfanuméricos. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 9., 1992. **Anais...** [S.l.:s.n.], 1992.
- [OSO 91] OSÓRIO, Fernando dos Santos. **Um Estudo Sobre Reconhecimento Visual de Caracteres Através de Redes Neurais**. Porto Alegre: CPGCC - UFRGS, 1991.

- [RAM 94] RAMOS, Antônio Rogério Machado, BARONE, Dante. Aplicação de Algoritmo Genético no Controle de Trajetória de um Veículo. In: SIMPÓSIO BRASILEIRO DE REDES NEURAIS, 1., 1994, Caxambu, MG. **Anais...** Porto Alegre: Sociedade Brasileira de Computação, 1994. p.147-152.
- [RAM 95a] RAMOS, Antônio Rogério Machado; BARONE, Dante. Application of Hybrid System In Engineering Pattern Recognition Problems. In: INTERNATIONAL CONFERENCE ON ENGINEERING APPLICATIONS OF NEURAL NETWORKS, 1995, Helsinki - Finlândia. **Proceedings...** Helsinki: [s.n.], 1995.
- [RAM 95b] RAMOS, Antônio Rogério Machado; BARONE, Dante. Presentation of a Hybrid Evolutionary Classifier System. In: WORD CONGRESS ON NEURAL NETWORKS, 1995, Washington. **Proceedings...** Washinton: [s.n.], 1995.
- [RAM 95c] RAMOS, Antônio Rogério Machado; BARONE, Dante. Intelligent Solutions for Cibernetics Vehicle Control. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBENETICS, 1995, Vancouver. **Proceedings...** Vancouver: [s.n.], 1995.
- [RIB 94] RIBEIRO FILHO, José L.; TRELEAVEN, Philip C. Genetic Algorithm Programming Environments. **Computer**, New York, p. 28-43, June 1994.
- [RYB 93] RYBAK, I. A. et al. Modelling of the Neural Organization of the Visual Cortex and Some Issues of Image Processing by Neural Networks. In: NEURAL NETWORKS - THEORY AND ARCHITETURE PROCEEDINGS IN NON-LINEAR SCIENCE, 1993. **Proceedings...** [S.l.]:Manchester University Press, 1993.
- [ROB 85] DE ROBERTIS, E. D. P.; DE ROBERTIS, E. M. F. **Bases da Biologia Celular e Molecular**. Rio de Janeiro: Guanabara Koogan, 1985. 332 p.
- [ROC 93] ROCK, Denny; HIRSH, Joel. Will GAs Breed with Aerospace? **AI Expert**, San Francisco, v. 8, n. 12, p. 29-34, Dec. 1993.
- [ROG 92] ROGNVALDSSON, Thorsteinn. Pattern Recognition with Artificial Neural Networks - a Benchmark Study of Scaling Behavior. In: ICANN, 1992. **Proceedings...**[S.l.:s.n.], 1992.
- [SAN 94] SANTOS, Elisangela Melo; MONTENEGRO, Anselmo Antunes. Heurística Genéticas para o "Travelling Purchaser Problem". SIMPÓSIO BRASILEIRO DE REDES NEURAIS, 1., 1994, Caxambu, MG. **Anais...** Porto Alegre: Sociedade Brasileira de Computação, 1994. p. 153-159.

- [SIM 94] SIMÕES, Eduardo; FERREIRA, Adenilton; BARONE Dante. **Aplicação de RNAs não Supervisionadas ao Reconhecimento de Padrões em Telecomunicações**. Porto Alegre: CPGCC-Informática UFRGS, 1994.
- [SRI 94] SRINIVAS, M; PATNAIK, Lalit. Genetic Algorithms: A Survey. **Computer**, New York, p. 17-26, June 1994.
- [TOD 89] TODD, Peter M.; MILLER, Geoffrey F. Disigning Neural Networks using Genetic Algorithms. In: INTERNATIONAL CONFERENCE GENETIC ALGORITHMS, 3, 1989. **Proceedings...**[S.l.:s.n.], 1989.
- [TRE 92] TRELEAVEN, Philip; GOONATILAKE, Suran. Intelligent Financial Technologies. In: PARALLEL PROBLEM SOLVING FROM NATURE: APPLICATION IN STATISTICS AND ECONOMICS, 1992. **Proceedings...** [S.l.]: Statistical Office of the European Communities, 1992.
- [WER 90] WERBOS, Paul J. Back Propagation Through Time: What It Does and How to Do It. **IEEE Proceedings**, New York, v. 78, n. 10, p.1550-1560, Oct. 1990.
- [WHI 93] WHITE, David; LIGOMENIDES, Panos. **GANNet: A Genetic Algorithms for Optimizing Topology and Weights in Neural Network Design**. [S.l.]: University of Maryland, 1993.
- [WID 90] WIDROW, Bernard; LEHR, Michael. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and BackPropagation. **IEEE Proceedings**, New York, v. 78, n. 9, p. 1415-1442, Sept. 1990.
- [WIG 92] WIGGINS, Ralphe. Docking A Truck: A Genetic Fuzzy Approach. **AI Expert**, San Francisco, v. 7, n. 5, p. 28-35, May 1992.

Orientador.





CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Aplicação de Sistemas Híbridos em Problemas de Otimização

por

Antônio Rogério Machado Ramos

Dissertação apresentada aos Senhores:

Prof. Dr. José Nelson Amaral (PUCRS)

Prof. Dr. Philippe Olivier Alexandre Navaux

Prof. Dr. Paulo Martins Engel

Vista e permitida a impressão.
Porto Alegre, 26/04/2020

Prof. Dr. Dante Augusto Couto Barone,
Orientador.