

Projetos de Operadores Invariantes à Translação Via Treinamento de Redes Neurais

Robson Pequeno de Sousa

Tese de Doutorado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba - Campus II como parte dos requisitos necessários para obtenção do grau de Doutor no domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação -
Processamento de Imagens

João Marques de Carvalho, Ph.D.
Orientador

Francisco Marcos de Assis, Dr.
Orientador

Campina Grande, Paraíba, Brasil

©Robson Pequeno de Sousa, Fevereiro de 2000



S725p Sousa, Robson Pequeno de
Projetos de operadores invariantes a translacao via
treinamento de redes neurais / Robson Pequeno de Sousa. -
Campina Grande, 2000.
102 f. : il.

Tese (Doutorado em Engenharia Eletrica) - Universidade
Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Redes Neurais 2. Morfologia Matematica 3. Algoritmos
Geneticos 4. Tese I. Carvalho, Joao Marques II. Assis,
Francisco Marcos de III. Universidade Federal da Paraiba -
Campina Grande (PB) IV. Título

CDU 681.3(043)

**PROJETOS DE OPERADORES INVARIANTES A TRANSIÇÃO VIA
TREINAMENTO DE REDES NEURAIAS**

ROBSON PEQUENO DE SOUSA

Tese Aprovada em 07.02.2000


JOÃO MARQUES DE CARVALHO, Ph.D., UFPB

Orientador


FRANCISCO MARCOS DE ASSIS, Dr., UFPB

Orientador

GERARD JEAN FRANCIS BANON, Docteur d'État., INPE

Componente da Banca


NELSON DELFINO D'AVILA MASCARENHAS, Ph.D., UFSCar

Componente da Banca


CARLOS EDUARDO PEDREIRA, Ph.D., PUC-Rio

Componente da Comissão

JACQUES FACON, Dr. Ing., PUC-PR

Componente da Comissão


LÚCIO FLÁVIO C. PESSOA, Ph.D., Motorola-USA

Componente da Comissão

CAMPINA GRANDE - PB

Fevereiro - 2000

Dedicatória

À minha esposa Gilma e às minhas filhas Patricia e Kelly, que durante todos estes anos sempre me apoiaram e incentivaram.

Agradecimentos

A esta etapa fundamental de minha existência, objetivo maior como profissional, agradeço a Deus que nas horas mais difíceis me iluminou.

Agradeço de forma carinhosa e com muito amor a minha esposa Gilma e as minhas filhas Patricia e Kelly, que sempre confiaram nesta realização, bem como, aos meus pais Miguel e Ziza, aos meus irmãos Roberto, Ronaldo e Ronildo que de certa forma contribuíram para conclusão deste trabalho.

Agradeço de forma especial ao Dr. Lúcio Flávio Cavalcanti Pessoa, pesquisador da Motorola, EUA, que contribuiu com sugestões fundamentais que viabilizaram a conclusão desta Tese. Antes de tudo um grande amigo, que através de longas discussões por telefone e por e-mails contribuiu para enriquecer este trabalho, como também, em momentos de desânimo me deu força e incentivo para continuar esta difícil caminhada.

À Coordenação de Pós Graduação em Engenharia Elétrica (COPELE) da Universidade Federal da Paraíba (UFPB) e a todos que a compõem, os quais possibilitaram a conclusão desta Tese.

Aos meus orientadores Dr. João Marques de Carvalho e Dr. Francisco Marcos de Assis, pelo apoio total e irrestrito que me deram durante as pesquisas realizadas e na preparação desta Tese.

Agradeço ao Prof. Gerald J. F. Banon da Divisão de Processamento de Imagens do Instituto Nacional de Pesquisas Espaciais (INPE), motivador principal deste trabalho. Ao Prof Nelson D'Avila Mascarenhas que foi o meu primeiro contato e orientador no INPE, quando em minha estada naquela renomada instituição.

Também agradeço ao Prof. Misael Elias de Moraes do Departamento de Engenharia Elétrica da UFPB que possibilitou minha ida ao INPE.

Ao Laboratório de Processamento de Sinais - LAPS, àqueles que o compõem, Rinaldo e Paulo Márcio, que sempre me incentivaram.

Aos amigos e colegas da pós-graduação, em especial: Joseana Macedo, Luciana, Vania, Madeiro, Wamberto, Suzete, Michel, Tomaz, Madavan, Waslon, Gustavo, Josemar e Galdino. Em todos os momentos eles souberam externar sua amizade e preocupação nos momentos difíceis do trabalho.

Aos coordenadores Prof. Rubem Alves da Silva e Wilson Guerreiro Pinheiro do Laboratório de Ensaaios, Manutenção, Calibração, Aferição e Desenvolvimento - LEMCAD, que me apoiaram totalmente, reservando um ambiente ideal para o desenvolvimento desta Tese.

Ao amigo Luis Carlos Dantas Matias que me auxiliou na confecção das figuras desta tese, bem como o amigo Paulo de Almeida Farias que sempre estava preocupado com a conclusão desta Tese e sempre me deu seu apoio.

Agradeço ao Departamento de Matemática e Estatística da Universidade Estadual da Paraíba, que me deu um voto de confiança por unanimidade, ao me liberar de forma incondicional para a conclusão desta Tese.

Finalmente, ao povo brasileiro que com seus impostos contribuíram para a realização deste trabalho.

Resumo

O principal objetivo desta tese é o desenvolvimento de um método analítico para projeto de operadores invariantes à translação via treinamento de redes neurais. Uma nova arquitetura de rede neural, denominada Rede Neural Modular Morfológica (RNMM), é definida através de um resultado fundamental em representações mínimas para mapeamentos de conjuntos invariantes à translação via morfologia matemática, proposto por Banon e Barrera (1991). A arquitetura geral da RNMM é capaz de aprender operadores invariantes à translação, tanto do tipo binário como em níveis de cinza. Para o seu treinamento, são utilizadas idéias do algoritmo de retro-propagação do erro (RP) e a metodologia proposta por Pessoa e Maragos (1997) para superar o problema de não-diferenciabilidade das funções posto. Também é desenvolvido um treinamento alternativo da RNMM via algoritmos genéticos (AG) e realizada uma análise comparativa dos treinamentos RP e AG em problemas de restauração de imagens e reconhecimento de padrões. A estrutura da RNMM pode ser vista como um caso especial da rede neural morfológica/posto/linear (RN-MPL) proposta por Pessoa e Maragos (1997), mas com arquitetura e regras de treinamento próprias. O bom desempenho dos treinamentos RP e GA propostos é encorajador, oferecendo uma ferramenta alternativa de projeto para a importante classe de operadores invariantes à translação.

Abstract

The main objective of this thesis is to develop an analytic method for designing translation-invariant operators via neural network training. A new neural network architecture, called Modular Morphological Neural Network (MMNN), is defined using a fundamental result of minimal representations for translation-invariant set mappings via Mathematical Morphology, proposed by Banon and Barrera (1991). The MMNN general architecture is capable of learning both binary and gray-scale translation-invariant operators. For its training, ideas of the Back-Propagation (BP) algorithm and the methodology proposed by Pessoa and Maragos (1997) for overcoming the problem of non-differentiability of rank functions are used. Also is developed an alternative MMNN training method via Genetic Algorithms (GA), and provide a comparative analysis of BP vs. GA training in problems of image restoration and pattern recognition. The MMNN structure can be viewed as a special case of the Morphological/Rank/Linear Neural Network (MRL-NN) proposed by Pessoa and Maragos (1997), but with specific architecture and training rules. The effectiveness of the proposed BP and GA training algorithms for MMNNs is encouraging, offering alternative design tools for the important class of translation-invariant operators.

Lista de Figuras

1.1	Decomposição do operador invariante à translação via sup-geradores . .	4
1.2	Decomposição do operador invariante à translação via inf-geradores . .	5
1.3	Decomposição do operador invariante à translação por erosões	6
1.4	Decomposição do operador invariante à translação por dilatações	7
1.5	Perceptron Multi-camadas.	8
1.6	Rede neural modular com k níveis de hierarquia (grupo). Cada nível de hierarquia contém módulos, bem como redes portão. Σ é a unidade integradora da rede.	11
2.1	Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via decomposição de Matheron (erosões).	20
2.2	Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via decomposição de Matheron (dilatações).	21
2.3	Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via sup-geradores.	22
2.4	Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via inf-geradores.	23
3.1	RNMM com múltiplas saídas. Observe que cada módulo contém uma RNMM.	40
3.2	Imagens de treinamento da RNMM para filtragem de ruído. Imagens em níveis de cinza (a) e (b), imagens binárias (c) e (d)	51

3.3	Convergência da RNMM via decomposição de erosões através da NSR. A rede teve melhor comportamento de convergência nos casos (a) e (b) ($\mu = 0,01$ e $\mu = 0,1$).	52
3.4	Convergência da RNMM via decomposição por dilatações através da NSR. A rede teve melhor comportamento de convergência nos casos (a) e (b) ($\mu = 0,01$ e $\mu = 0,1$).	53
3.5	Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 1$ a rede não aprendeu.	54
3.6	Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,01$ a rede aprendeu.	55
3.7	Imagens de treinamento e teste da RNMM via decomposição por 8 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 1$ a rede não aprendeu.	56
3.8	Imagens de treinamento e teste da RNMM via decomposição por 8 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,01$ a rede aprendeu.	57
3.9	Imagens de treinamento e teste da RNMM via decomposição por 25 erosões. (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,1$	58
3.10	As imagens mostram a saída do filtro projetado com a solução obtida via treinamento da rede com 25 erosões. (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,1$	59

3.11	Imagens de treinamento da RNMM para o problema de extração de bordas. (a) imagem binária de entrada, (b) imagem binária desejada (gradiente), (c) imagens em níveis de cinza, (d) imagem em níveis de cinza desejada (gradiente).	64
3.12	Imagens de treinamento e teste da RNMM via decomposição por 8 superadores. (a) Imagem de entrada, (b) saída final do treinamento, (c) entrada de teste, (d) saída do teste (gradiente obtido). Imagem em níveis de cinza.	65
3.13	Imagens de treinamento e teste da RNMM via decomposição por 8 superadores. (a) Imagem de entrada, (b) saída final do treinamento, (c) entrada de teste, (d) saída do teste (gradiente obtido). Imagem binária	66
3.14	Conjunto de treinamento X_1	67
3.15	Conjunto de treinamento X_2	68
4.1	Cruzamento simples	75
4.2	Ciclo do algoritmo genético e seus principais operadores.	75
4.3	Imagens de treinamento e teste da RNMM via decomposição por 25 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.	82
4.4	Imagens de treinamento e teste da RNMM via decomposição por 25 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.	83
4.5	Imagem filtrada com a solução da RNMM via treinamento AG: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Decomposição por 25 erosões.	84
4.6	Imagem filtrada com a solução da RNMM via treinamento AG: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Decomposição por 25 dilatações.	85

4.7	Convergência da RNMM via treinamento RP e AG. (a) decomposição por erosões e (b) decomposição por erosões	87
4.8	Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.	88
4.9	Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por RP.	89
4.10	Imagens de treinamento e teste da RNMM via decomposição por 8 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.	90
4.11	Conjunto de treinamento da RNMM para o experimento conjugado, isto é, filtragem de ruído e extração de bordas.	92
4.12	Imagens de treinamento da RNMM para o problema de extração de bordas e ruído, decomposição por 8 sup-geradores: (a) imagem binária de entrada, (b) imagem binária desejada, (c) imagens de teste, (d) saída do operador extrator de borda e ruído. Treinamento via AG.	93
4.13	Imagens de treinamento da RNMM para o problema de extração de bordas e ruído, decomposição por 8 sup-geradores: (a) imagem binária de entrada, (b) imagem binária desejada, (c) imagens de teste, (d) saída do operador extrator de borda e ruído. Treinamento via RP.	94

Lista de Tabelas

3.1	Desempenho da RNMM para decomposição via erosões.	60
3.2	Desempenho da RNMM Para decomposição via dilatações.	60
3.3	Comparações entre a RNMM com o filtro da mediana. Observa-se que o melhor resultado foi para o caso em que o número de erosões que compõem a arquitetura da RNMM é igual a 25	61
3.4	Desempenho do classificador projetado via treinamento da RNMM com o conjunto de treinamento X_1	69
3.5	Desempenho do classificador projetado via treinamento da RNMM com o conjunto de treinamento X_2	70
4.1	Comparação entre os treinamentos da RNMM via AG e retro-propagação do erro, melhor resultado para o treinamento via retro-propagação.	86
4.2	Comparação entre os treinamentos da RNMM via AG e RP. Melhor resultado para o treinamento via AG.	87
4.3	Esforço computacional: Número de operações realizadas entre os dois tipos de treinamento (RP e AG).	91

Índice

1	Introdução	1
1.1	Motivação	2
1.2	Redes neurais	3
1.2.1	Arquitetura da Rede Neural Modular	9
1.3	Proposta da Tese	10
1.4	Organização Desta Tese	13
2	Rede Neural Modular Morfológica	14
2.1	Aspecto da Morfologia Matemática	14
2.2	Conceitos Básicos em Morfologia Matemática	15
2.3	Definição e Arquitetura da RNMM	19
2.3.1	Fundamentos da RNMM para a decomposição de Banon e Barrera	24
2.3.2	Fundamentos da RNMM para decomposição de Matheron	27
2.4	Considerações Finais	28
3	Treinamento da RNMM Via Retro-Propagação do Erro	30
3.1	Treinamento da RNMM para a decomposição de Banon e Barrera	31
3.1.1	Determinação das derivadas das equações de treinamento da RNMM, decomposição Banon e Barrera.	37
3.1.2	Algoritmo de treinamento da RNMM para decomposição via sup- geradores ou inf-geradores	41
3.2	Treinamento da RNMM para a decomposição de Matheron	43

3.2.1	Determinação das derivadas da equação de treinamento da RNMM, decomposição de Matheron	45
3.2.2	Algoritmo de treinamento da RNMM para decomposição via erosões ou dilatações	46
3.3	Aplicação em filtragem de imagens (RNMM via decomposição de Matheron)	48
3.3.1	Simulações	49
3.4	Aplicação em problemas de extração de bordas (RNMM via decomposição de Banon e Barrera)	62
3.4.1	Simulação	62
3.5	Aplicação da RNMM em Reconhecimento de Padrões (decomposição de Banon e Barrera)	63
3.5.1	Experimentos	63
3.6	Considerações Finais	70
4	Treinamento da RNMM Via Algoritmo Genético	72
4.1	Introdução	72
4.2	Algoritmo Genético	73
4.3	Usando Algoritmo Genético para treinar a RNMM	76
4.3.1	Detalhando o algoritmo de treinamento AG da RNMM (decomposição de Banon e Barrera)	77
4.3.2	Detalhando o algoritmo de treinamento AG da RNMM (decomposição de Matheron)	79
4.4	Aplicações em Restauração de Imagens (decomposição de Matheron)	80
4.4.1	Simulações	81
4.5	Aplicação em extração de Bordas (decomposição de Banon e Barrera)	91
4.5.1	Simulação	92
4.6	Considerações Finais	95
5	Contribuições Desta Tese	96
5.1	Sugestões Para Trabalhos Futuros	98

Capítulo 1

Introdução

O projeto de operadores invariantes à translação, tem se tornado um problema de fundamental importância em Morfologia Matemática, possibilitando diversos tipos de aplicações em processamento e análise de imagens, como: restauração, extração de bordas, reconhecimento de objetos em imagens, dentre outros. Dentre os trabalhos publicados na área de projetos de operadores destacam-se os seguintes:

1. Dougherty e Loce [1], projetam operadores sub-ótimo do tipo crescente, baseado na decomposição de Matheron [2]. Utilizam métodos nos quais o operador morfológico é tratado como um estimador estatístico. No entanto, as aplicações estão restritas a imagens do tipo binárias.
2. Barrera, Dougherty e Tomita [3], apresentam uma generalização da formalização proposta por [1], para projeto de um conjunto de operadores sub-ótimos. Este trabalho utiliza o contexto da teoria da aprendizagem computacional. Também neste caso, as aplicações estão restrita a imagens binárias.
3. Yang e Maragos[4], projetam operadores (classificadores min-max) de acordo com a decomposição de Matheron [2]. Utilizam o critério do erro médio quadrático de tal forma que uma função custo seja minimizada. A metodologia empregada para o treinamento do classificador min-max, é baseada na metodologia de Salembier [5].

4. Harvey e Marshall [6], usam algoritmos genéticos do tipo simples [7] no projeto de filtros (operadores) morfológicos ótimos. O objetivo é determinar o elemento estruturante ótimo que parametriza o operador extrator de ruído. Porém, não estenderam a aplicação para a decomposição de Matheron [2] e nem para o caso mais geral da decomposição de Banon e Barrera [8].
5. Oliveira [9], generalizou o trabalho de Harvey e Marshall [6] ao implementar a decomposição de Banon e Barrera [8] via treinamento de algoritmo genético simples [7].
6. Tavares [10], apresenta uma rede neural morfológica em que a aprendizagem é não supervisionada, do tipo competitivo. Esta rede se propõe a detectar padrões na presença de ruído do tipo aditivo, ou a reconhecer padrões com pequenas deformações.

Mostramos acima algumas referências importantes na área de projetos de operadores invariantes à translação. Nesta tese vamos definir uma nova arquitetura de rede neural do tipo supervisionada, satisfazendo as condições dos teoremas da decomposição de Banon e Barrera [8] e de Matheron [2]. A mesma será utilizada em projetos de operadores morfológicos invariantes à translação, tanto para processamento de imagens binárias, como para imagens em níveis de cinza.

1.1 Motivação

Banon e Barrera [8] apresentaram um resultado importante para a área da Morfologia Matemática, que é o teorema da decomposição de operadores invariantes à translação, o qual mostra que todo operador invariante à translação pode ser representado como a união ou como a interseção de operadores do tipo sup-geradores ou do tipo inf-geradores, respectivamente. Isto significa que uma operação complexa pode ser decomposta em várias operações simples. Este teorema pode ser aplicado em diversas classes de problemas, tais como: extração de bordas, reconhecimento de padrões, restauração de imagens, etc.

O objetivo principal desta tese é a elaboração de um algoritmo de treinamento de redes neurais como forma de resolver o projeto de operadores invariantes à translação para processamento e análise de imagens através do teorema de Matheron [2] e do caso mais geral, o teorema de Banon e Barrera [8]. De acordo com o teorema de Banon e Barrera [8], o operador é decomposto em uma sequência de operações simples do tipo erosão (ε) e anti-dilatação (δ^a). A interseção de uma erosão com uma anti-dilatação produz um sup-gerador. A saída final do sistema é a integração de todas os sup-geradores através de uma operação de união. Uma estrutura que implementa estas operações é mostrada na figura 1.1, na qual pode ser observada que os sup-geradores são produzidos em paralelo. Dualmente, no caso da decomposição via inf-geradora, a união de uma dilatação (δ) com uma anti-erosão (ε^a) produz um inf-gerador. A saída final do sistema é a interseção de todos os inf-geradores como ilustrado na figura 1.2. No caso de Matheron [2] o operador é decomposto em uma sequência de erosões ou de dilatações, a saída final sendo obtida através de uma operação de união ou de interseção, respectivamente, como ilustrado nas figuras 1.3 e 1.4.

Portanto, a contribuição principal desta tese é a definição analítica de um método de treinamento que possibilite a realização de projetos de operadores invariantes à translação via rede neural, utilizando uma arquitetura da rede equivalente as decomposições de Matheron [2] e de Banon e Barrera [8], ilustradas nas figuras 1.1, 1.2, 1.3 e 1.4.

1.2 Redes neurais

Redes neurais são sistemas de processamento da informação, construídos a partir de elementos de processadores simples, massivamente interconectados, capazes de aprender, memorizar e generalizar. O conhecimento é adquirido pela rede através de um processo de aprendizado, sendo armazenado nas interconexões (pesos sinápticos) entre os elementos (neurônios).

A figura 1.5 nos mostra uma arquitetura de rede do tipo Perceptron de Multicamadas com vários aprendizados, ou camadas ocultas, em que cada neurônio tem a forma

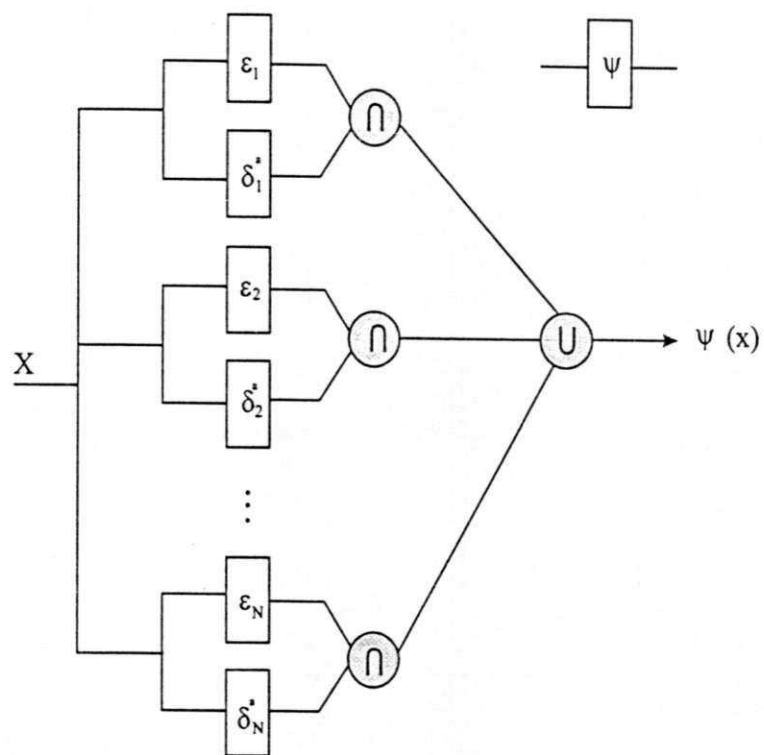


Figura 1.1: Decomposição do operador invariante à translação via sup-geradores

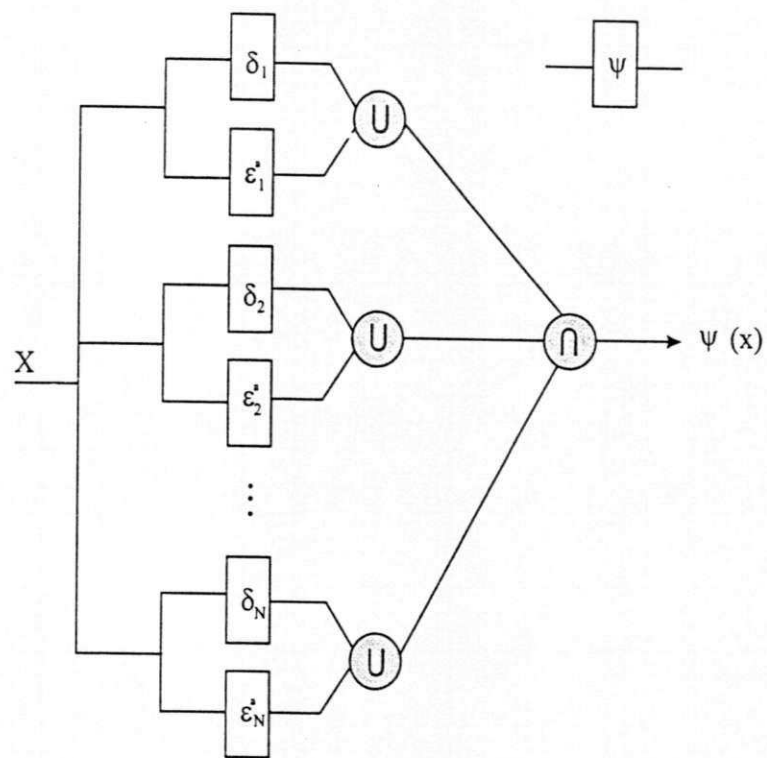


Figura 1.2: Decomposição do operador invariante à translação via inf-geradores

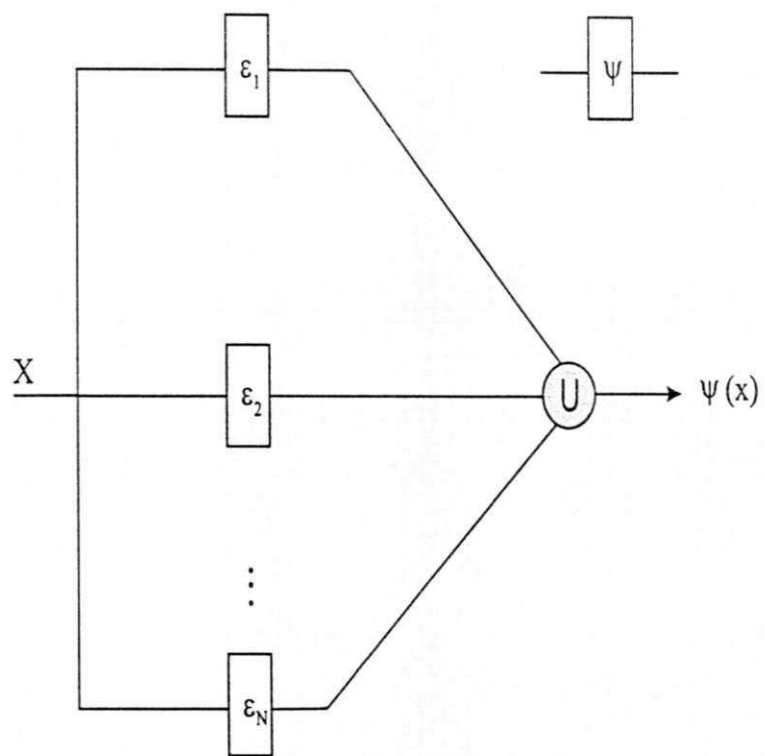


Figura 1.3: Decomposição do operador invariante à translação por erosões

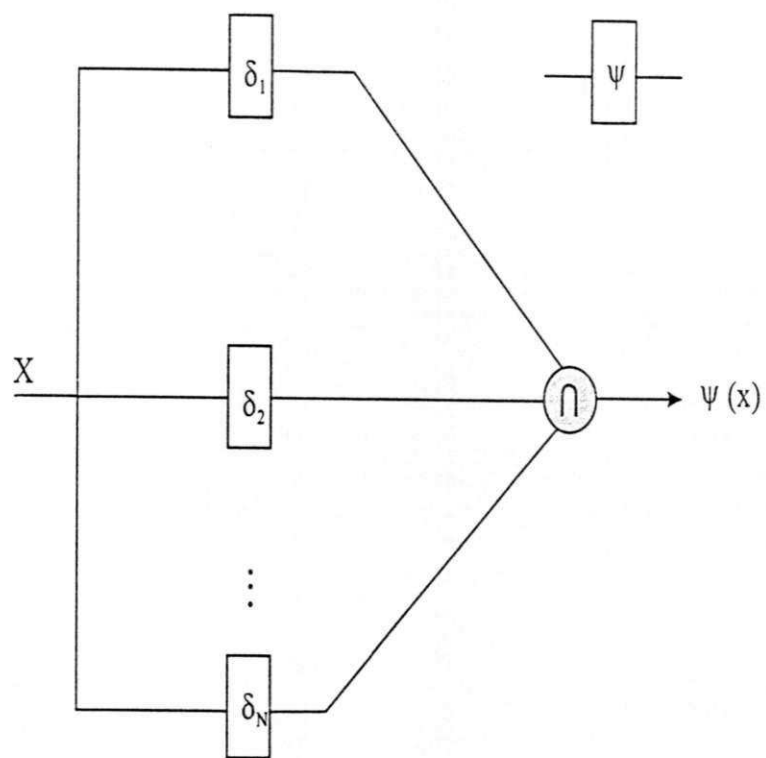


Figura 1.4: Decomposição do operador invariante à translação por dilatações

do modelo perceptron [11]. O algoritmo para treinamento deste tipo de rede neural foi desenvolvido em 1986 por McClelland e Rumelhart [12], tornando-se conhecido como Retro-Propagação do Erro (RP). Este algoritmo é utilizado para modificação dos pesos das conexões da rede neural, sendo aplicado à rede neural com uma ou mais camadas escondidas, em que cada camada é globalmente conectada com a camada subsequente.

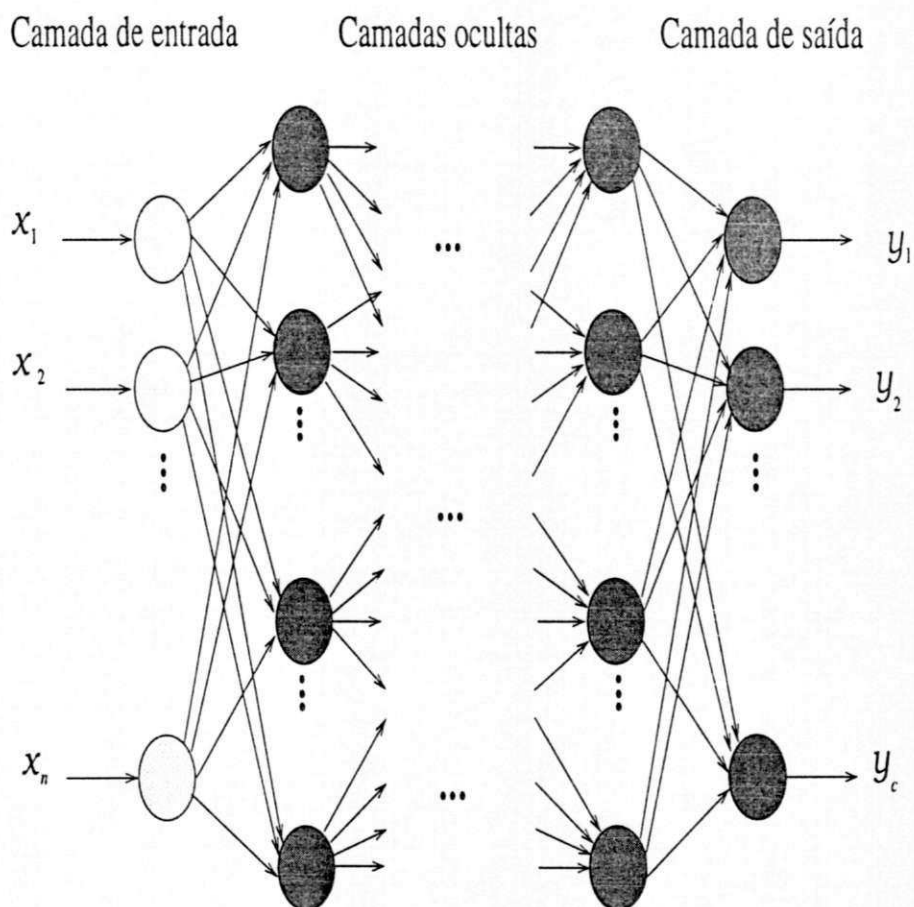


Figura 1.5: Perceptron Multi-camadas.

Existe também a classe das redes neurais morfológicas [13], que diferem das redes neurais clássicas pela computação algébrica realizada nos nós de cada rede. Na rede neural clássica a saída em cada nó é produzida por uma função de ativação (sigmóide), aplicada sobre a soma ponderada das entradas. Por outro lado, a computação em cada nó da rede neural morfológica (RNM) é feita através de operações de máximo e

mínimo definidas sob o contexto de álgebra das imagens [14]. O treinamento básico da RNM é feito através de uma tabela de correções de peso (*look-up table*). O grau de complexidade desta tabela aumenta consideravelmente quando se trabalha com imagens em níveis de cinza, tornando assim praticamente inviável a sua aplicação em termos práticos.

Em [15], Shalkoff apresenta uma rede neural do tipo morfológica com aprendizagem baseada na regra delta [16] com função de ativação aproximada por uma função de transição brusca. Porém, esta rede está limitada à aprendizagem de operadores binários do tipo erosão e dilatação.

Haykin [17], Jordan e Jacobs [18], utilizam um tipo de rede neural (figura 1.6) com uma arquitetura semelhante à arquitetura da figura 1.1, conhecida por rede neural modular. Esta rede foi projetada para resolver um problema de regressão não linear, tendo sido utilizada também por Menu [19] para resolver problemas de reconhecimento de caracteres manuscritos. Uma rede neural é dita modular se a computação desempenhada pela rede pode ser decomposta em dois ou mais módulos (sub-sistemas) que operam independentemente. As saídas dos módulos são mediadas por uma unidade integradora que não permite que as informações da saída sejam realimentadas. A rede neural modular tem a incubência de resolver problemas computacionalmente complexos, dividindo-os em substratos e combinando soluções individuais destes substratos. O modelo modular foi inspirado em fundamentos neuro-biológicos, baseados na modularidade da arquitetura do sistema nervoso dos vertebrados. O princípio de modularidade é a decomposição de um problema complexo em um conjunto de problemas de complexidades menores.

1.2.1 Arquitetura da Rede Neural Modular

A arquitetura de uma rede neural modular [18] é composta de um número de redes especialistas adaptativas (módulos) reunidas em grupos, que mapeiam o vetor de entrada \vec{x} para o vetor de saída \vec{y} (figura 1.6). Existe também um número de redes portão que realizam a atividade de mediador entre módulos do mesmo grupo. A saída do i -ésimo grupo é dada por:

$$y_i = \sum_j g_{j|i} y_{ij} \quad (1.1)$$

em que $g_{j|i}$ é a função de ativação da j -ésima unidade de saída da rede portão do i -ésimo grupo. A saída geral da rede modular é dada por:

$$y = \sum_i g_i y_i \quad (1.2)$$

em que g_i é a função de ativação da unidade de saída da rede portão do nível superior. Jacob [18] assume que a saída da rede portão é dada por:

$$g_i = \frac{e^{u_i}}{\sum_j e^{u_j}} \quad (1.3)$$

e

$$g_{j|i} = \frac{e^{u_{ji}}}{\sum_k e^{u_{ki}}} \quad (1.4)$$

em que u_i e $u_{j|i}$ são as somas ponderadas dos pesos da rede portão com a entrada geral da rede.

O sistema de aprendizagem da rede modular pode ser supervisionado ou não supervisionado. O processo de ajuste dos pesos é dividido em duas etapas: (a) ajuste dos pesos das redes especialistas e (b) ajuste dos pesos das redes portão. Geralmente, no caso supervisionado usa-se o algoritmo do gradiente descendente para treinar as redes especialistas com uma única camada de neurônios. Caso as redes especialistas tenham mais de uma camada de neurônios, então usa-se o algoritmo RP para treiná-las.

1.3 Proposta da Tese

Como solução da proposta desta tese, apresentamos uma nova arquitetura de rede neural modular que difere da rede mostrada na figura 1.6 em três aspectos fundamentais:

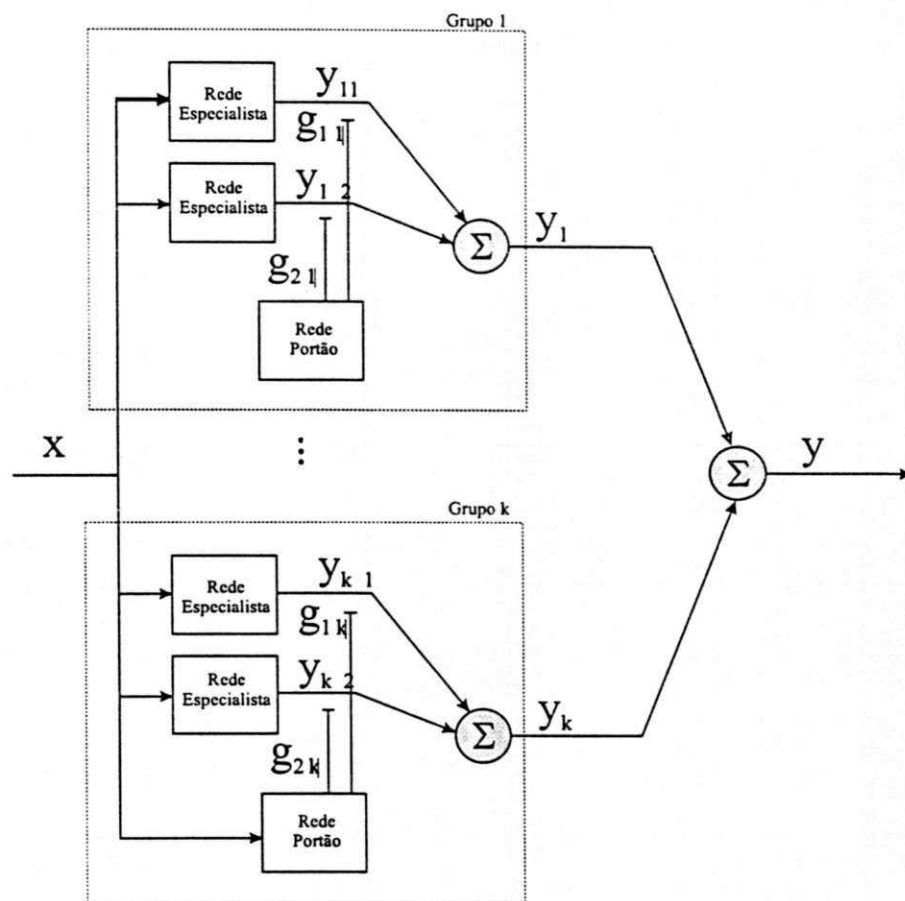


Figura 1.6: Rede neural modular com k níveis de hierarquia (grupo). Cada nível de hierarquia contém módulos, bem como redes portão. Σ é a unidade integradora da rede.

1. A não inclusão da rede portão, devido à natureza das operações morfológicas.
2. A utilização de um novo algoritmo de aprendizagem, que será posteriormente definido.
3. A definição de cada módulo da rede como um operador morfológico. Utilizado no lugar de redes neurais especialistas.

Com esta rede pretendemos resolver, via treinamento, o problema de projetos de operadores invariantes à translação, de acordo com os teoremas da decomposição Banon e Barrera [8] e de Matheron [2]. A rede é bastante geral, podendo aprender o operador para imagens binárias assim como para imagens em níveis de cinza. Esta nova rede será denominada de Rede Neural Modular Morfológica (RNMM).

Quanto ao treinamento da RNMM apresentaremos dois métodos originais, baseados no algoritmo de retro-propagação do erro (RP) [12] e no algoritmo genético (AG) [7], respectivamente. O treinamento (RP) da RNMM toma como ponto de partida o trabalho de Pessoa e Maragos [20], que apresenta uma classe geral de filtros híbridos, denominados filtros morfológicos/posto/linear (MPL), com planejamento ótimo adaptativo. O filtro MPL consiste de uma combinação linear entre um filtro morfológico/posto e um filtro linear, cuja componente não linear é baseada na função posto. Os operadores morfológicos de erosão e dilatação podem ser obtidos como casos especiais da função posto. Pessoa também apresenta em [21] uma classe bastante geral de redes neurais denominadas redes neurais MPL (RN - MPL), cuja unidade fundamental (nó), é o filtro MPL. A RN - MPL unifica o perceptron multiaprendizado com redes neurais morfológicas [22].

A RNMM proposta nesta tese pode ser vista como um caso especial da RN - MPL. Entretanto, a arquitetura e a regra de treinamento utilizadas na RNMM são próprias, representando contribuições originais deste trabalho.

1.4 Organização Desta Tese

- No capítulo 2, é dada uma introdução à morfologia matemática bem como a definição e fundamentação da Rede Neural Modular Morfológica.
- No Capítulo 3, o algoritmo de treinamento da RNMM via retro-propagação do erro é apresentado, com resultados de simulações em problemas de restauração de imagens, extração de bordas e na área de reconhecimento de padrões.
- No capítulo 4, o treinamento da RNMM é feito através de algoritmo genético. Resultados comparativos entre os dois treinamentos propostos são avaliados.
- No capítulo 5, apresentamos as conclusões e propostas para trabalhos futuros.

Capítulo 2

Rede Neural Modular Morfológica

Neste capítulo apresentaremos os conceitos básicos da morfologia matemática necessários para o entendimento do problema proposto nesta tese, bem como a definição de uma nova arquitetura de rede neural, a qual denominamos de Rede Neural Modular Morfológica (RNMM).

2.1 Aspecto da Morfologia Matemática

Morfologia Matemática é uma das disciplinas que compõem a área de Processamento Digital de Imagens, a qual utiliza um enfoque não linear em processamento de imagens, envolvendo análise geométrica pelo uso de elementos estruturantes. Os elementos estruturantes são pequenos padrões que operam no domínio espacial e extraem informações sobre formas geométricas presentes na imagem. A morfologia matemática originou-se em meados dos anos sessenta, tendo sido desenvolvida por Matheron [2] e Serra [23], na Escola de Minas de Paris. O uso inicial da morfologia matemática foi para processar imagens booleanas. Posteriormente, Sternberg [24] e Serra [23] estenderam os conceitos para imagens com valores em níveis de cinza, usando noções de sombra. Os fundamentos matemáticos rigorosos da morfologia matemática em termos da álgebra dos reticulados foram estabelecidos por Heijmans [25]. Dentre os diversos resultados da morfologia matemática, Banon e Barrera [8] demonstraram que todo operador invariante à translação, não necessariamente crescente ou decrescente, pode

ser decomposto em uma combinação de operadores básicos que são: dilatação, erosão, anti-dilatação e anti-erosão. O trabalho de Banon e Barrera [8] também mostra a caracterização dos elementos estruturantes na decomposição. Este resultado fundamental é uma generalização do teorema da decomposição canônica para operadores crescentes de Matheron [2] e da representação de Maragos [26]. Posteriormente, Banon e Barrera [27] estenderam os resultados para imagens em níveis de cinza.

Neste capítulo os conceitos e definições da morfologia matemática apresentados são aqueles estritamente necessários para o entendimento do trabalho. Para uma abordagem mais completa veja, [28], [25] e [23].

2.2 Conceitos Básicos em Morfologia Matemática

As operações morfológicas com imagens binárias serão definidas sob o ponto de vista da teoria dos conjuntos. A morfologia matemática clássica fundamenta-se em duas operações básicas: soma e subtração de Minkowski [29].

Definição : Dados dois conjuntos $X \subset R^N$ e $B \subset R^N$, as adições e subtrações de Minkowski são definidas respectivamente por,

$$X \oplus B = \bigcup_{b \in B} X_b \quad (2.1)$$

$$X \ominus B = \bigcap_{b \in B^r} X_b \quad (2.2)$$

em que, $X_b = \{x + b : x \in X\}$ e $B^r = \{-b : b \in B\}$, B^r é a reflexão de B .

A transformação de X em $X \oplus B$ e X em $X \ominus B$ são denominadas de dilatação e erosão, respectivamente, nas quais o conjunto X representa a imagem de entrada e o conjunto B o elemento estruturante. Na prática assume-se que o elemento estruturante B inclui $0 \in R^N$. Todas as transformações da morfologia matemática são geradas da combinação de quatro operações básicas, dilatação, erosão, anti-dilatação e anti-erosão [8], definidas a seguir:

$$\text{Dilatação : } \delta_B(X) = X \oplus B \quad (2.3)$$

$$\text{Erosão : } \varepsilon_B(X) = X \ominus B \quad (2.4)$$

$$\text{Anti-Dilatação : } \delta_B^a(X) = (X \oplus B^{rc})^c \quad (2.5)$$

$$\text{Anti-Erosão : } \varepsilon_B^a(X) = (X \ominus B^{rc})^c \quad (2.6)$$

Em que, $B^{rc} = \{-b : b \notin B\}$ é o complemento da reflexão de B . As equações 2.5 e 2.6 foram obtidas dualmente de acordo com [8, 1785-1786].

Seja $E = R^N$, definiremos $P(E)$ como a coleção de todos os subconjuntos de E . Um operador Ψ é um mapeamento do tipo $\Psi : P(E) \rightarrow P(E)$.

Definição : Sejam $X \in P(E)$ e $\vec{h} \in E$ um vetor. O translado de X por \vec{h} é definido por

$$X_{\vec{h}} = \{x + \vec{h} : x \in X\}. \quad (2.7)$$

Definição : Um operador $\Psi : P(E) \rightarrow P(E)$, é invariante à translação se

$$\Psi(X_{\vec{h}}) = (\Psi(X))_{\vec{h}} \quad (2.8)$$

em que, $X \in P(E)$ e $\vec{h} \in E$.

Definição : Um operador $\Psi : P(E) \rightarrow P(E)$ é crescente se $X \subset Y \Rightarrow \Psi(X) \subset \Psi(Y), \forall X, Y \in P(E)$.

Definição : Um operador $\Psi : P(E) \longrightarrow P(E)$ é decrescente se $X \supset Y \Rightarrow \Psi(X) \supset \Psi(Y), \forall X, Y \in P(E)$.

Definição : Seja $L \subset E$ finito. Um operador $\Psi : P(E) \longrightarrow P(E)$ é definido como operador janela L se e somente se, para todo $x \in E$,

$$x \in \Psi(X) \Leftrightarrow x \in \Psi(X \cap L_x). \quad (2.9)$$

Observe que a operação acima é uma operação local, em que L_x é o translado de L por x . O operador de janela é denominado representação morfológica padrão [3].

Uma transformação invariante à translação mais geral de imagens em morfologia matemática é a transformação $\lambda_{(A,B)}$, denominada de transformação sup-geradora e sua dual $\mu_{(A,B)}$, denominada transformação inf-geradora. Em casos particulares estes operadores podem ser utilizados para detectar pequenos padrões na imagem.

Definição : Seja $L \subseteq E$, L finito, sejam $A, B \subset P(E)$, $A \subseteq B$. Os operadores sup-gerador e inf-gerador (relativo a janela L) são definidos respectivamente por:

$$\lambda_{(A,B)}^L = \varepsilon_A(X) \cap \delta_B^a(X) \quad (2.10)$$

$$\mu_{(A,B)}^L = \delta_A(X) \cup \varepsilon_B^a(X) \quad (2.11)$$

As operações 2.10 e 2.11, são operações locais e invariantes à translação.

O teorema da decomposição canônica de Matheron [2] para operadores crescentes, bem como o caso mais geral para operadores não crescentes, que é o teorema da decomposição de Banon e Barrera [8], são fundamentais para definição e composição da Rede Neural Modular Morfológica (RNMM). O teorema da decomposição de Banon e Barrera [8] a princípio foi enunciado para imagens tipo binária [8], posteriormente para imagens em níveis de cinza [27]. Estes teoremas são enunciados a seguir:

Teorema da Decomposição via sup-geradores (Banon e Barrera): Seja $\Psi : P(E) \rightarrow P(E)$ um operador de janela (relativo a janela L) invariante à translação. Então existe uma família de subconjuntos $A_k \subset P(E)$ e uma família de subconjuntos $B_k \subset P(E)$, $k = 1, 2, \dots, N$, com $A_k \subset B_k$, tal que, a decomposição de Ψ via sup-geradores é dada por,

$$\Psi = \bigcup_k^N \lambda_{(A_k, B_k)}^L. \quad (2.12)$$

Teorema da Decomposição via inf-geradores (Banon e Barrera): Seja $\Psi : P(E) \rightarrow P(E)$ um operador de janela (relativo a janela L) invariante à translação. Então existe uma família de subconjuntos $A_k \subset P(E)$ e uma família de subconjuntos $B_k \subset P(E)$, $k = 1, 2, \dots, N$, com $A_k \subset B_k$, tal que, a decomposição de Ψ via inf-geradores é dada por,

$$\Psi = \bigcap_k^N \mu_{(A_k, B_k)}^L. \quad (2.13)$$

Teorema da Decomposição via erosões (Matheron): Seja $\Psi : P(E) \rightarrow P(E)$ um operador de janela (relativo a janela L) crescente e invariante à translação. Então existe uma família de subconjuntos $A_k \subset P(E)$, $k = 1, 2, \dots, N$, tal que, a decomposição de Ψ via erosões é dada por,

$$\Psi = \bigcup_k^N \varepsilon_{A_k}^L(X). \quad (2.14)$$

Teorema da Decomposição via dilatações (Matheron): Seja $\Psi : P(E) \rightarrow P(E)$ um operador de janela (relativo a janela L) crescente e invariante à translação. Então existe uma família de subconjuntos $A_k \subset P(E)$, $k = 1, 2, \dots, N$, tal que, a decomposição de Ψ via dilatações é dada por,

$$\Psi = \bigcap_k^N \delta_{A_k}^L(X). \quad (2.15)$$

Os teoremas da decomposição de Matheron [2] nos dizem que qualquer operador Ψ crescente invariante à translação, pode ser decomposto como a união de erosões ou a interseção de dilatações, em que N representa o número de erosões e de dilatações na decomposição. Os teoremas da decomposição de Banon e Barrera nos dizem que qualquer operador não necessariamente crescente, pode ser decomposto como a união de operadores do tipo sup-geradores ou como a interseção de operadores do tipo inf-geradores, respectivamente, em que N representa o número de sup-geradores e inf-geradores na decomposição. Os resultados são extensivos a imagens em níveis de cinza [27].

De posse dos fundamentos básicos da morfologia matemática e dos principais resultados que fundamentam a RNMM, na próxima seção definiremos as equações que compõem a rede, assim como sua arquitetura.

2.3 Definição e Arquitetura da RNMM

A arquitetura da RNMM proposta neste trabalho é semelhante à arquitetura da rede neural modular [17], exceto por duas diferenças: a primeira é que não será necessário a mediação da rede portão, devido à natureza das operações morfológicas; a segunda diferença reside no algoritmo de aprendizagem. As figuras 2.1, 2.2, 2.3 e 2.4 mostram as possíveis arquiteturas da RNMM, as quais representam, respectivamente, a decomposição do operador morfológico via erosões, dilatações, sup-geradores e inf-geradores. Observa-se que esta rede contém vários módulos paralelos que operam independentes um do outro. Cada módulo representa uma operação morfológica padrão, ou seja, erosão, dilatação, anti-erosão e anti-dilatação.

Para a decomposição do operador via erosões (equação 2.14), a saída de cada módulo da rede (figura 2.1) irá convergir para uma unidade integradora total que fará uma operação de união (máximo); de forma dual para a decomposição via dilatações (equação 2.15), a saída de cada módulo da rede (figura 2.2) irá convergir para uma unidade integradora total que fará uma operação de interseção (mínimo), entre elas.

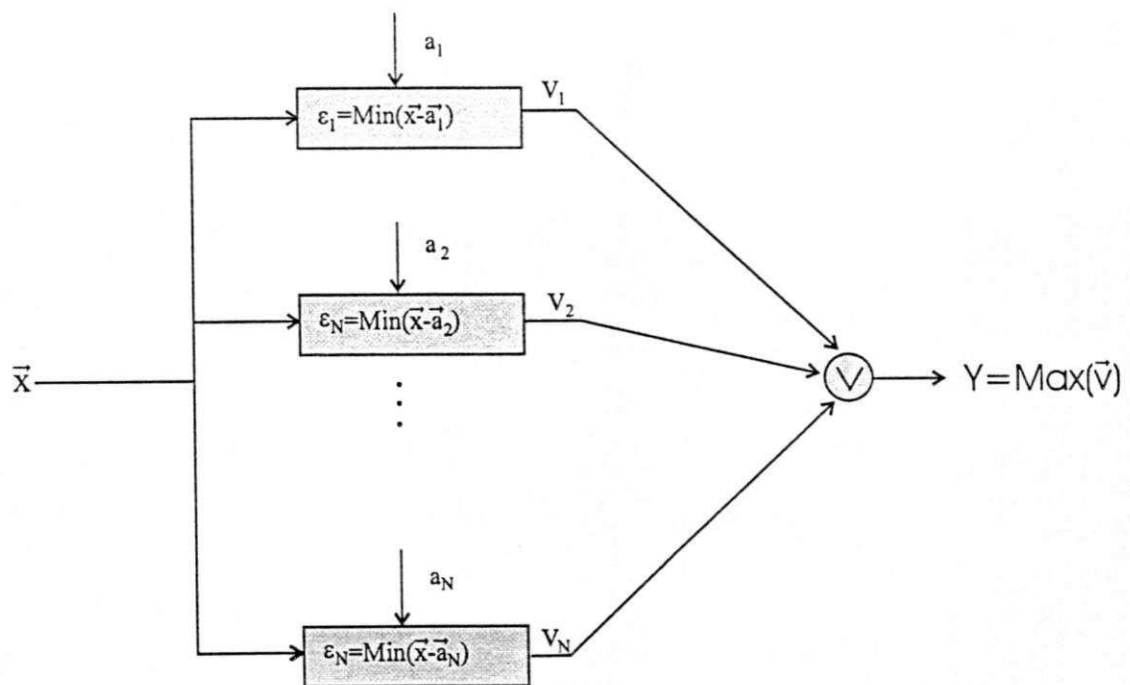


Figura 2.1: Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via decomposição de Matheron (erosões).

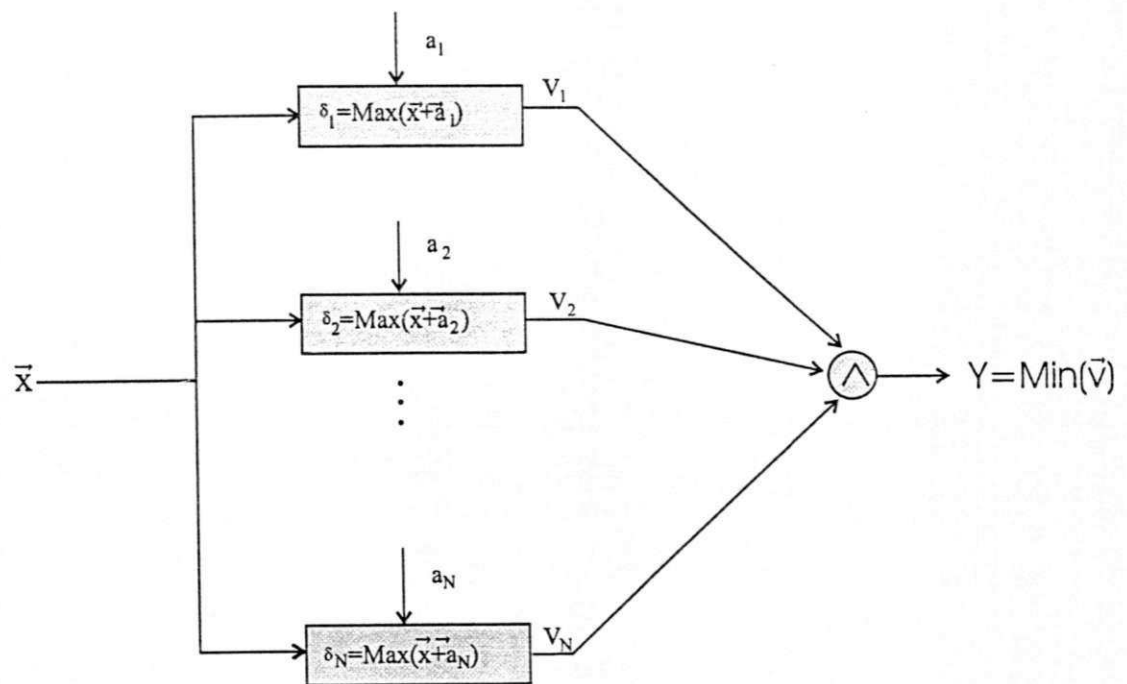


Figura 2.2: Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via decomposição de Matheron (dilatações).

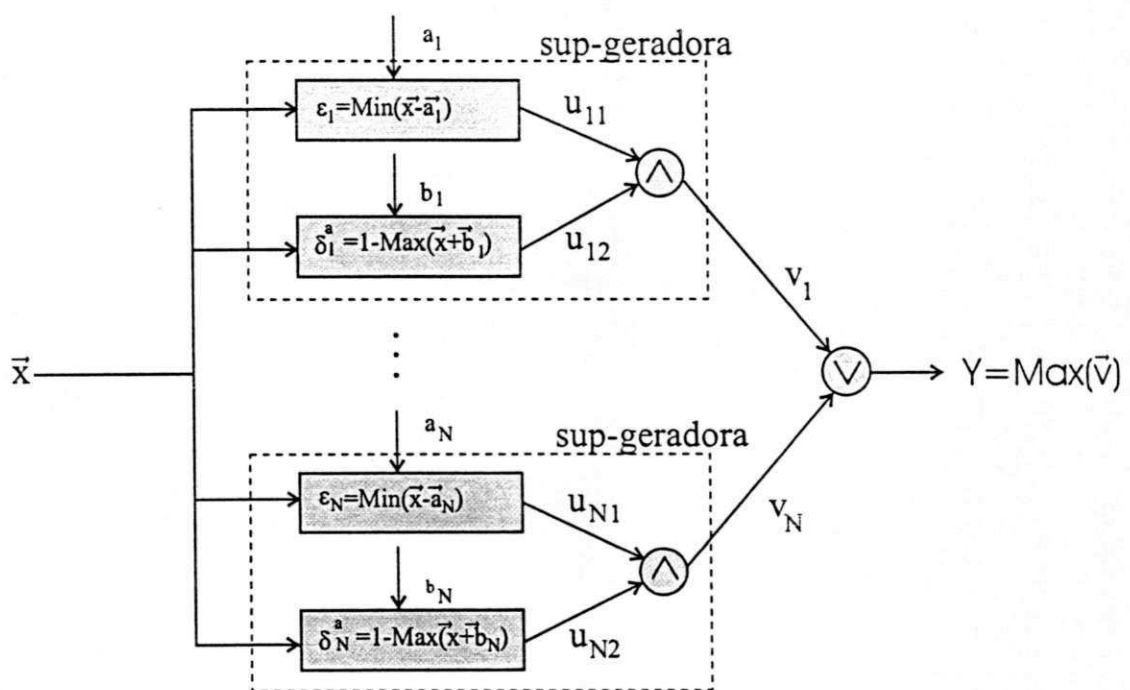


Figura 2.3: Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via sup-geradores.

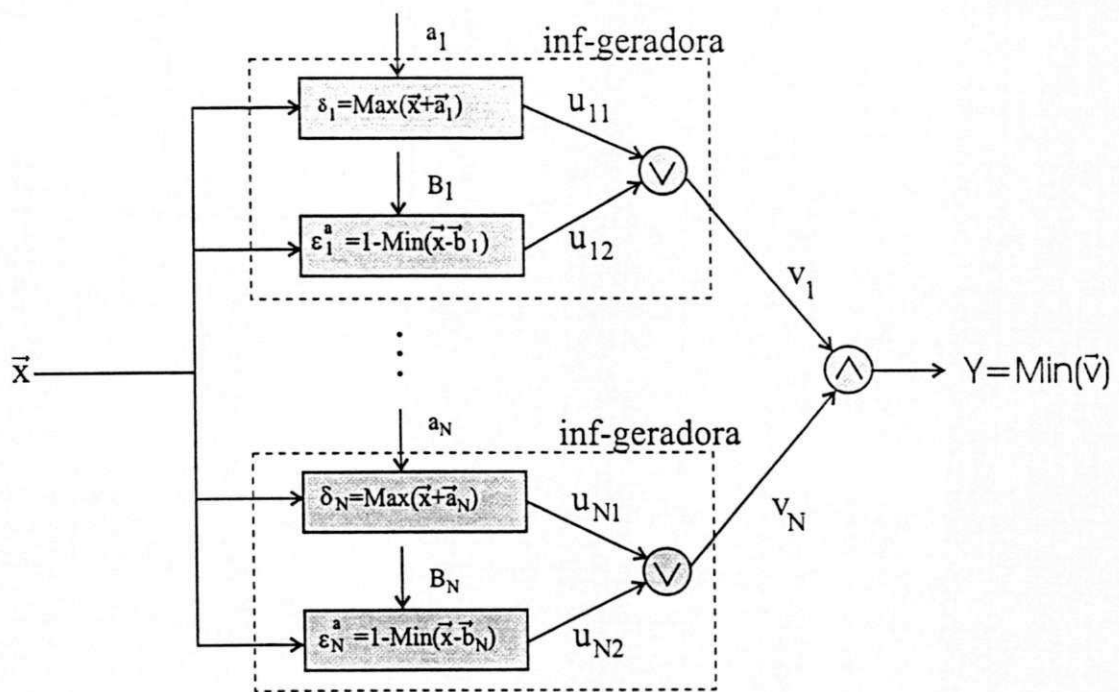


Figura 2.4: Rede Neural Modular Morfológica, arquitetura satisfazendo a decomposição do operador via inf-geradores.

No caso da decomposição do operador via sup-geradores (equação 2.12), a saída de cada módulo da rede (figura 2.3) irá convergir para uma sub-unidade integradora que fará uma operação de intersecção (mínimo) entre uma erosão e uma anti-dilatação. A saída de cada sub-unidade integradora por sua vez, irá convergir para a unidade integradora total, que fará uma operação de união (máximo) de todas as sub-unidades integradoras. De forma dual para a decomposição do operador via inf-geradora (equação 2.13), as operações de intersecção (mínimo) são substituídas por operações de união(máximo), e a operação de união é substituída por uma operação de intersecção (mínimo), como ilustrada na figura 2.4. Os vetores \vec{a}_k representam elementos estruturantes das operações de erosão e dilatação, enquanto que, os vetores \vec{b}_k representam a reflexão do complemento dos elementos estruturantes das operações de anti-dilatação e anti-erosão.

2.3.1 Fundamentos da RNMM para a decomposição de Banon e Barrera

As seguintes equações definem a arquitetura da RNMM ilustrada na figura 2.3, para a decomposição de Banon e Barrera [8] via sup-geradores. As equações que compõem a arquitetura da RNMM são analiticamente definidas da seguinte forma:

$$\text{Erosão : } \varepsilon_k = u_{k1} = \text{Min}(\vec{x} - \vec{a}_k) \quad (2.16)$$

$$\text{Anti-Dilatação : } \delta_k^a = u_{k2} = 1 - \text{Max}(\vec{x} + \vec{b}_k) \quad (2.17)$$

$$\text{Sup-Gerador : } v_k = \text{Min}(\vec{u}_k), k = 1, 2, \dots, N, \quad (2.18)$$

em que,

$$\vec{u}_k = (u_{k1}, u_{k2}), k = 1, 2, \dots, N \quad (2.19)$$

$$\text{Saída da RNMM : } y = \text{Max}(\vec{v}) \quad (2.20)$$

em que,

$$\vec{v} = (v_1, v_2, \dots, v_N) \quad (2.21)$$

As matrizes pesos A e B da RNMM são definidas por,

$$A = \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \cdot \\ \cdot \\ \vec{a}_N \end{pmatrix}, B = \begin{pmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \cdot \\ \cdot \\ \vec{b}_N \end{pmatrix} \quad (2.22)$$

Nas equações 2.16 e 2.17, \vec{x} é o sinal de entrada da RNMM, \vec{a}_k e \vec{b}_k (\vec{b}_k representa a reflexão do complemento dos elementos estruturantes da anti-dilatação) são os pesos quaisquer da RNMM, com $k = 1, 2, \dots, N$. Os símbolos \wedge e \vee representam operações de mínimo e máximo respectivamente.

De forma dual, de acordo com a figura 2.4, define-se uma arquitetura da RNMM para a decomposição via inf-geradores. Para isso é suficiente trocar as operações de erosão por dilatação e anti-dilatação por anti-erosão. Nas unidades sub-integradoras, a operação será de máximo e na unidade integradora geral (saída geral da RNMM) a operação será de mínimo, ficando pois as equações da rede da seguinte forma.

$$\text{Dilatação : } \delta_k = u_{k1} = \text{Max}(\bar{x} + \bar{a}_k) \quad (2.23)$$

$$\text{Anti-Erosão : } \varepsilon_k^a = u_{k2} = 1 - \text{Min}(\bar{x} - \bar{b}_k) \quad (2.24)$$

$$\text{Inf-Gerador : } v_k = \text{Max}(\bar{u}_k), k = 1, 2, \dots, N, \quad (2.25)$$

em que,

$$\bar{u}_k = (u_{k1}, u_{k2}), k = 1, 2, \dots, N \quad (2.26)$$

$$\text{Saída da RNMM : } y = \text{Min}(\bar{v}), \quad (2.27)$$

em que,

$$\bar{v} = (v_1, v_2, \dots, v_N), \quad (2.28)$$

Na equação 2.24 \bar{b}_k representa a reflexão do complemento dos elementos estruturantes da anti-erosão.

É importante observar que ε_k , δ_k , δ_k^a e ε_k^a são medidas, isto é, mapeamentos do conjunto de subimagens definidos numa janela para uma escala de cinzas [30].

De acordo com o exposto acima as arquiteturas da RNMM para a decomposição de operadores invariantes à translação via sup-geradores e inf-geradores satisfazem as condições do teorema da decomposição de Banon e Barrera [8].

2.3.2 Fundamentos da RNMM para decomposição de Matheron

A figura 2.1 apresenta a arquitetura da RNMM para decomposição do operador de acordo com a equação 2.14, para a decomposição de Matheron [2] via erosões.. As seguintes equações definem esta arquitetura:

$$\text{Erosão : } \varepsilon_k = v_k = \text{Min}(\bar{x} - \bar{a}_k) \quad (2.29)$$

$$\text{Saída da RNMM : } y = \text{Max}(\bar{v}) \quad (2.30)$$

em que,

$$\bar{v} = (v_1, v_2, \dots, v_N) \quad (2.31)$$

A matriz peso A da RNMM é definida por.

$$A = \begin{pmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \cdot \\ \cdot \\ \bar{a}_N \end{pmatrix} \quad (2.32)$$

Na equação 2.29 \bar{x} é o sinal de entrada da RNMM e $\bar{a}_k \in R^N$, $k = 1, 2, \dots, N$, é o elemento estruturante. Nesta tese \bar{a}_k é referido como peso da rede. O símbolo \vee representa a operação de máximo.

De forma dual, de acordo com a equação 2.15, define-se a arquitetura da RNMM para decomposição de Matheron via dilatações. Para isto basta trocar as erosões por

dilatações e \vee por \wedge , em que \wedge representa a operação de mínimo, veja a figura 2.2. As seguintes equações definem a arquitetura da RNMM para o caso da decomposição de Matheron [2] via dilatações:

$$\text{Dilatação : } \delta_k = v_k = \text{Max}(\vec{x} + \vec{a}_k) \quad (2.33)$$

$$\text{Saída da RNMM : } y = \text{Min}(\vec{v}) \quad (2.34)$$

em que,

$$\vec{v} = (v_1, v_2, \dots, v_N) \quad (2.35)$$

As arquiteturas da RNMM das figuras 2.1, 2.2 satisfazem as condições da decomposição de Matheron.

2.4 Considerações Finais

As equações da rede RNMM apresentadas neste capítulo podem ser utilizadas indistintamente para imagens em níveis de cinza e para imagens binárias. No caso da utilização com imagens em níveis de cinza, basta normalizar os valores da imagem no intervalo $[0, 1]$. Com imagens binárias, a seguinte transformação deve ser feita com as matrizes pesos A e B : $0 \Rightarrow -\infty$ e $1 \Rightarrow 0$, isto é, mapeia-se o 0 (zero) como menos infinito e o 1 (um) como zero. Esta transformação é necessária devido à natureza das equações que compõem a RNMM. Observe que todas as operações são localmente definidas. Observe também que os elementos estruturantes em níveis de cinza podem ser gerados dentro do intervalo $[-1, 1]$.

Neste capítulo ressaltamos dois aspectos fundamentais:

a) a definição e fundamentação da RNMM, satisfazendo as condições do teorema da decomposição de Matheron [2], para operadores do tipo crescente, bem como a

fundamentação da RNMM satisfazendo as condições do teorema da decomposição de Banon e Barrera [8], para operadores não necessariamente crescentes.

b) As equações que compõem a RNMM foram definidas analiticamente.

No próximo capítulo, desenvolveremos uma regra de treinamento original via algoritmo de retro-propagação do erro (back-propagation) para a RNMM, que resolve o problema proposto nesta tese.

Capítulo 3

Treinamento da RNMM Via Retro-Propagação do Erro

Neste capítulo é apresentado o treinamento da RNMMn baseado no algoritmo do erro propagado [12], tanto para o caso da decomposição de Banon/Barrera como para o caso da decomposição de Matheron . O treinamento da RNMM via erro propagado toma como ponto de partida o trabalho de Pessoa e Maragos [20]. O qual apresentam uma classe geral de filtros híbridos denominados filtros morfológicos/posto/linear (MPL) com planejamento adaptativo ótimo. O filtro MPL consiste de uma combinação linear entre um filtro morfológico/posto e um filtro linear, cuja componente não linear é baseada na função posto. Os operadores morfológicos de erosão e dilatação podem ser obtidos como casos especiais da função posto. Em [21], Pessoa apresenta uma classe de rede neural bastante geral a Rede Neural MPL (RN-MPL) cuja unidade fundamental (nó) é o Filtro MPL. A RN-MPL unifica o perceptron multiaprendizado com redes neurais morfológicas/posto [22].

3.1 Treinamento da RNMM para a decomposição de Banon e Barrera

O treinamento da RNMM é supervisionado, sob o critério do erro médio quadrático, tomando como base as idéias do algoritmo do erro propagado [17]. O objetivo principal é determinar um conjunto de matrizes peso A e B de tal forma que a função custo $J(A, B)$ seja minimizada.

Seja então o par

$$\{\bar{x}(m), d(m)\}, m = 1, \dots, M \quad (3.1)$$

o conjunto de treinamento da RNMM, em que, \bar{x} é o vetor de entrada e $d(m)$ é a resposta desejada da rede.

A função custo a ser minimizada no treinamento é definida por

$$J(A, B) = \frac{1}{2}e(m)^2, \quad (3.2)$$

em que $e(m) = (d(m) - y(m))$, $m = 1, 2, \dots, M$ é o sinal erro instantâneo, ou seja, diferença entre saída desejada d e a saída da rede y .

Queremos determinar A e B de tal forma que a função custo seja minimizada, isto é,

$$(A^*, B^*) = ArgMin[J(A, B)] \quad (3.3)$$

nos quais, A^* , $(B)^*$ são os pesos ótimos da rede.

Como os módulos da RNMM são independentes, de acordo com as figuras 2.3 e 2.4, o treinamento da rede sob o ponto de vista supervisionado é dado por

$$A(n+1) = A(n) - \mu \nabla_A J(A, B), n = 1, 2, 3, \dots \quad (3.4)$$

$$B(n+1) = B(n) - \mu \nabla_B J(A, B), \quad n = 1, 2, 3, \dots \quad (3.5)$$

Em que,

A → matriz de elementos estruturantes.

B → matriz com a reflexão do complemento dos elementos estruturantes.

$0 < \mu < 1$ → é o parâmetro de convergência fixo da RNMM.

n → representa o n -ésimo padrão de treinamento apresentado a rede.

∇ → representa os gradientes, dados por:

$$\nabla_A J(A, B) = \begin{pmatrix} \frac{\partial J}{\partial \bar{a}_1} \\ \frac{\partial J}{\partial \bar{a}_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial J}{\partial \bar{a}_N} \end{pmatrix} \quad (3.6)$$

$$\nabla_B J(A, B) = \begin{pmatrix} \frac{\partial J}{\partial \bar{b}_1} \\ \frac{\partial J}{\partial \bar{b}_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial J}{\partial \bar{b}_N} \end{pmatrix} \quad (3.7)$$

Tomando como base as idéias do algoritmo retro-propagações do erro [17] temos,

$$\frac{\partial J}{\partial \bar{a}_k} = -e(m) \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial u_{k1}} \frac{\partial u_{k1}}{\partial \bar{a}_k} \quad (3.8)$$

$$\frac{\partial J}{\partial \bar{b}_k} = -e(m) \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial u_{k2}} \frac{\partial u_{k2}}{\partial \bar{b}_k} \quad (3.9)$$

com $k = 1, 2, \dots, N$ e $m = 1, 2, \dots, M$

Surge agora um problema: como determinar as derivadas acima, uma vez que as funções de máximo e de mínimo não são diferenciáveis do ponto de vista usual [31]? Pessoa e Maragos [20] apresentam uma alternativa para estimar derivadas de funções não diferenciáveis utilizando vetores indicadores de posto e a função impulso suave. Estas funções serão utilizadas para resolver o problema acima. Para o entendimento das mesmas serão necessárias algumas definições e proposições apresentadas a seguir (as provas estão demonstradas em [20]).

Definição (Função Posto) : Dado um vetor $\vec{t} = (t_1, t_2, \dots, t_n) \in R^n$, seja $t_{(1)} \geq t_{(2)} \geq \dots \geq t_{(n)}$ os componentes do vetor \vec{t} ordenados em ordem decrescente, escolha o r -ésimo elemento da lista ordenada. Define-se a r -ésima função posto de $\vec{t} = (t_1, t_2, \dots, t_n)$ com sendo,

$$R_r(\vec{t}) \triangleq t_{(r)}, \quad r = 1, 2, \dots, n \quad (3.10)$$

Definição (Função Amostra Unitária) : A função amostra unitária $q(w)$, $w \in R$ é definida por,

$$q(w) \triangleq \begin{cases} 1, & \text{para } w = 0 \\ 0, & \text{caso contrário.} \end{cases} \quad (3.11)$$

Aplicando q a todos os componentes de um vetor $\vec{w} \in R^n$, é produzido um vetor de funções amostra unitária, dado por

$$Q(\vec{w}) \equiv (q(w_1), q(w_2), \dots, q(w_n)) \quad (3.12)$$

Definição (Vetor Indicador de Posto) : Dado um vetor $\vec{t} = (t_1, t_2, \dots, t_n) \in R^n$, e um posto $r \in \{1, 2, \dots, n\}$, o r -ésimo vetor indicador de posto \vec{c} de \vec{t} é definido por

$$\bar{c}(\vec{t}, r) = \frac{Q(R_r(\vec{t}) \cdot \vec{1} - \vec{t})}{Q(R_r(\vec{t}) \cdot \vec{1} - \vec{t}) \cdot \vec{1}^T} \quad (3.13)$$

em que $\vec{1} = (1, 1, \dots, 1)$. O vetor indicador de posto indica a localização do posto $R_r(\vec{t})$ em \vec{t} .

Proposição (Representação via Produto Interno) : Seja $\vec{t} \in R^n$, $r \in \{1, 2, \dots, n\}$, e $c = \bar{c}(\vec{t}, r)$. Então,

$$\bar{c} \cdot \vec{t}^T = R_r(\vec{t}). \quad (3.14)$$

A proposição mostra uma maneira de calcular o r -ésimo posto do vetor \vec{t} .

Para um melhor entendimento vamos mostrar um exemplo de como o vetor indicador de posto \bar{c} funciona.

Exemplo: Seja $\vec{t} = (5, 7, 8, 0, 1, 3, 5, 2, 2)$, vamos determinar o quinto vetor indicador de posto $\bar{c}(\vec{t}, r) = \bar{c}((5, 7, 8, 0, 1, 3, 5, 2, 2), 5)$.

1. Ordene os componentes de \vec{t} em ordem decrescente, isto é, $8, 7, 5, 5, 3, 2, 2, 1, 0$
2. $R_r(\vec{t}) = R_5((5, 7, 8, 0, 1, 3, 5, 2, 2)) = 3$
- 3.

$$\begin{aligned} \bar{c}(\vec{t}, 5) &= \frac{Q(R_r(\vec{t}) \cdot \vec{1} - \vec{t})}{Q(R_r(\vec{t}) \cdot \vec{1} - \vec{t}) \cdot \vec{1}^T} \\ &= \frac{Q((3 \cdot \vec{1}) - (5, 7, 8, 0, 1, 3, 5, 2, 2))}{Q((3 \cdot \vec{1}) - (5, 7, 8, 0, 1, 3, 5, 2, 2)) \cdot \vec{1}^T} \\ &= \frac{Q(-2, -4, -5, 3, 2, 0, -2, 1, 1)}{Q(-2, -4, -5, 3, 2, 0, -2, 1, 1) \cdot \vec{1}^T} \\ &= \frac{(0, 0, 0, 0, 0, 1, 0, 0, 0)}{1} \end{aligned}$$

4. Verifique agora a proposição (Representação via Produto Interno), isto é,

$$\vec{c} \cdot \vec{t}^T = R_r(\vec{t})$$

$$\begin{aligned}\vec{c}(\vec{t}, 5) \cdot \vec{t}^T &= (0, 0, 0, 0, 0, 1, 0, 0, 0) \cdot (5, 7, 8, 0, 1, 3, 5, 2, 2)^T = 3 \\ \vec{c}(\vec{t}, 5) \cdot \vec{t}^T &= R_r(\vec{t}) = 3\end{aligned}$$

Proposição : Seja $\vec{c} = \vec{c}(\vec{t}, r)$, para um determinado r fixo. Se \vec{c} é constante numa vizinhança de algum \vec{t}_0 , então a r -ésima função posto $R_r(\vec{t})$ é diferenciável em \vec{t}_0 e

$$\left. \frac{\partial R_r(\vec{t})}{\partial \vec{t}} \right|_{\vec{t}=\vec{t}_0} = \vec{c}(\vec{t}_0, r). \quad (3.15)$$

Nos pontos \vec{t}_0 em cuja a vizinhança o vetor \vec{c} não é constante a função posto não é diferenciável.

Dem. De acordo com a proposição (Representação via Produto Interno) temos,

$$\frac{\partial R_r(\vec{t})}{\partial \vec{t}} = \frac{\partial \vec{c}(\vec{t}, r) \cdot \vec{t}^T}{\partial \vec{t}},$$

como \vec{c} por hipótese é constante numa vizinhança de \vec{t}_0 implica que

$$\left. \frac{\partial R_r(\vec{t})}{\partial \vec{t}} \right|_{\vec{t}=\vec{t}_0} = \left. \frac{\partial \vec{c}(\vec{t}_0, r) \cdot \vec{t}^T}{\partial \vec{t}} \right|_{\vec{t}=\vec{t}_0} = \vec{c}(\vec{t}_0, r).$$

É importante notar que a proposição nos dá uma maneira de calcular a derivada da função posto $R_r(\vec{t})$, desde que \vec{c} seja constante numa vizinhança de \vec{t}_0 .

A função amostra unitária q produz transformações abruptas. Uma outra maneira de definir a função q é a utilização da função impulso suave q_σ que depende de um parâmetro de escala σ , satisfazendo as seguintes condições:

1. $q_\sigma(w) = q_\sigma(-w)$
2. $\lim_{\sigma \rightarrow 0} q_\sigma(w) = q(w)$
3. $\lim_{\sigma \rightarrow \infty} q_\sigma(w) = 1$

Um bom candidato para q_σ é a função gaussiana $\exp[-\frac{1}{2}(\frac{w}{\sigma})^2]$. Facilmente se verifica que esta função satisfaz as três condições acima. O vetor suave da função amostra unitária é dado por:

$$Q_\sigma(\vec{w}) \equiv (q_\sigma(w_1), q_\sigma(w_2), \dots, q_\sigma(w_n)) \quad (3.16)$$

Definição (Função Posto Suave) : Se q_σ é um impulso suave satisfazendo as condições acima, então a r -ésima função posto é definida como:

$$R_{r,\sigma}(\vec{t}) \equiv \vec{c}_\sigma \vec{t}^T \quad (3.17)$$

em que

$$\vec{c}_\sigma(\vec{t}, r) \equiv \frac{Q_\sigma(R_r(\vec{t}) \cdot \vec{1} - \vec{t})}{Q_\sigma(R_r(\vec{t}) \cdot \vec{1} - \vec{t}) \cdot \vec{1}^T} \quad (3.18)$$

é o vetor indicador de posto suave.

É importante notar que $\vec{c}_\sigma \rightarrow \vec{c}$ e $R_{r,\sigma}(\vec{t}) \rightarrow R_r(\vec{t})$ quando $\sigma \rightarrow 0$.

Nos pontos em que a função posto não é diferenciável, o vetor indicador de posto suave é utilizado para *estimar* os gradientes durante o treinamento da RNMM. Dessa forma, para os casos particulares em que $r = 1$ (dilatação) e $r = n$ (erosão), usamos $\vec{c}_\sigma(\vec{t}, 1)$ e $\vec{c}_\sigma(\vec{t}, n)$ no cálculo dos gradientes.

3.1.1 Determinação das derivadas das equações de treinamento da RNMM, decomposição Banon e Barrera.

Nesta subsecção serão determinadas as derivadas das equações 3.8 e 3.9 de acordo com a metodologia de Pessoa e Maragos [20].

a) Determinando a derivada $\frac{\partial y}{\partial \vec{v}}$.

$$\frac{\partial y}{\partial \vec{v}} = \frac{\partial \text{Max}(\vec{v})}{\partial \vec{v}} = \vec{c}$$

em que,

$$\vec{c} = \frac{Q_{\sigma}(\text{Max}(\vec{v}) \cdot \vec{1} - \vec{v})}{Q_{\sigma}(\text{Max}(\vec{v}) \cdot \vec{1} - \vec{v}) \cdot \vec{1}^T} \quad (3.19)$$

b) Determinando a derivada $\frac{\partial v_k}{\partial \vec{u}_k}$,

$$\frac{\partial v_k}{\partial \vec{u}_k} = \frac{\partial \text{Min}(\vec{u}_k)}{\partial \vec{u}_k}$$

para o qual,

$$\frac{\partial \text{Min}(\vec{u}_k)}{\partial \vec{u}_k} = \vec{c}_k^{\rightarrow} = (\hat{c}_{k1}, \hat{c}_{k2}) \quad (3.20)$$

$$= \frac{Q_{\sigma}(\text{Min}(\vec{u}_k) \cdot \vec{1} - \vec{u}_k)}{Q_{\sigma}(\text{Min}(\vec{u}_k) \cdot \vec{1} - \vec{u}_k) \cdot \vec{1}^T} \quad (3.21)$$

c) Determinando a derivada $\frac{\partial u_{k1}}{\partial \vec{a}_k}$.

$$\frac{\partial u_{k1}}{\partial \vec{a}_k} = \frac{\partial \text{Min}(\vec{x} - \vec{a}_k)}{\partial \vec{a}_k} = \vec{c}_{k1}$$

em que,

$$\bar{c}_{k1} = -\frac{Q_{\sigma}(\text{Min}(\bar{x} - \bar{a}_k) \cdot \bar{\mathbf{I}} - \bar{x} + \bar{a}_k)}{Q_{\sigma}(\text{Min}(\bar{x} - \bar{a}_k) \cdot \bar{\mathbf{I}} - \bar{x} + \bar{a}_k) \cdot \bar{\mathbf{I}}^T} \quad (3.22)$$

d) Determinando a derivada $\frac{\partial u_{k2}}{\partial \bar{b}_k}$

$$\frac{\partial u_{k2}}{\partial \bar{b}_k} = \frac{\partial(1 - \text{Max}(\bar{x} + \bar{b}_k))}{\partial \bar{b}_k} = -\frac{\partial(\text{Max}(\bar{x} + \bar{b}_k))}{\partial \bar{b}_k} = \bar{c}_{k2}$$

em que,

$$\bar{c}_{k2} = -\frac{Q_{\sigma}(\text{Max}(\bar{x} + \bar{b}_k) \cdot \bar{\mathbf{I}} - \bar{x} - \bar{b}_k)}{Q_{\sigma}(\text{Max}(\bar{x} + \bar{b}_k) \cdot \bar{\mathbf{I}} - \bar{x} - \bar{b}_k) \cdot \bar{\mathbf{I}}^T}$$

Mas,

$$1 - u_{k2} = \text{Max}(\bar{x} + \bar{b}_k)$$

resultando em,

$$\frac{\partial u_{k2}}{\partial \bar{b}_k} = \bar{c}_{k2} = -\frac{Q_{\sigma}((1 - u_{k2}) \cdot \bar{\mathbf{I}} - \bar{x} - \bar{b}_k)}{Q_{\sigma}((1 - u_{k2}) \cdot \bar{\mathbf{I}} - \bar{x} - \bar{b}_k) \cdot \bar{\mathbf{I}}^T} \quad (3.23)$$

Em termos matriciais, os gradientes das equações 3.8 e 3.9 ficam então,

$$\nabla_A J(A, B) = -e \cdot \text{diag}(\bar{c}) \cdot \text{diag}(\bar{c}_1) \cdot C_1 \quad (3.24)$$

$$\nabla_B J(A, B) = -e \cdot \text{diag}(\bar{c}) \cdot \text{diag}(\bar{c}_2) \cdot C_2 \quad (3.25)$$

Nas equações 3.24 e 3.25, $\text{diag}(\bar{c}_1)$ e $\text{diag}(\bar{c}_2)$ são matrizes diagonais cujos elementos da diagonal principal são representados pelos vetores c , \bar{c}_1 e \bar{c}_2 respectivamente; as

matrizes C_1 e C_2 são obtidas pelas colunas dos vetores \bar{c}_{k1} e \bar{c}_{k2} , tendo dimensões iguais ao número de sup-geradores (linhas) pelo número de pontos no elemento estruturante (colunas).

As equações 3.24 e 3.25 são as equações finais de treinamento da RNMM, definidas de acordo com as equações 3.19, 3.20, 3.22 e 3.23. Os vetores \bar{c}_1 , \bar{c}_2 , C_1 e C_2 são dados por,

1. $\bar{c}_1 = (\hat{c}_{11}, \hat{c}_{21}, \dots, \hat{c}_{N1})$.
2. $\bar{c}_2 = (\hat{c}_{12}, \hat{c}_{22}, \dots, \hat{c}_{N2})$.
3. $C_1 = (\bar{c}_{11}^T, \bar{c}_{21}^T, \dots, \bar{c}_{N1}^T)^T$.
4. $C_2 = (\bar{c}_{12}^T, \bar{c}_{22}^T, \dots, \bar{c}_{N2}^T)^T$.

De forma dual determina-se a regra de treinamento para a RNMM (decomposição via inf-geradores) como ilustrada na figura 2.4 . As derivadas das equações 3.8 e 3.9, são obtidas de forma semelhante, via metodologia de Pessoa e Maragos [20], resultando em:

$$\frac{\partial y}{\partial \bar{v}} = \frac{Q_\sigma(\text{Min}(\bar{v}) \cdot \bar{1} - \bar{v})}{Q_\sigma(\text{Min}(\bar{v}) \cdot \bar{1} - \bar{v}) \cdot \bar{1}^T} = \bar{c}, y = \text{Min}(\bar{v}) \quad (3.26)$$

$$\frac{\partial v_k}{\partial \bar{u}_k} = \frac{Q_\sigma(\text{Max}(\bar{u}_k) \cdot \bar{1} - \bar{u}_k)}{Q_\sigma(\text{Max}(\bar{u}_k) \cdot \bar{1} - \bar{u}_k) \cdot \bar{1}^T} = \hat{c}_k^{\rightarrow} = (\hat{c}_{k1}, \hat{c}_{k2}), v_k = \text{Max}(\bar{u}_k) \quad (3.27)$$

$$\frac{\partial u_{k1}}{\partial \bar{a}_k} = \frac{Q_\sigma(\text{Max}(\bar{x} + \bar{a}_k) \cdot \bar{1} - \bar{x} - \bar{a}_k)}{Q_\sigma(\text{Max}(\bar{x} + \bar{a}_k) \cdot \bar{1} - \bar{x} - \bar{a}_k) \cdot \bar{1}^T} = \bar{c}_{k1}, u_{k1} = \text{Max}(\bar{x} + \bar{a}_k) \quad (3.28)$$

$$\frac{\partial u_{k1}}{\partial \bar{a}_k} = \frac{Q_\sigma(u_{k1} \cdot \bar{1} - \bar{x} - \bar{a}_k)}{Q_\sigma(u_{k1} \cdot \bar{1} - \bar{x} - \bar{a}_k) \cdot \bar{1}^T} = \bar{c}_{k1} \quad (3.29)$$

$$\frac{\partial u_{k2}}{\partial \vec{b}_k} = \frac{Q_\sigma((1 - u_{k2}) \cdot \vec{1} - \vec{x} + \vec{b}_k)}{Q_\sigma((1 - u_{k2}) \cdot \vec{1} - \vec{x} + \vec{b}_k) \cdot \vec{1}^t} = \vec{c}_{k2}, \quad u_{k2} = 1 - \text{Min}(\vec{x} - \vec{b}_k) \quad (3.30)$$

As regras de treinamento apresentadas acima são para as redes das figuras 2.3 e 2.4, porém podem ser também aplicadas para a rede da figura 3.1, que é uma RNMM com múltiplas saídas. O treinamento da RNMM com múltiplas saídas é mais complexo, pois cada saída da mesma requer um processamento equivalente ao treinamento da RNMM das figuras 2.3 e 2.4. Este tipo de arquitetura pode ser diretamente aplicado a problemas de reconhecimento de padrões com multiclass.

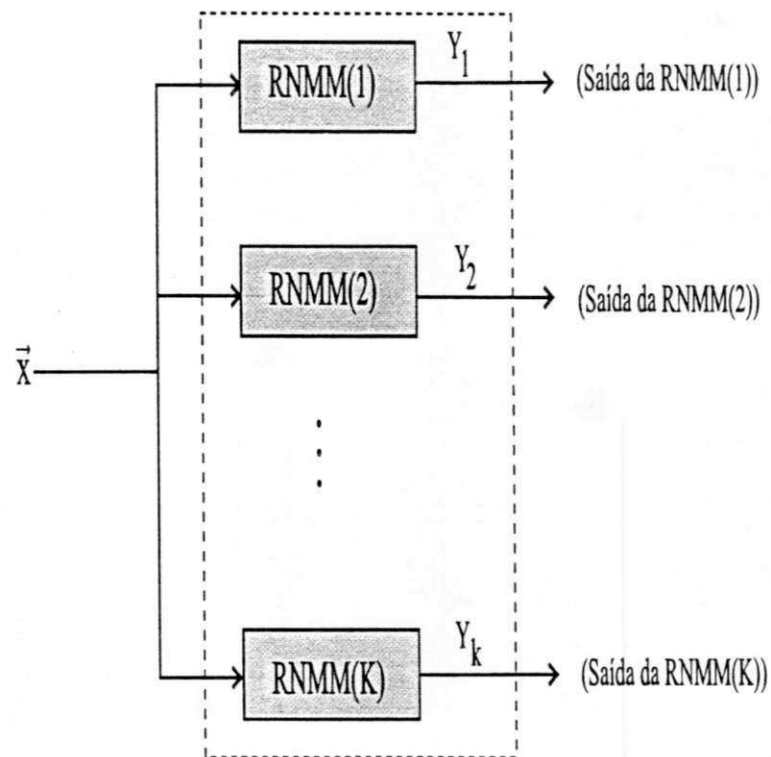


Figura 3.1: RNMM com múltiplas saídas. Observe que cada módulo contém uma RNMM.

De forma geral, o treinamento da RNMM com múltiplas saídas é feito de forma

idêntica ao treinamento da RNMM vista nos casos anteriores. Como os módulos são independentes um do outro, as matrizes gradiente são obtidas da seguinte forma

$$\nabla_A J(A, B) = \text{diag}(\nabla_{A_1}, \nabla_{A_2}, \dots, \nabla_{A_K}) \quad (3.31)$$

$$\nabla_B J(A, B) = \text{diag}(\nabla_{B_1}, \nabla_{B_2}, \dots, \nabla_{B_K}) \quad (3.32)$$

em que

$$A = \text{diag}(A_1, A_2, \dots, A_K)$$

$$B = \text{diag}(B_1, B_1, \dots, B_K)$$

(A_K, B_K) , são os pesos da RNMM(k), $k = 1, 2, \dots, K$

Observe que a RNMM com múltiplas saídas comporta RMNNs de dimensões diferentes.

Apresentaremos agora o algoritmo de treinamento da RNMM tanto para a arquitetura da figura 2.3 como para a arquitetura da figura 2.4.

3.1.2 Algoritmo de treinamento da RNMM para decomposição via sup-geradores ou inf-geradores

Entre com um conjunto de pares de treinamento $\{\vec{x}(m), d(m)\}$, $m = 1, \dots, M$.

1. Inicialização:

- Defina o número de sup-geradores, $K = 1, 2, \dots, N$.
- Defina o valor de μ , $\mu > 0$.
- Defina o número máximo de épocas.
- Defina o erro mínimo tolerável E_{Min} .
- Inicialize as matrizes pesos (elementos estruturantes) A e B com valores aleatórios uniformemente distribuídos entre $[-1, 1]$.

2. Evoluindo:

- Apresente para a entrada da rede $\vec{x}(m)$, $m = 1, 2, \dots, M$.
- Calcule

$$y = \text{Max}(\vec{v}), \text{ para sup-geradores}$$

$$y = \text{Min}(\vec{v}), \text{ para inf-geradores}$$

- Calcule

$$e(m) = d(m) - y(m), m = 1, 2, \dots, M$$

3. Retropropagando:

- Calcule os gradientes locais utilizando as equações 3.8 e 3.9

$$\nabla_A J(A, B) : \frac{\partial J}{\partial \vec{a}_k} = -e(m) \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial u_{k1}} \frac{\partial u_{k1}}{\partial \vec{a}_k}, k = 1, 2, \dots, N$$

$$\nabla_B J(A, B) : \frac{\partial J}{\partial \vec{b}_k} = -e(m) \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial u_{k2}} \frac{\partial u_{k2}}{\partial \vec{b}_k}, k = 1, 2, \dots, N$$

- Em termos matriciais, os gradientes locais são calculados utilizando as equações 3.24 e 3.25.

$$\nabla_A J(A, B) = -e \cdot \text{diag}(\mathbf{c}) \cdot \text{diag}(\vec{c}_1) \cdot C_1 \quad (3.33)$$

$$\nabla_B J(A, B) = -e \cdot \text{diag}(\mathbf{c}) \cdot \text{diag}(\vec{c}_2) \cdot C_2 \quad (3.34)$$

4. Atualize os pesos da rede:

- Utilize as equações 3.4 e 3.5 para atualizar as matrizes pesos A e B

$$A(n+1) = A(n) - \mu \nabla_A J(A, B), n = 1, 2, 3, \dots$$

$$B(n+1) = B(n) - \mu \nabla_B J(A, B), n = 1, 2, 3, \dots$$

5. Repita:

- Repita os passos 2, 3 e 4 até que os pesos sejam estabilizados e a função custo seja minimizada ou, até que o número de épocas seja satisfeito.

Observe que as equações de treinamento da RNMM foram definidas com parâmetro de convergência fixo. Podem ocorrer as seguintes situações indesejáveis durante o processo de treinamento da RNMM: a convergência da RNMM é lenta ou a RNMM converge para um mínimo local. Uma forma de resolver tais problemas seria a utilização do parâmetro de convergência variável [32].

3.2 Treinamento da RNMM para a decomposição de Matheron

Como visto anteriormente, o treinamento da RNMM será supervisionado, sob o critério do erro médio quadrático, tomando como base as idéias do algoritmo do erro propagado [17]. O objetivo principal é determinar uma matriz peso A de tal forma que a função custo $J(A)$ seja minimizada. Para derivar a regra de treinamento da RNMM via decomposição de Matheron [2] para os casos da decomposição via erosões ou via dilatações (figuras 2.1 e 2.2) respectivamente, utilizaremos a metodologia adotada anteriormente.

Seja então o par

$$\{\bar{\mathbf{x}}(m), d(m)\}, m = 1, \dots, M \quad (3.35)$$

o conjunto de treinamento da RNMM, em que, $\bar{\mathbf{x}}$ é o vetor de entrada e $d(m)$ é a resposta desejada da rede.

A função custo que será utilizada no treinamento é definida por

$$J(A) = \frac{1}{2}e(m)^2, \quad (3.36)$$

$e(m) = (d(m) - y(m))$, $m = 1, 2, \dots, M$ é o sinal erro instantâneo, ou seja, a diferença entre saída desejada da rede d e a saída da rede y .

Queremos determinar A de tal forma que a função custo seja minimizada, isto é,

$$A^* = \text{ArgMin}[J(A)] \quad (3.37)$$

em que, A^* é o peso ótimo da rede.

De acordo com as figuras 2.1 e 2.2, o treinamento da rede sob o ponto de vista supervisionado é dado por

$$A(n+1) = A(n) - \mu \nabla_A J(A), n = 1, 2, 3, \dots \quad (3.38)$$

O parâmetro $\mu > 0$ é um parâmetro de convergência da RNMM, n representa o n -ésimo padrão de treinamento apresentado a rede. O gradiente ∇ é dado por:

$$\nabla_A J(A) = \begin{pmatrix} \frac{\partial J}{\partial \bar{a}_1} \\ \frac{\partial J}{\partial \bar{a}_2} \\ \cdot \\ \cdot \\ \frac{\partial J}{\partial \bar{a}_N} \end{pmatrix} \quad (3.39)$$

Tomando como base algoritmo retro-propagações do erro [17] temos,

$$\frac{\partial J}{\partial \vec{a}_k} = -e \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial \vec{a}_k}, k = 1, 2, \dots, N \quad (3.40)$$

3.2.1 Determinação das derivadas da equação de treinamento da RNMM, decomposição de Matheron

Vamos agora determinar as derivadas da equação 3.40 de acordo com as equações 2.29, 2.31 e 2.30(decomposição via erosões), segundo a metodologia de Pessoa e Maragos [20].

a) Determinando a derivada $\frac{\partial y}{\partial \vec{v}}$.

$$\frac{\partial y}{\partial \vec{v}} = \frac{\partial Max(\vec{v})}{\partial \vec{v}} = \vec{c} = \frac{Q_\sigma(Max(\vec{v}).\vec{1} - \vec{v})}{Q_\sigma(Max(\vec{v}).\vec{1} - \vec{v}).\vec{1}^T} \quad (3.41)$$

b) Determinando a derivada de $\frac{\partial v_k}{\partial a_k}$.

$$\frac{\partial v_k}{\partial \vec{a}_k} = \frac{\partial Min(\vec{x} - \vec{a}_k)}{\partial \vec{a}_k} = \vec{c}_k = -\frac{Q_\sigma(Min(\vec{x} - \vec{a}_k).\vec{1} - \vec{x} + \vec{a}_k)}{Q_\sigma(Min(\vec{x} - \vec{a}_k).\vec{1} - \vec{x} + \vec{a}_k).\vec{1}^T} \quad (3.42)$$

De forma dual, determina-se as derivadas da decomposição de Matheron via dilatações (figura 2.2). As derivadas da equação 3.40 são determinadas de acordo com as equações 2.33, 2.35 e 2.34, via metodologia de Pessoa e Maragos [20].

a) Determinando a derivada de $\frac{\partial y}{\partial \vec{v}}$.

$$\frac{\partial y}{\partial \vec{v}} = \frac{\partial Min(\vec{v})}{\partial \vec{v}} = \vec{c} = \frac{Q_\sigma(Min(\vec{v}).\vec{1} - \vec{v})}{Q_\sigma(Min(\vec{v}).\vec{1} - \vec{v}).\vec{1}^T} \quad (3.43)$$

b) Determinando a derivada de $\frac{\partial v_k}{\partial a_k}$.

$$\frac{\partial v_k}{\partial \bar{a}_k} = \frac{\partial \text{Max}(\bar{x} + \bar{a}_k)}{\partial \bar{a}_k} = \bar{c}_k = \frac{Q_\sigma(\text{Max}(\bar{x} + \bar{a}_k) \cdot \bar{1} - \bar{x} - \bar{a}_k)}{Q_\sigma(\text{Max}(\bar{x} + \bar{a}_k) \cdot \bar{1} - \bar{x} - \bar{a}_k) \cdot \bar{1}^T} \quad (3.44)$$

Em termos matriciais, a equação final de treinamento da RNMM sob o ponto de vista da decomposição de Matheron fica:

$$\nabla_A J(A) = -e \cdot \text{diag}(\bar{c}) \cdot C, \quad (3.45)$$

em que $C = (\bar{c}_1^T, \bar{c}_2^T, \dots, \bar{c}_N^T)^T$.

3.2.2 Algoritmo de treinamento da RNMM para decomposição via erosões ou dilatações

Entre com um conjunto de pares de treinamento $\{\bar{x}(m), d(m)\}$, $m = 1, \dots, M$.

1. Inicialização:

- Defina o número de sup-geradores, $K = 1, 2, \dots, N$.
- Defina o valor de μ , $\mu > 0$.
- Defina o número máximo de épocas.
- Defina o erro mínimo tolerável E_{Min} .
- Inicialize as matrizes pesos (elementos estruturantes) A com valores aleatórios uniformemente distribuídos entre $[-1, 1]$.

2. Evoluindo:

- Apresente para a entrada da rede $\bar{x}(m)$, $m = 1, 2, \dots, M$.
- Calcule

$y = \text{Max}(\vec{v})$, decomposição via erosões

$y = \text{Min}(\vec{v})$, decomposição via dilatações

- Calcule

$$e(m) = d(m) - y(m), m = 1, 2, \dots, M$$

3. Retropropagando:

- Calcule os gradientes locais utilizando a equação

$$\nabla_A J(A) : \frac{\partial J}{\partial \vec{a}_k} = -e \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial a_k}, k = 1, 2, \dots, N$$

- Em termos matriciais o gradiente local é calculado utilizando a equação

$$\nabla_A J(A) = -e \cdot \text{diag}(\vec{e}) \cdot C$$

4. Atualize os pesos da rede:

- Atualize a matriz peso A pela equação

$$A(n+1) = A(n) - \mu \nabla_A J(A), n = 1, 2, 3, \dots$$

5. Repita:

- Repita os passos 2, 3 e 4 até que os pesos sejam estabilizados e a função custo seja minimizada, ou até que o número de épocas seja satisfeito.

Na próxima seção apresentaremos uma aplicação da RNMM, na área de filtragem de imagens. O objetivo desta aplicação é projetar filtros morfológicos que suprimem ruído em imagens binárias e em níveis de cinza.

3.3 Aplicação em filtragem de imagens (RNMM via decomposição de Matheron)

Um problema clássico na área de processamento de imagens é o de filtragem de imagens corrompidas por ruído [33]. Um tipo de ruído particularmente interessante é o ruído sal e pimenta. Para suprimir este tipo de ruído, o filtro clássico é o da mediana [33], que para determinados níveis ou densidades de ruído tem se mostrado bastante eficiente. O nosso problema específico é projetar o filtro morfológico através de treinamento da RNMM via decomposição de Matheron [2], que suprime o ruído acima definido, utilizando para este filtro a matriz peso A^* fornecida pelo processo de aprendizagem da RNMM. O operador morfológico Ψ que suprime o ruído do tipo sal e pimenta consiste de uma operação de abertura seguida de uma operação de fechamento [33]. Como o método proposto é adaptativo, se espera que o desempenho final seja bem próximo do filtro ideal.

Seguindo a metodologia de Pessoa e Maragos [20], o sinal de entrada para o projeto do filtro é a imagem corrompida com ruído aleatório do tipo sal e pimenta. O sinal desejado é a imagem original sem ruído.

O conjunto de treinamento da RNMM é obtido da seguinte forma:

1. Defina uma janela quadrada L_m de tamanho $n = (2m + 1)^2$, $m = 1, 2, 3, \dots$
2. Centre L_m no pixel com coordenada (i, j) da imagem de entrada X .
3. Desloque L_m sobre X da esquerda para direita e de cima para baixo. A medida que L_m se move, transforme a região correspondente de X em vetores \vec{x} , os quais irão compor o conjunto de treinamento Ω da RNMM. Se $m = 1$, L_1 terá tamanho 9, e por sua vez \vec{x}^T também terá comprimento 9.
4. Os vetores peso (elementos estruturantes) \vec{a}_k e \vec{b}_k ambos terão dimensões iguais ao vetor \vec{x} .

Para cada pixel na posição (i, j) , $i = 1, 2, \dots, I$ e $j = 1, 2, \dots, J$ da imagem de entrada, apresentaremos à RNMM, o par de treinamento (\vec{x}, d) , sendo d a saída desejada para

o pixel na posição (i, j) . No caso de imagens em níveis de cinza, todos os pixels da mesma serão normalizados para o intervalo $[0, 1]$. Neste experimento usaremos a função amostra unitária suavizada, definida pela função gaussiana $q_\sigma(w) = \exp[-\frac{1}{2}(\frac{w}{\sigma})^2]$, com $\sigma = 0,05$. Utilizaremos a razão ruído sinal (NSR) expressa em dB em todo o experimento, definida como:

$$NSR = 10 \log_{10} \frac{\overline{(d - y)^2}}{\overline{(d)^2}}, \quad (3.46)$$

em que $\overline{(d - y)^2}$ é a energia média do erro (segundo momento do erro), e $\overline{d^2}$ é a energia média do objetivo (segundo momento do objetivo). A expressão 3.46 nos diz que a medida que $y \rightarrow d$, a $NSR \rightarrow -\infty$. Quanto mais próximos forem os valores de y e d , menor será a NSR . No caso ideal a $NSR = -\infty$. Os gráficos de convergência serão plotados em função da NSR , isto é, NSR versus número de épocas.

Para definir o número inicial de erosões ou dilatações que compõem a arquitetura da RNMM, utilizaremos a experiência de Oliveira [9], uma vez que não existe ainda um resultado teórico que defina o número de erosões ou de dilatações necessárias para representar a decomposição do operador invariante à translação Ψ . Como este problema não foi resolvido, a solução determinada via treinamento da RNMM será sub-ótima, sob este ponto de vista.

3.3.1 Simulações

Em todas as simulações realizadas foram utilizadas 25% da imagem ruidosa para treinar a RNMM (figura 3.2 (a) e (b)). Para testar o filtro projetado via treinamento da RNMM, utilizou-se a imagem total do treinamento da rede. Outras imagens em níveis de cinza também foram utilizadas para medir o desempenho do filtro.

A tabela 3.1 mostra os resultados das simulações realizadas com a RNMM (figura 2.1), tomando como ponto de partida, como mencionado anteriormente, a experiência de Oliveira [9] para definir o número de erosões iniciais na composição da arquitetura da RNMM. O número de épocas utilizado para o treinamento da rede, manteve-se constante em torno de 100 nas quatro situações testadas em que o parâmetro de aprendizagem

μ tomou os valores $\mu = 0,01; 0,1; 0,5$ e 1 . A figura 3.3 mostra o comportamento de convergência da RNMM através da *NSR*. Para os valores de $\mu = 0,01$ e $0,1$ observa-se que o comportamento da convergência foi normal. Na situação em que $\mu = 0,5$, o gráfico da *NSR* apresenta um comportamento oscilatório entre as épocas 1 e 15, ficando estável a partir da época 20. A situação mais crítica é apresentada na figura 3.3 em o parâmetro de convergência $\mu = 1$. Em nenhum momento do treinamento da rede a *NSR* ficou estável, e o gráfico mostra uma grande oscilação em todas as épocas, demonstrando que a RNMM não aprendeu, o resultado pode ser observado na figura 3.5 (d), que representa a saída final do filtro. Na tabela 3.1, verifica-se que a *NSR* final varia de acordo com os valores do parâmetro de convergência μ . A melhor situação observada para *NSR* final é de $-19,65dB$, com $\mu = 0,01$. A figura 3.6 (d), apresenta a saída final correspondente do filtro projetado com o treinamento da RNMM.

A tabela 3.2 mostra as simulações realizadas com a RNMM via decomposição do operador Ψ por dilatações, como mostrado na figura 2.2. É interessante notar que os resultados obtidos foram melhores do que os obtidos via decomposição por erosões. A figura 3.4 mostra o comportamento da convergência da rede através de *NSR*. Analisando a pior situação, que é o caso onde a *NSR* final ficou em torno de $-12,63 dB$ para $\mu = 1$, (figura 3.7 (d)), observa-se que a rede não conseguiu aprender o operador. Isto se deve ao fator de convergência μ , que impôs maior velocidade de convergência à rede levando a oscilações muito grandes. Já na melhor situação, temos a *NSR* = $-21,70dB$ para $\mu = 0,01$. A figura 3.8 (d) mostra a saída do filtro Ψ^* projetado com a solução da rede via decomposição por dilatações. Comparando os valores de *NSR* de ambas as tabelas, vemos que há uma diferença pró dilatações.

Comparando os melhores valores das tabelas 3.1 e 3.2, com o valor obtido com o filtro da mediana clássico (*NSR* = $-21,5547dB$) para uma densidade de ruído de 5% (tabela3.3), vemos que houve pequena diferença entre os filtros projetados via RNMM com 8 erosões e com 8 dilatações. Após exaustivas simulações, aumentando paulatinamente o número de erosões e dilatações que compõem a arquitetura da RNMM até 25, obteve-se *NSR* igual a $-22,42dB$ e a $-22,63dB$, respectivamente (veja figura 3.9 (d)), deste modo superando o filtro da mediana em aproximadamente $1dB$.

Testou-se o filtro projetado via treinamento RNMM com outra imagem com ruído

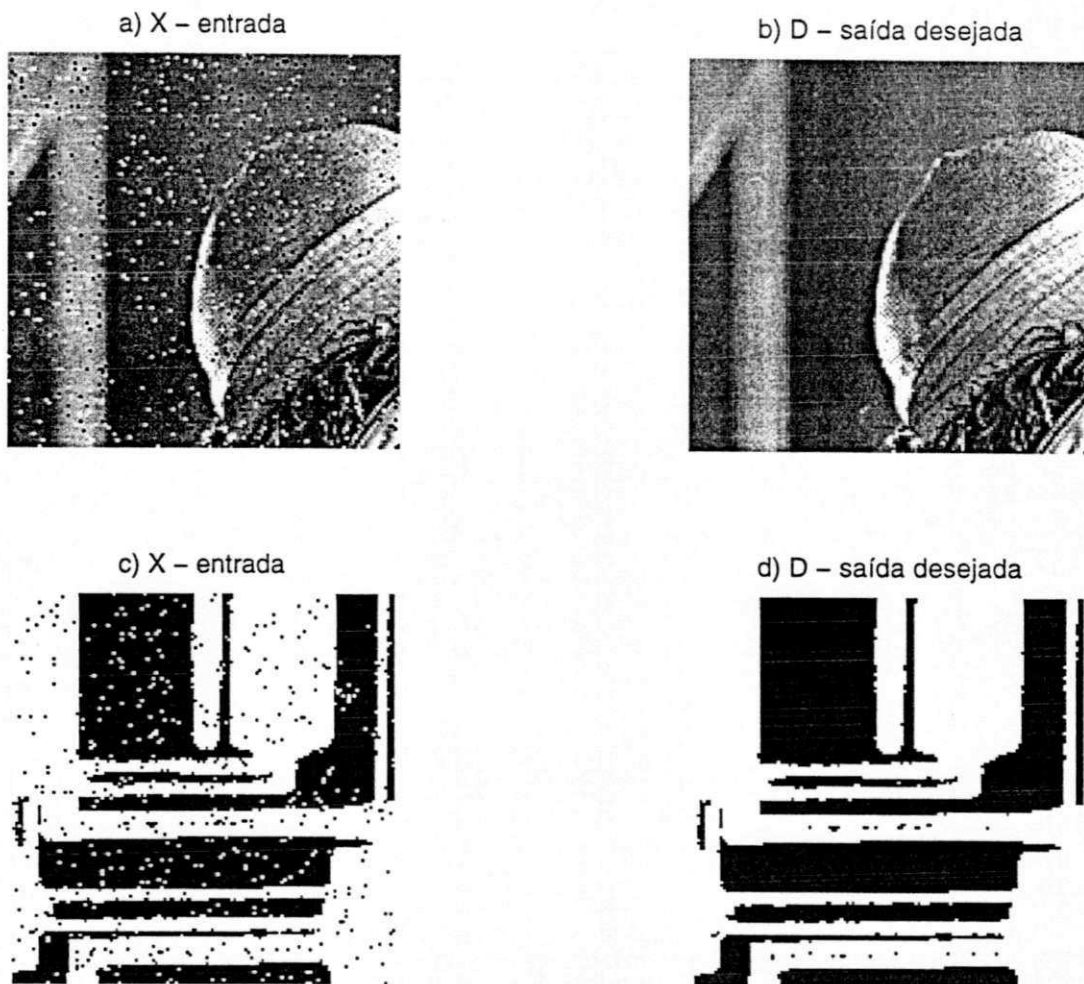


Figura 3.2: Imagens de treinamento da RNMM para filtragem de ruído. Imagens em níveis de cinza (a) e (b), imagens binárias (c) e (d)

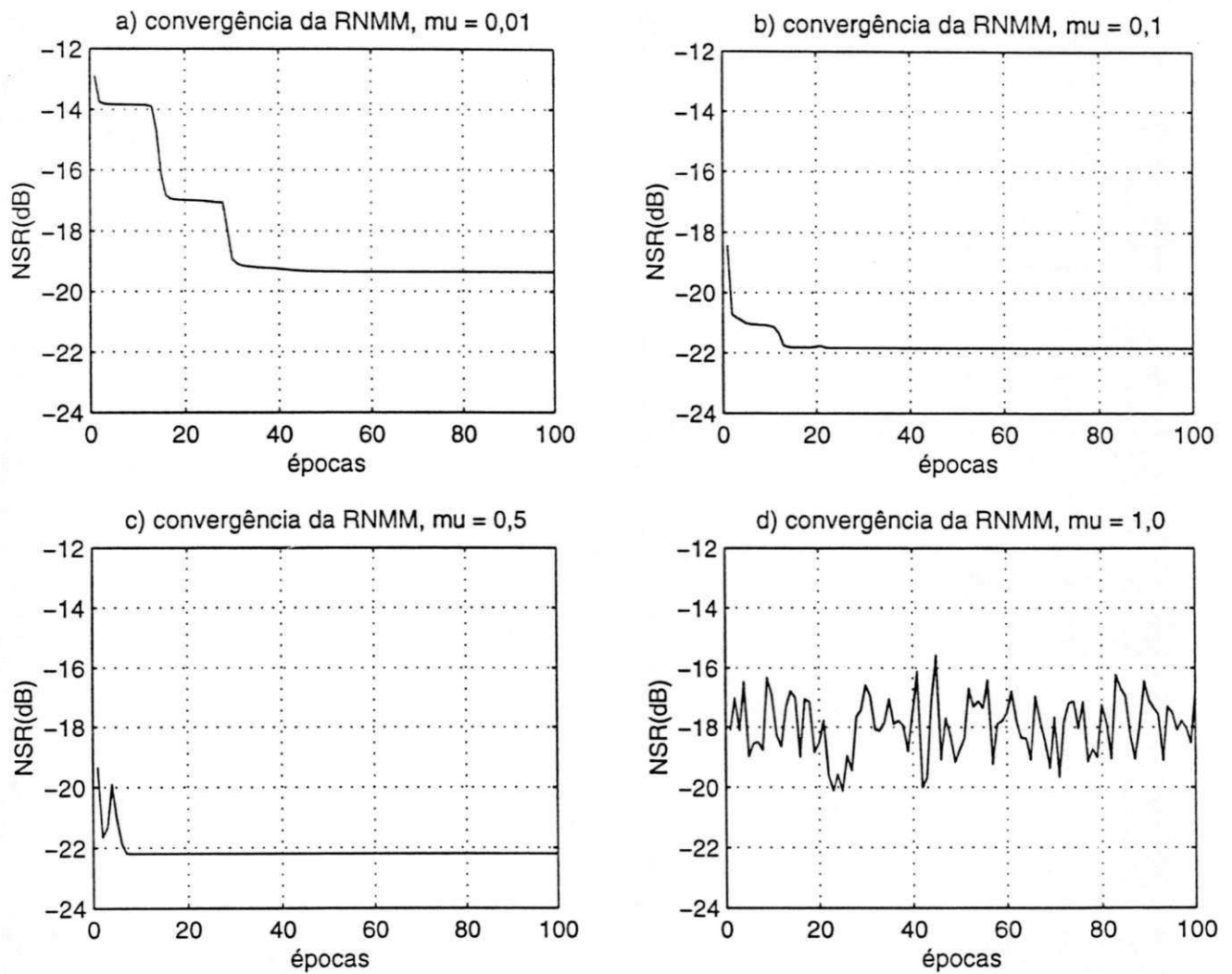


Figura 3.3: Convergência da RNMM via decomposição de erosões através da NSR. A rede teve melhor comportamento de convergência nos casos (a) e (b) ($\mu = 0,01$ e $\mu = 0,1$).

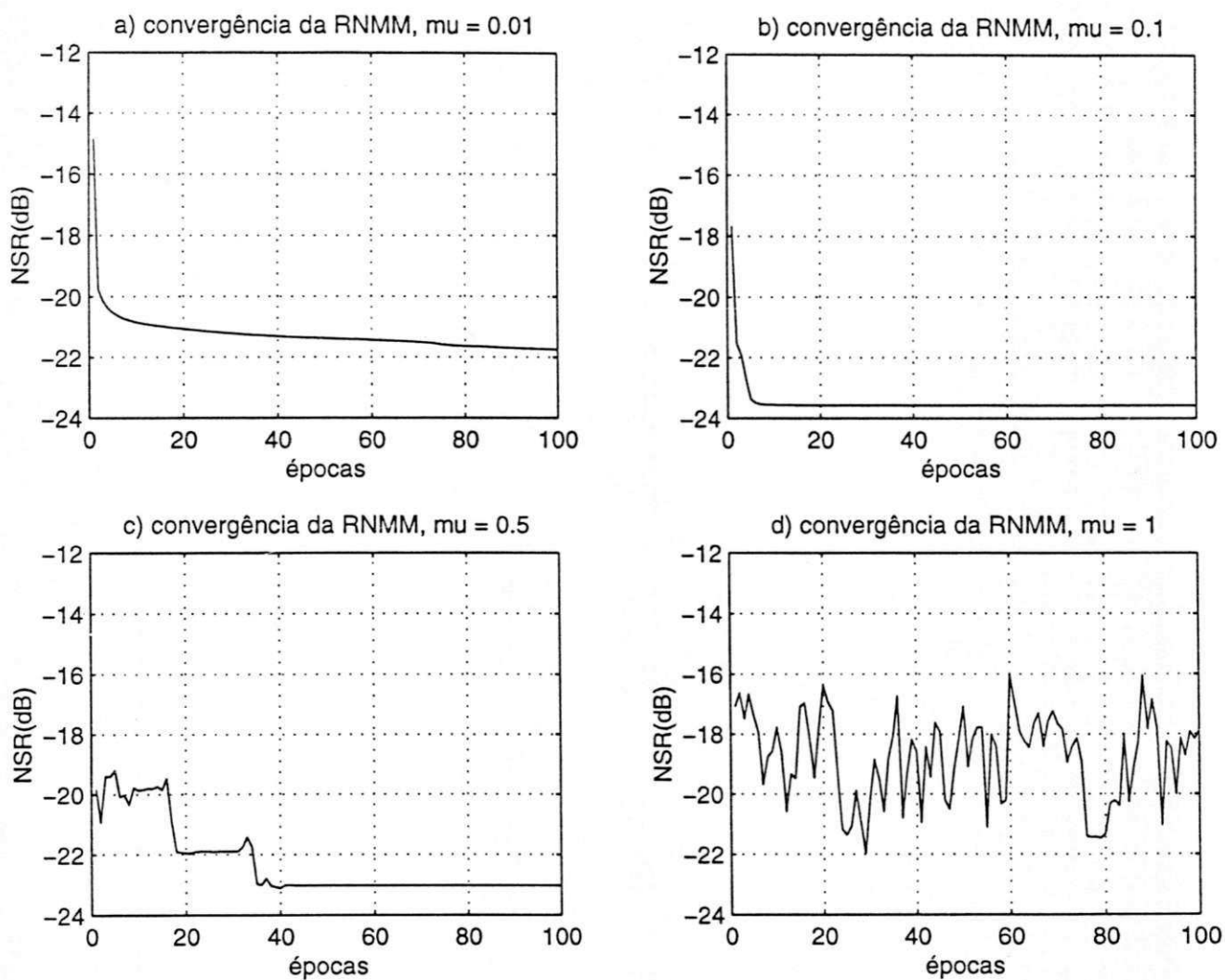


Figura 3.4: Convergência da RNMM via decomposição por dilatações através da NSR. A rede teve melhor comportamento de convergência nos casos (a) e (b) ($\mu=0,01$ e $\mu=0,1$).

a) X – treinamento



b) saída



c) X – teste



d) saída



Figura 3.5: Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 1$ a rede não aprendeu.

a) X – treinamento



b) saída



c) X – teste



d) saída



Figura 3.6: Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,01$ a rede aprendeu.

a) X – treinamento



b) saída



c) X – teste



d) saída

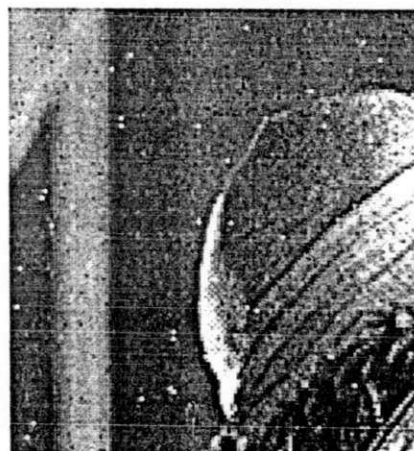


Figura 3.7: Imagens de treinamento e teste da RNMM via decomposição por 8 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 1$ a rede não aprendeu.

a) X – treinamento



b) saída



c) X – teste



d) saída



Figura 3.8: Imagens de treinamento e teste da RNMM via decomposição por 8 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,01$ a rede aprendeu.

a) X - treinamento



b) saída



c) X - teste



d) saída



Figura 3.9: Imagens de treinamento e teste da RNMM via decomposição por 25 erosões. (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,1$.

a) X - treinamento



b) saída



c) X - teste



d) saída



Figura 3.10: As imagens mostram a saída do filtro projetado com a solução obtida via treinamento da rede com 25 erosões. (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Parâmetro de convergência $\mu = 0,1$.

N^0 de erosões	N^0 de Épocas	μ	Densidade de Ruídos %	NSR inicial (dB)	NSR final (dB)
8	100	0,01	5	-12,8112	-19,65
8	100	0,1	5	-12,6655	-17,94
8	100	0,5	5	-12,7575	-14,15
8	100	1,0	5	-12,8016	-6,9

Tabela 3.1: Desempenho da RNMM para decomposição via erosões.

N^0 de Geradores	Inf-	N^0 de Épocas	μ	Densidade de Ruídos %	NSR inicial (dB)	NSR final (dB)
8		100	0,01	5	-12,8112	-21,10
8		100	0,1	5	-12,6655	-21,07
8		100	0,5	5	-12,7575	-19,9
8		100	1,0	5	-12,8016	-12,63

Tabela 3.2: Desempenho da RNMM Para decomposição via dilatações.

de 5% (figura 3.10). A imagem filtrada com o filtro Ψ^* projetado com 25 erosões resultou em NSR igual a $-24,96dB$. O filtro da mediana clássico teve um desempenho inferior, ficando com NSR igual a $-23,15dB$.

Denominação	N^0 de Épocas	μ	Densidade de Ruídos %	NSR final (dB)
RNMM-ERO(8)	100	0,01	5	-19,65
RNMM-DIL(8)	100	0,01	5	-21,10
FILTRO MEDIANA			5	-21,55
RNMM-ERO(25)	100	0,1	5	-22,42
RNMM-DIL(25)	100	0,1	5	-22,63

Tabela 3.3: Comparações entre a RNMM com o filtro da mediana. Observa-se que o melhor resultado foi para o caso em que o número de erosões que compõem a arquitetura da RNMM é igual a 25

Baseado nos experimentos apresentados nessa seção, podemos salientiar os seguintes fatos:

1. O desempenho da RNMM para o problema de filtragem de imagens ruidosas depende fundamentalmente do ajuste dos parâmetros da rede, isto é, do valor de μ e do número de módulos (erosões ou dilatações).
2. A complexidade computacional é moderada, uma vez que a rede só realiza dois tipos de operações: soma e ordenação.
3. Apesar de termos obtido valores de NSR final melhores do que os produzidos pelo filtro da mediana, a complexidade computacional da RNMM correspondente teve de aumentar consideravelmente, ficando a rede com 25 módulos.
4. A RNMM pode ser utilizada para investigar o comportamento de diversos tipos de decomposição de operadores crescentes e invariantes à translação.

3.4 Aplicação em problemas de extração de bordas (RNMM via decomposição de Banon e Barrera)

Um problema importante na área de processamento de imagens é o de extração de bordas, muito utilizada como etapa preliminar no reconhecimento de objetos contidos na imagem. Bordas correspondem a regiões de alto gradiente na imagem, ou seja, regiões onde ocorrem transições abruptas de níveis de cinza. Estas regiões correspondem, em geral, aos contornos dos objetos na imagem. Dessa forma, uma maneira de extrair ou identificar bordas, é através do cálculo do gradiente da imagem.

Nesta seção o nosso objetivo é treinar a RNMM via decomposição de Banon e Barrera para aprender o operador morfológico que extrai o gradiente da imagem, denominado de gradiente morfológico [33]. O gradiente morfológico utilizado neste trabalho é definido por:

$$G = X - \text{Min}(\bar{x} - \bar{a}_k) \quad (3.47)$$

3.4.1 Simulação

Em todas as simulações realizadas foi utilizada 25% da imagem X para treinar a RNMM via retro-propagação do erro (figura 3.11), tanto para imagens em níveis de cinza como para imagens binárias. Para testar o projeto do operador que extrai bordas da imagem X , utilizou-se a imagem X total. A arquitetura da rede para este experimento foi definida com 8 sup-geradores. Os parâmetros da RNMM foram definidos da seguinte forma: $\mu = 0,01$, número de épocas igual a 100. Treinamos a rede para extrair bordas com imagens em níveis de cinza e imagens binárias.

A figura 3.12 (d), mostra a saída final produzida pelo operador projetado. Conforme se observa as plumas do chapéu se destacam muito bem. Já a figura 3.13 (d), mostra um bom desempenho do operador projetado com a solução obtida da RNMM para

imagens binárias. Conforme se vê, as bordas da imagem estão bem distintas idicando uma boa aproximação do gradiente.

3.5 Aplicação da RNMM em Reconhecimento de Padrões (decomposição de Banon e Barrera)

A verificação da eficiência e limitações da RNMM em problemas de classificação, é realizada através de um experimento que consiste em classificar pontos gerados aleatoriamente em dois quadrados no plano.

O conjunto de treinamento da RNMM é gerado aleatoriamente, para o qual os seguintes parâmetros devem ser definidos:

- $\bar{x} = (x_1, x_2)$ vetor de atributos.
- Número de elementos por classe: N_e .
- Referência horizontal para o centro de gravidade: h_o .
- Referência vertical para o centro de gravidade: v_e .
- Distância entre os quadrados (classes): D .
- Comprimento do lado do quadrado: L .
- Ângulo (em graus) para rotacionar os quadrados (as classes) com respeito ao centro de gravidade: θ .

3.5.1 Experimentos

Os experimentos têm como objetivo observar o desempenho da RNMM em dois problemas de classificação. A estratégia é treinar a RNMM aumentando paulatinamente o número de sup-geradores que a compõem, iniciando com um sup-gerador e indo até oito sup-geradores. O primeiro conjunto de treinamento X_1 é definido com os seguintes parâmetros: $N_e = 1000; h_o = 1; v_e = 1; D = 2,5; L = 2; \theta = 45$. A figura 3.14 mostra o conjunto de treinamento X_1 , com os pontos gerados aleatoriamente nas duas classes. A figura 3.15 mostra o segundo conjunto de treinamento X_2 , definido com os seguintes parâmetros: $N_e = 1000; h_o = 0; v_e = 0; D = 2,5; L = 2; \theta = 0$. Observe que

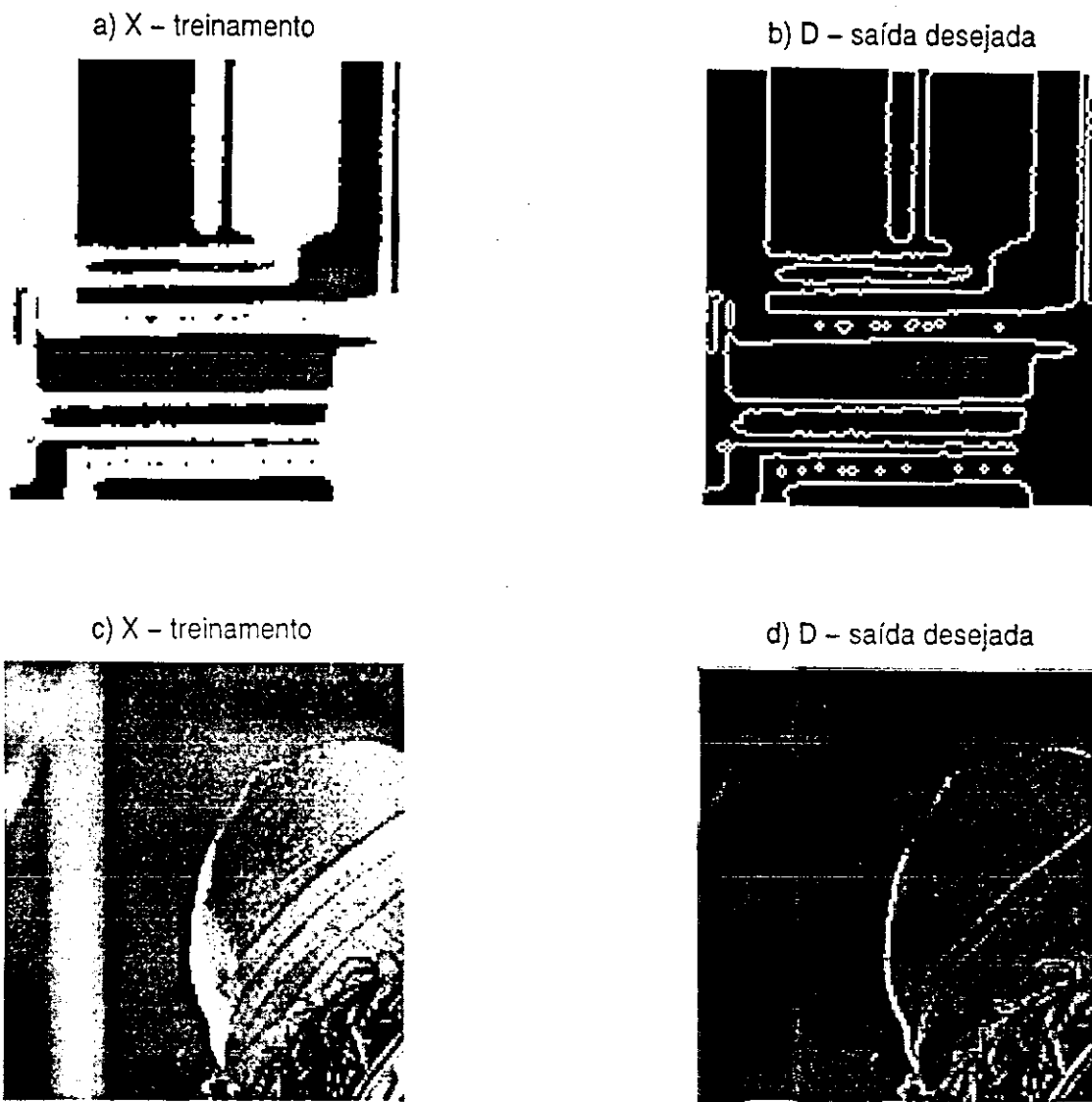


Figura 3.11: Imagens de treinamento da RNMM para o problema de extração de bordas. (a) imagem binária de entrada, (b) imagem binária desejada (gradiente), (c) imagens em níveis de cinza, (d) imagem em níveis de cinza desejada (gradiente).

a) X – treinamento



b) saída



c) X – teste



d) saída



Figura 3.12: Imagens de treinamento e teste da RNMM via decomposição por 8 supergeradores. (a) Imagem de entrada, (b) saída final do treinamento, (c) entrada de teste, (d) saída do teste (gradiente obtido). Imagem em níveis de cinza.

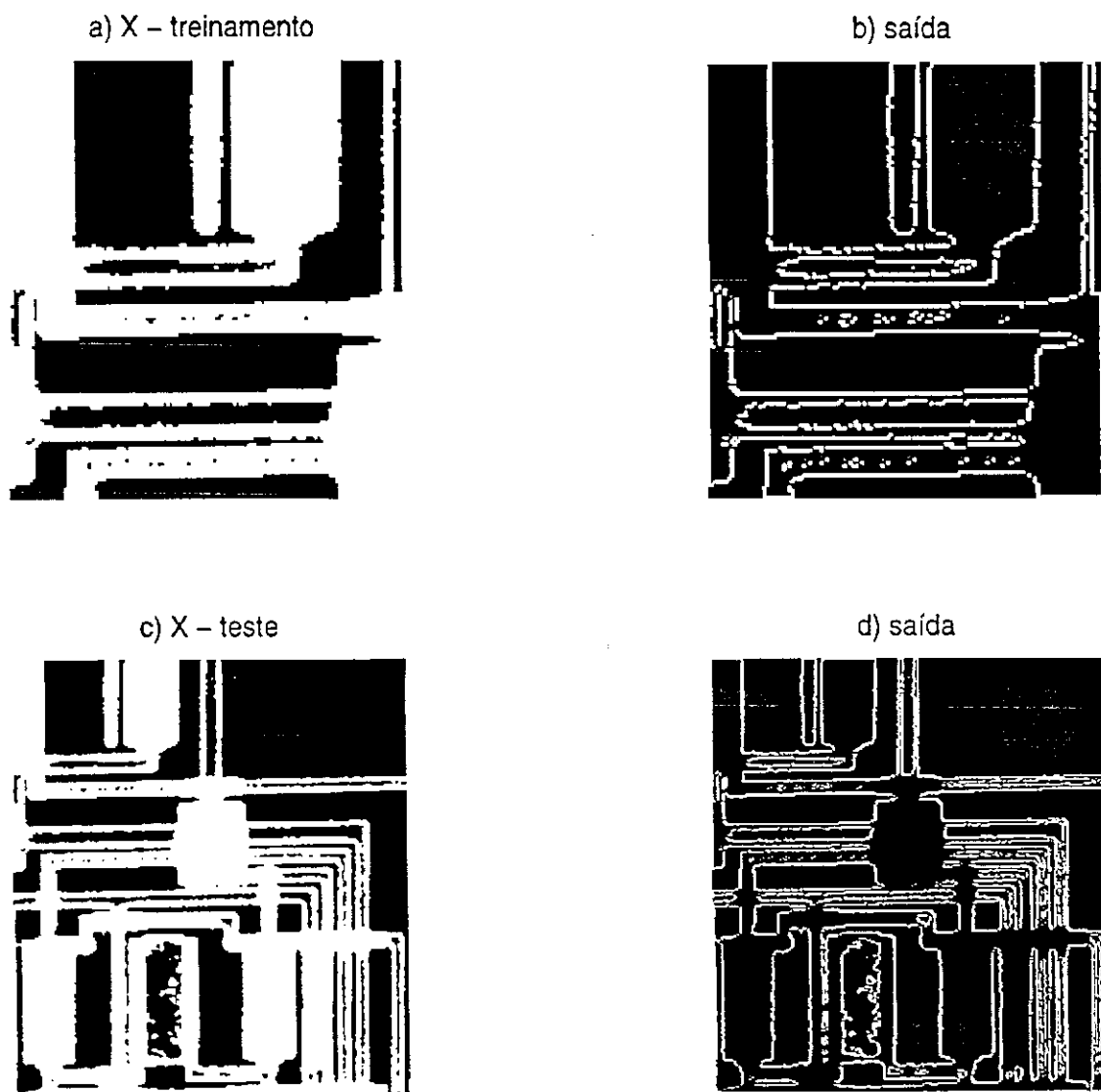


Figura 3.13: Imagens de treinamento e teste da RNMM via decomposição por 8 supergeradores. (a) Imagem de entrada, (b) saída final do treinamento, (c) entrada de teste, (d) saída do teste (gradiente obtido). Imagem binária

temos total controle sobre os conjuntos de treinamentos X_1 e X_2 . Com isso podemos forçar uma situação ideal de classificação; por exemplo: aumentando a distância entre os dois quadrados (classes), rotacionando os quadrados em relação ao eixo central, etc. A razão do erro de classificação é definida por:

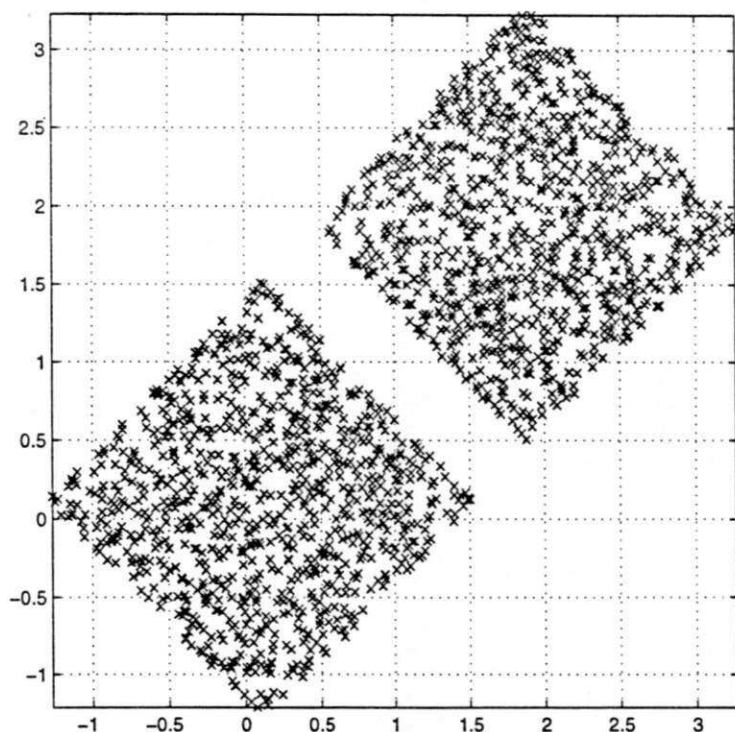


Figura 3.14: Conjunto de treinamento X_1

$$E(T) = \frac{\# \text{ de pontos mal classificados}}{\# \text{ total de pontos}} \cdot 100 \quad (3.48)$$

$T \in [0, 1]$ é o limiar de classificação, neste experimento foi usado $T = 0,5$. Portanto, se a saída do classificador for menor do que 0,5, o ponto pertence ao primeiro quadrado; caso contrário pertence ao segundo.

A tabela 3.4 mostra os resultados do experimento realizado com o classificador projetado via treinamento da RNMM, com o primeiro conjunto de treinamento X_1 , em que o ângulo de inclinação dos quadrados em relação ao eixo central é de 45° . Testamos

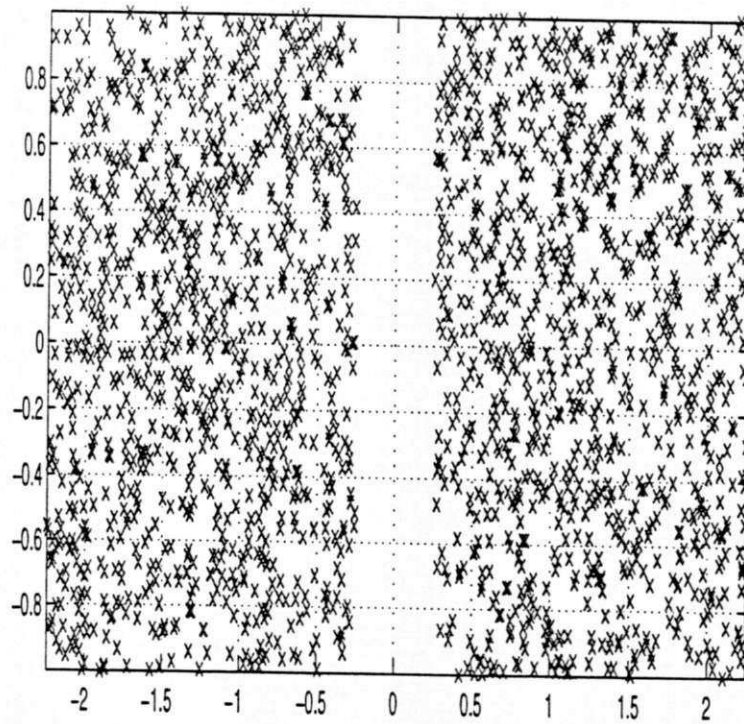


Figura 3.15: Conjunto de treinamento X_2

o classificador gerando aleatoriamente um conjunto de teste contendo 20000 pontos em cada quadrado. Na tabela 3.4 vemos que para a rede composta de um sup-gerador obtivemos uma razão de erro de classificação $E(T)$ de 8,42%. Para a rede composta com sete sup-geradores, obtivemos a razão de erro de classificação $E(T)$ de 1,44 %. Notamos também uma variabilidade do erro de classificação com relação ao número de sup-geradores contido na arquitetura da rede. Os resultados apresentados na tabela 3.4 não nos permitem tirar conclusões definitivas, no entanto, sugerem que uma boa classificação depende do número de sup-geradores que definem a rede, como do conjunto de dados para o treinamento.

A tabela 3.5 mostra os resultados do classificador projetado via treinamento da RNMM, com o conjunto de treinamento X_2 . Os resultados obtidos foram bons, mostrando inicialmente um erro de classificação em torno de 17,2%, para a arquitetura da rede com apenas um sup-gerador. Conforme observamos, o erro de classificação é anulado para a arquitetura da rede composta de 4,5,6,7 e 8 sup-geradores, confirmando assim nossa observação anterior, de que o bom desempenho de classificação da rede depende tanto do conjunto de treinamento bem como do número de sup-geradores que compõem a arquitetura da RNMM.

N^0 de Geradores	Sup-	N^0 de Épocas	μ	Razão do Erro de classificação em %
1		50	0,01	8,43
2		50	0,01	3,38
3		50	0,01	7,19
4		50	0,01	2,80
5		50	0,01	3,00
6		50	0,01	2,70
7		50	0,01	1,44
8		50	0,01	3,4

Tabela 3.4: Desempenho do classificador projetado via treinamento da RNMM com o conjunto de treinamento X_1

N ^o de Geradores	Sup-	N ^o de Épocas	μ	Razão do Erro de classificação em %
1		50	0,01	17,21
2		50	0,01	9,61
3		50	0,01	16,25
4		50	0,01	0,00
5		50	0,01	0,00
6		50	0,01	0,00
7		50	0,01	0,00
8		50	0,01	0,00

Tabela 3.5: Desempenho do classificador projetado via treinamento da RNMM com o conjunto de treinamento X_2 .

3.6 Considerações Finais

Vimos que o treinamento da RNMM via decomposição de Matheron [2] é bem eficiente no projeto de filtros para eliminação do ruído não Gaussiano do tipo sal e pimenta, superando em determinadas situações o filtro clássico da Mediana. A rede se mostrou capaz de aprender operadores do tipo crescente, sendo portanto uma ferramenta fundamental para projetos de operadores que satisfaçam as condições do teorema de Matheron.

Ao fazermos experimentação com a RNMM via decomposição de Banon e Barrera, exploramos toda a sua generalidade. O projeto do operador extrator de bordas foi eficiente pois, de acordo com os resultados apresentados, a RNMM mostrou que é capaz de aprender operadores do tipo não crescente, oferecendo, portanto, uma ferramenta alternativa para projetos de operadores morfológicos.

Com os experimentos de reconhecimento de padrões descritos neste capítulo, podemos concluir que a RNMM tem poder de classificação limitado, dependendo portanto do tipo de mapeamento ou seleção de atributos que é realizado nos dados. O exemplo dos quadrados no plano é bastante simples, quando comparado com outros problemas

na área de reconhecimento de padrões [34]. No entanto, conforme vimos, o mesmo nos permite concluir que as redes puramente morfológica tem o poder de classificação limitado, levando a crer que as superfícies de classificação são paralelas aos planos de referência (definidos pelos eixos de referência do espaço vetorial).

No Próximo capítulo, serão apresentados os resultados do treinamento da RNMM via Algoritmo genético e comparados com os resultados obtidos através do treinamento via retro-propagação do erro.

Capítulo 4

Treinamento da RNMM Via Algoritmo Genético

Neste capítulo apresentaremos os resultados do treinamento da RNMM via algoritmos genéticos, e faremos comparações de desempenho entre o método de treinamento da RNMM via retro-propagação do erro (capítulo 3) e o método proposto neste capítulo. Abordaremos a decomposição de Banon e Barrera [8] bem como a decomposição de Matheron [2].

4.1 Introdução

Harvey e Marshall [6] empregaram um algoritmo genético simples que utiliza três operadores básicos: reprodução, cruzamento e mutação [7] na busca do filtro morfológico ótimo para uma particular tarefa. No entanto, não estenderam o trabalho para a busca do conjunto de elementos estruturantes que satisfaz o teorema da decomposição de Matheron [2] para operadores (filtros) crescentes invariantes à translação ou, no caso geral da decomposição Banon e Barrera [8], para operadores invariantes à translação, não necessariamente crescentes. Oliveira [9] utilizou algoritmos genéticos simples [7], para determinar o conjunto sub-ótimo de elementos estruturantes que compõem a decomposição canônica via sup-geradores [8]. É importante salientar que a principal preocupação de Oliveira [9] foi a de estabelecer uma forma de implementar, sob o

ponto de vista prático a decomposição de operadores invariantes à translação.

Baseado nas idéias dos trabalhos de Harvey [6] e Oliveira [9], uma alternativa para o treinamento da RNMM é o uso de Algoritmos Genéticos (AG). Basicamente, as aplicações de AG na área de redes neurais são duas: estimação dos parâmetros da rede (estimar os pesos) [35] e síntese da arquitetura da rede. Na síntese, a idéia é realizar uma busca genética no espaço de possíveis arquiteturas da rede neural. Um exemplo deste tipo de aplicação é o sistema Neuro-Genesys [36].

Neste capítulo abordaremos o primeiro problema, isto é, treinaremos a RNMM via AG, com objetivo de determinar um conjunto de pesos ótimos (elementos estruturantes (A^*, B^*)).

4.2 Algoritmo Genético

Nos anos sessenta, os Algoritmos Genéticos foram pesquisados e desenvolvidos por Holland [37]. Sua pesquisa inicial não foi para planejar algoritmos que resolvessem problemas específicos, mas para formalizar o fenômeno de adaptação que ocorre na natureza e desenvolver mecanismos com os quais os procedimentos de adaptação naturais pudessem ser incorporados a sistemas de computação. Em 1975, Holland apresenta em seu livro [37] o algoritmo genético como uma abstração da evolução biológica. O método em si evolui uma população inicial para uma nova, pelo uso da seleção natural com operadores inspirados geneticamente. Os elementos básicos que compõe o algoritmo genético são:

1. População de cromossomos. Cada posição no cromossomo tem dois possíveis alelos (0 ou 1). Cada cromossomo representa um ponto no espaço de busca Ω .
2. Critério de seleção de acordo com a adequabilidade do indivíduo na população.
3. Mecanismo de cruzamento que produz novos filhos.
4. Mecanismo de mutação aleatória dos novos filhos.

5. Operadores genéticos.

a) Seleção: Este operador seleciona o cromossomo na população para reprodução. Um dado cromossomo pode ser escolhido mais de uma vez, dependendo de sua adequabilidade na população.

b) Cruzamento: Este operador escolhe uma posição no cromossomo e troca os subsequentes. O processo envolve dois cromossomos para criar dois novos filhos (figura 4.1). Trata-se de uma operação aleatória que ocorre com probabilidade P_c , geralmente entre 0,75 e 1,00.

c) Mutação: Seleciona-se aleatoriamente uma posição nos novos cromossomos (filhos) e muda-se o valor do gene correspondente para um outro alelo possível. Geralmente a probabilidade P_m de mutação é muito pequena, variando entre 0,0001 e 0,1.

Um algoritmo genético simples [38] é apresentado a seguir:

1. Estabeleça uma população base de cromossomos.
2. Determine o valor da adequabilidade de cada cromossomo.
3. Duplique os cromossomos de acordo com seu valor de adequabilidade e crie novos cromossomos por casamento dos cromossomos correntes, e aplique os operadores de cruzamento e mutação.
4. Delete todos os membros indesejáveis da população.
5. Insira os novos cromossomos na população e forme uma nova população.
6. Se o critério de parada é satisfeito, então pare e retorne o melhor cromossomo. Caso contrário, vá para o passo 2.

A figura 4.2 mostra o funcionamento do algoritmo genético com os seus principais operadores. Na próxima seção, apresentaremos o treinamento da RNMM via algoritmo genético. O algoritmo utilizado é praticamente o mesmo, mas com algumas modificações para satisfazer as condições da RNMM.

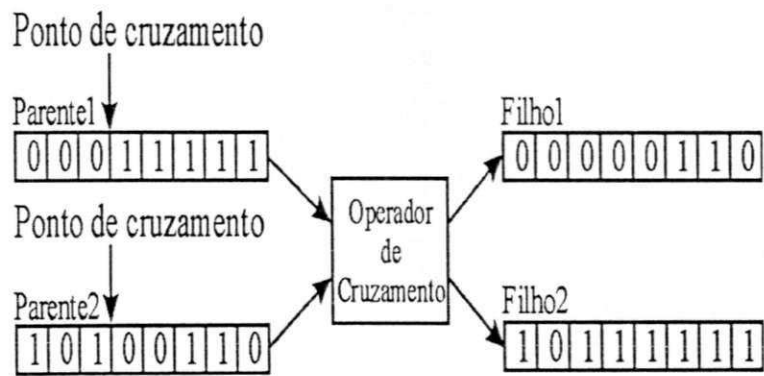


Figura 4.1: Cruzamento simples

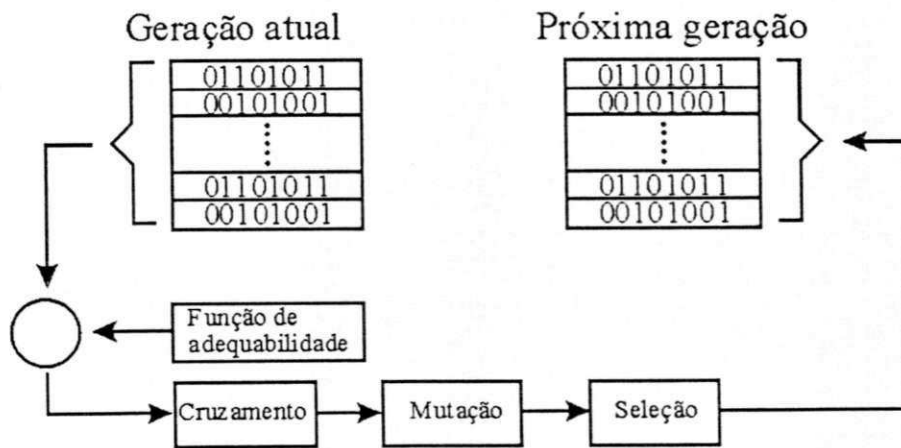


Figura 4.2: Ciclo do algoritmo genético e seus principais operadores.

4.3 Usando Algoritmo Genético para treinar a RNMM

O algoritmo de treinamento da RNMM segue o esquema da figura 4.2. Os detalhes do treinamento da RNMM via algoritmo genético serão apresentados a seguir.

O treinamento da RNMM via algoritmo genético é supervisionado, sob o critério da função de adequabilidade E (que será posteriormente definida). O objetivo principal é determinar um conjunto ótimo de matrizes pesos A^* e B^* (elementos estruturantes). No caso da decomposição de Matheron, basta determinar a Matriz peso A .

Seja então

$$\{X, D\} \quad (4.1)$$

o conjunto de treinamento da RNMM, em que X é a imagem de entrada e D a imagem desejada.

A função de erro ou imagem erro, que é utilizada durante o treinamento, é definida por

$$E = D - Y \quad (4.2)$$

em que Y é a saída da rede. No caso de imagens binárias, é suficiente utilizar a relação $E = (D \text{ XOR } Y)$, no qual o operador (**XOR**) representa o operador (**OU**) exclusivo. Afim de manter uma uniformidade geral no trabalho, adotaremos a razão ruído-sinal (NSR) definida no capítulo 3 (equação 3.46), como medida de adequabilidade dos elementos da população. portanto, a equação de adequabilidade é definida da seguinte forma:

$$NSR = 10 \log_{10} \frac{\overline{E^2}}{\overline{D^2}} \quad (4.3)$$

em que $\overline{E^2}$ é a energia média da imagem erro (segundo momento do erro), e $\overline{D^2}$ é a energia média da imagem objetivo (segundo momento do objetivo). Quanto menor for o erro, maior será a adequação do indivíduo na população.

Observe que existe uma diferença entre o treinamento da RNMM via algoritmo genético e via algoritmo de retro-propagação do erro. No caso do treinamento via retro-propagação do erro visto no capítulo 3, a lógica de adaptação é local, e o gradiente é estimado para cada janela da imagem. No treinamento via algoritmo genético, os dados de entrada para treinamento são as imagens com ruído e sem ruído. Por outro lado, não é necessário codificar os pesos da rede (elementos estruturantes), pois neste caso especial, utilizaremos elementos estruturantes binários para imagens binárias. No caso de imagens em níveis de cinza, utilizamos elementos estruturantes planos, que de certa forma, são binários.

4.3.1 Detalhando o algoritmo de treinamento AG da RNMM (decomposição de Banon e Barrera)

O algoritmo de treinamento da RNMM é descrito a seguir.,

1. Fixe a arquitetura da RNMM, isto é, um número fixo de sup-geradores ou inf-geradores.
2. Gere aleatoriamente um conjunto de matrizes $(A, B)_i, i = 1, 2, \dots$ que parametrizam a RNMM. Cada par $(A, B)_i$ corresponde a um elemento da população. Por exemplo, se $i = 30$, o tamanho da população é 30, correspondendo a 30 pares de matrizes. Cada matriz A e B contém os elementos estruturantes que parametrizam a RNMM. O número de linhas de A e B representa o número de elementos estruturantes que compõem a rede, e o número de colunas representa o tamanho ou a dimensão dos elementos estruturantes. De forma geral, se a rede contém N sup-geradores, as matrizes de elementos estruturantes A e B terão respectivamente, N pares \vec{a}_k e $\vec{b}_k, k = 1, 2, \dots, N$ de vetores pesos.

$$A = \begin{pmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \cdot \\ \cdot \\ \bar{a}_N \end{pmatrix}, B = \begin{pmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \cdot \\ \cdot \\ \bar{b}_N \end{pmatrix}$$

3. Como os elementos da população são formados por cromossomos binários, é necessário fazer a seguinte transformação: $0 \Rightarrow -\infty$ e $1 \Rightarrow 0$, isto é, mapeia-se o 0 (zero) como menos infinito e o 1 (um) como (zero). Esta transformação é necessária devido a natureza das equações da rede. Computa-se a saída da população inicial através da RNMM e calcula-se a adequabilidade de cada elemento por meio da equação 4.3. Isto diz que, quanto menor for a *NSR*, maior será a adequação do elemento na população.
4. Recombina-se os elementos da população por um processo aleatório, em que a probabilidade de recombinação $P_c = 0.85$. O procedimento é o seguinte:
 - a) Dois cromossomos pais são selecionados aleatoriamente na população.
 - b) O ponto ou os pontos de recombinação são selecionados aleatoriamente ao longo do comprimento do cromossomo .
 - c) dois novos filhos são formados dos pais, trocando-se os genes à direita do ponto de recombinação entre os pais.
5. Utilize a mutação em um só ponto de cada filho com probabilidade $P_m = 0,01$.
6. Avalie a nova população através da RNMM.
7. Seleção: para cada imagem processada através da RNMM, o *NSR* é estimado e comparado com os valores de *NSR* dos dois pais. Caso o novo elemento gerado tenha *NSR* menor do que a dos pais, ele tomará o lugar daquele que tem maior *NSR*. Isto significa que no final do treinamento teremos um conjunto de elementos estruturantes com menor *NSR* que o inicial.

8. Pare o algoritmo caso o critério de parada seja satisfeito. Caso contrário vá para 3.

4.3.2 Detalhando o algoritmo de treinamento AG da RNMM (decomposição de Matheron)

A diferença em relação ao algoritmo anterior reside nos itens 1 e 2, ficando pois:

1. Fixe a arquitetura da RNMM, isto é, um número fixo de erosões ou dilatações.
2. Gere aleatoriamente um conjunto de matrizes $(A)_i, i = 1, 2, \dots$ que parametrizam a RNMM. Cada matriz $(A)_i$ corresponde a um elemento da população. Por exemplo, se $i = 30$, o tamanho da população é 30, correspondendo a 30 matrizes. Cada matriz A contém os elementos estruturantes que parametrizam a RNMM. O número de linhas de A representam o número de elementos estruturantes que compõem a rede, e número de colunas representam o tamanho ou a dimensão dos elementos estruturantes. De forma geral, se a rede contém N erosões, a matriz de elementos estruturantes A terá $\vec{a}_k, k = 1, 2, \dots, N$ elementos estruturantes.

$$A = \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vdots \\ \vec{a}_N \end{pmatrix}$$

3. Como os elementos da população são formados por cromossomos binários é necessário fazer a seguinte transformação: $0 \Rightarrow -\infty$ e $1 \Rightarrow 0$, isto é, mapeia-se o 0 (zero) como menos infinito e o 1 (um) como zero. Esta transformação é necessária devido a natureza das equações da rede. Computa-se a saída da população inicial através da RNMM e calcula-se a adequabilidade de cada elemento por meio da equação 4.3. Isto diz que, quanto menor for a NSR , maior será a adequação do elemento na população.

4. Recombina-se os elementos da população por um processo aleatório, em que a probabilidade de recombinação $P_c = 0.85$. O procedimento é o seguinte:
 - a) Dois cromossomos pais são selecionados aleatoriamente na população.
 - b) O ponto ou os pontos de recombinação são selecionados aleatoriamente ao longo do comprimento do cromossomo.
 - c) dois novos filhos são formados dos pais, trocando-se os genes à direita do ponto de recombinação entre os pais.
5. Utilize a mutação em um só ponto de cada filho com probabilidade $P_m = 0,01$.
6. Avalie a nova população através da RNMM.
7. Seleção: para cada imagem processada através da RNMM, o NSR é estimado e comparado com os valores de NSR dos dois pais. Caso o novo elemento gerado tenha NSR menor do que a dos pais, ele tomará o lugar daquele que tem maior NSR . Isto significa que no final do treinamento teremos um conjunto de elementos estruturantes com menor NSR que o inicial.
8. Pare o algoritmo caso o critério de parada seja satisfeito. Caso contrário vá para 3.

4.4 Aplicações em Restauração de Imagens (decomposição de Matheron)

O problema de filtragem de imagens com ruído do tipo sal pimenta foi bem definido no capítulo 3, não vamos repetir toda a formalização do mesmo. O nosso objetivo é comparar os resultados obtidos pela rede via treinamento genético com os resultados obtidos via treinamento de retro-propagação do erro, para os casos da RNMM (figura 2.1 e 2.2).

4.4.1 Simulações

Em todas as simulações realizadas foram utilizadas 25% da imagem ruidosa X (figura 3.2) para treinar a RNMM via algoritmo genético. Para testar o filtro projetado via treinamento da RNMM, utilizou-se a imagem X . Outras imagens em níveis de cinza, foram utilizadas para medir o desempenho do filtro.

Tomando também como base a experiência do treinamento da RNMM via retro-propagação do erro que produziu melhores resultados (tabela 3.3), fixaremos as redes em 25 erosões e 25 dilatações (dualmente), para fazer uma comparação com os resultados obtidos via treinamento AG. Treinamos a rede com uma população inicial de 30 elementos; o número máximo de gerações (ciclo do AG) 100, probabilidade de recombinação $P_c = 0,85$ e probabilidade de mutação $P_m = 0,01$.

A tabela 4.1 mostra os resultados obtidos do treinamento, para os casos da decomposição via erosões, dilatações e retro-propagação do erro. Conforme se observa, a rede treinada por retro-propagação obteve melhor resultado que as demais, produzindo uma $NSR = -22,42dB$, isto representa uma diferença pro RNMM via treinamento de retro-propagação em torno de $4dB$, com relação ao treinamento via AG para decomposição via erosões. Equivalentemente, em relação ao treinamento via AG para decomposição via dilatações, a diferença pro RNMM via treinamento retro-propagação é de aproximadamente $3dB$. As figuras 4.3 e 4.4 mostram as imagens de entrada e saída da saída dos filtros projetado com a solução obtida da RNMM via treinamento AG. A figura 4.3 mostra que o filtro composto por 25 erosões teve um bom desempenho com o ruído pimenta, enquanto a figura 4.4 mostra que o filtro composto por 25 dilatações teve desempenho melhor com o ruído sal.

Utilizando a solução obtida pela RNMM via treinamento AG, testamos o filtro com outra imagem. A figura 4.5 mostra os resultados para o filtro projetado com A^* para decomposição por erosões. Neste teste, a NSR final para saída do filtro, ficou em torno de $-20,02dB$. No caso do treinamento da RNMM via AG para decomposição por dilatações, a NSR final do filtro ficou em torno de $-20,09dB$, veja figura 4.6.

No caso de imagens binárias, treinamos a rede via retro-propagação do erro para decomposição por 8 erosões, com parâmetro de convergência $\mu = 0.01$ e em 50 épocas.

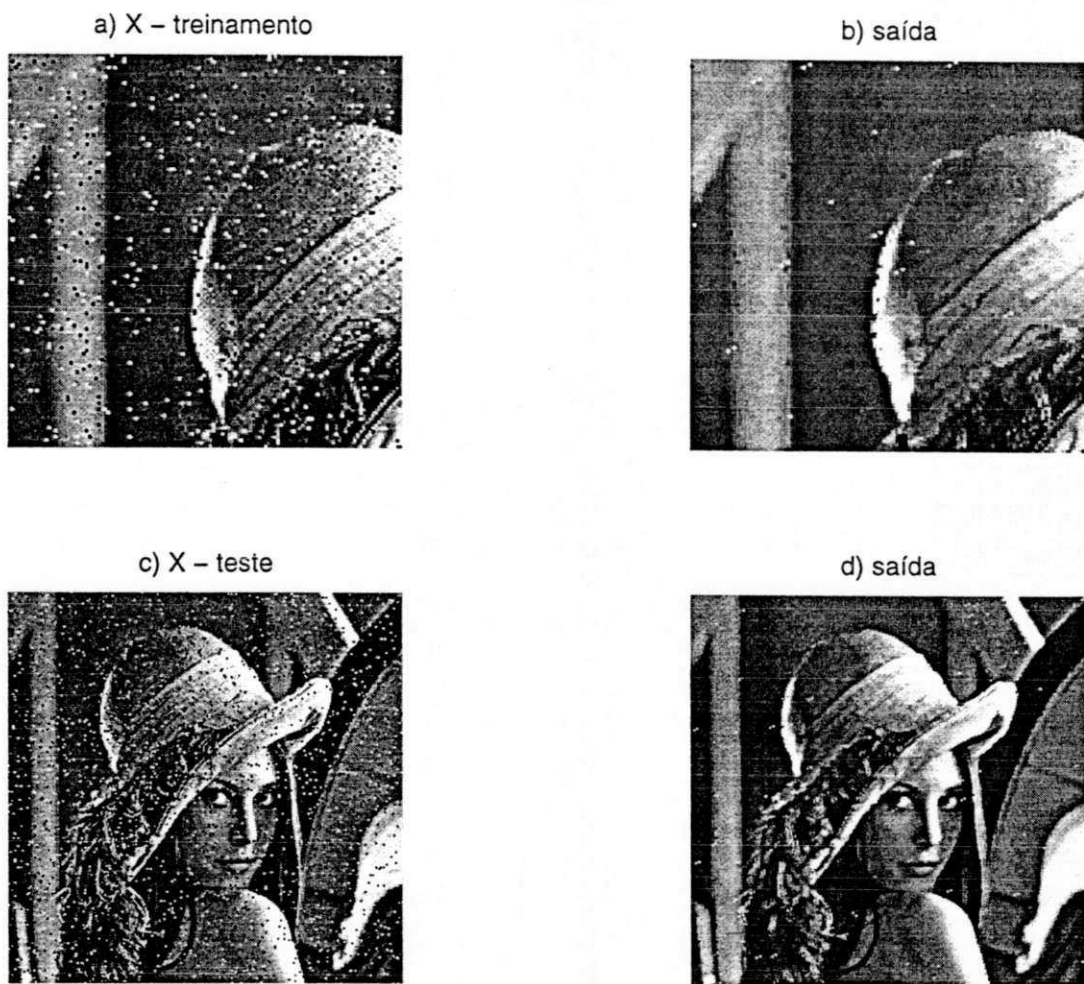
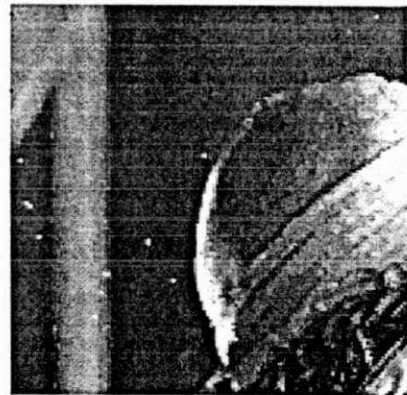


Figura 4.3: Imagens de treinamento e teste da RNMM via decomposição por 25 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.

a) X – treinamento



b) saída



c) X – teste



d) saída



Figura 4.4: Imagens de treinamento e teste da RNMM via decomposição por 25 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.

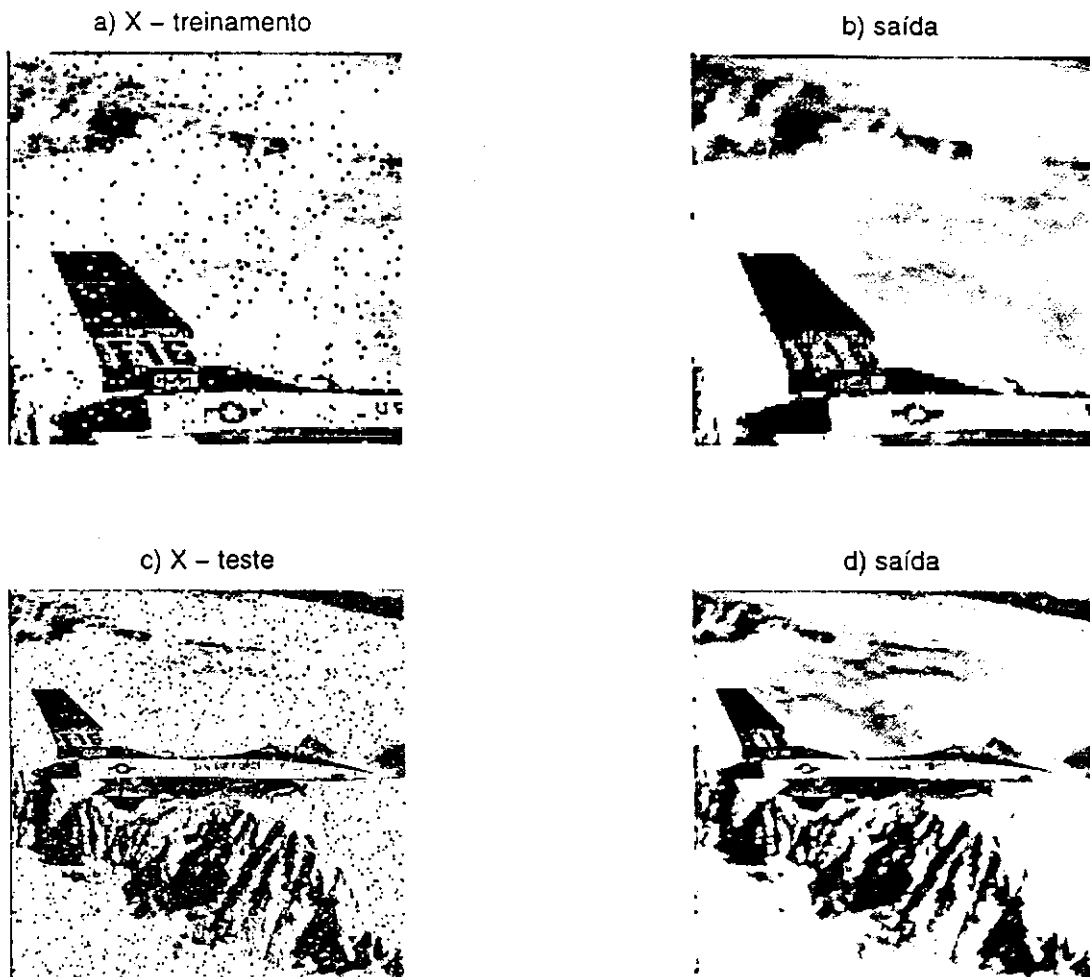
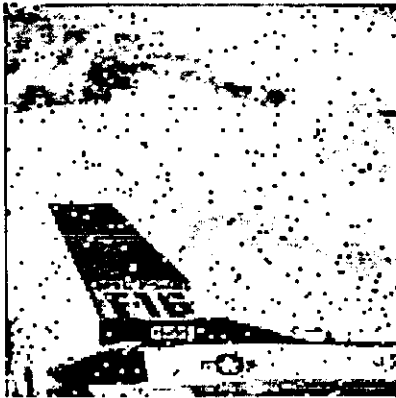


Figura 4.5: Imagem filtrada com a solução da RNMM via treinamento AG: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Decomposição por 25 erosões.

a) X – treinamento



b) saída



c) X – teste



d) saída



Figura 4.6: Imagem filtrada com a solução da RNMM via treinamento AG: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Decomposição por 25 dilatações.

Denominação	Tipo de treinamento	Nº de Épocas (Geração)	μ	Densidade de Ruídos %	<i>NSR</i> final (dB)
RNMM-ERO(25)	AG	100		5	-18,40
RNMM-DIL(25)	AG	100		5	-19,70
RNMM-ERO(25)	RP	100	0,1	5	-22,42
RNMM-DIL(25)	RP	100	0,1	5	-22,63

Tabela 4.1: Comparação entre os treinamentos da RNMM via AG e retro-propagação do erro, melhor resultado para o treinamento via retro-propagação.

Comparamos o resultado obtido do filtro projetado via treinamento retro-propagação do erro, com o resultado obtido do filtro projetado via treinamento da rede por AG para decomposição por 8 erosões. Os resultados foram melhores para o filtro projetado via treinamento AG, resultando em uma diferença pro AG em torno de $2dB$. As figuras 4.8 e 4.9 mostram as saídas finais dos dois filtros. É interessante observar o gráfico de convergência da *NSR* (figura 4.7) em relação aos dois tipos de treinamento. O gráfico de convergência para a RNMM via treinamento retro-propagação do erro se estabilizou por volta da época 10, enquanto o gráfico de convergência da RNMM via treinamento AG ainda não está estabilizado até a geração 50. Isto leva a crer que a tendência de queda irá se propagar nas gerações futuras, produzindo assim melhores desempenho para os filtros.

Treinamos também a rede via decomposição por 8 dilatações, o resultado foi muito bom, ficando a *NSR* final em torno de $-19dB$ (figura 4.10), a tabela 4.2 mostra os resultados da simulação.

A tabela 4.3 mostra o esforço computacional em relação aos dois tipos de treinamento da RNMM (RP e AG). Todos os algoritmos foram implementados em Matlab. Conforme se observa, o treinamento da RNMM via RP, obteve um menor esforço computacional para o projeto de filtros binários. Equanto que, o treinamento da RNMM via AG, obteve um menor esforço computacional para projetos de filtros em níveis de

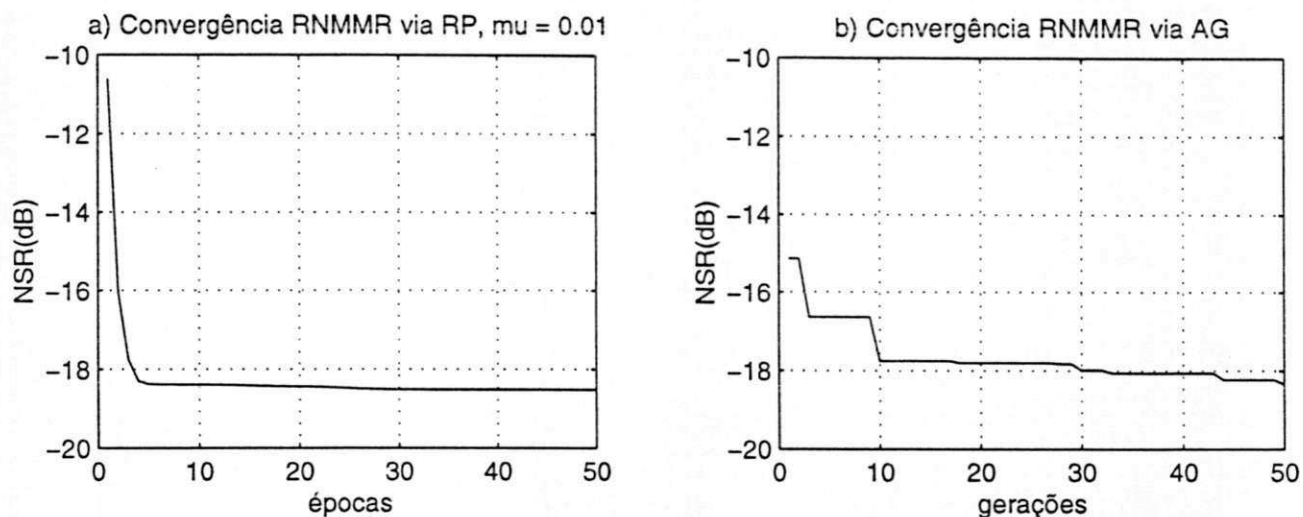


Figura 4.7: Convergência da RNMM via treinamento RP e AG. (a) decomposição por erosões e (b) decomposição por erosões

Denominação	Tipo de treinamento	N^0 de Épocas (Geração)	μ	Densidade de Ruídos %	NSR final (dB)
RNMM-ERO(8)	AG	50		5	-17,63
RNMM-DIL(8)	AG	50		5	-19,00
RNMM-ERO(8)	RP	50	0,01	5	-17,40
RNMM-DIL(8)	RP	50	0,01	5	-17,72

Tabela 4.2: Comparação entre os treinamentos da RNMM via AG e RP. Melhor resultado para o treinamento via AG.

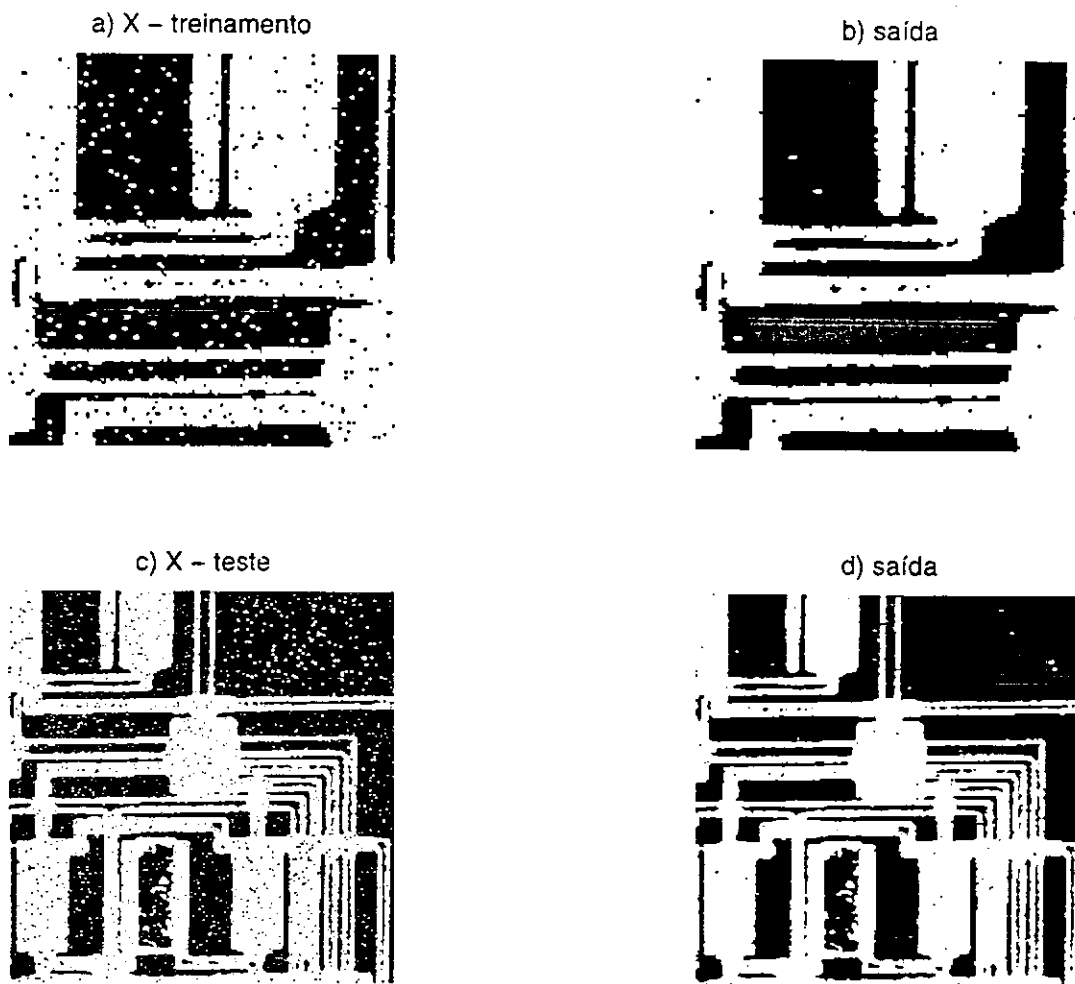


Figura 4.8: Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.

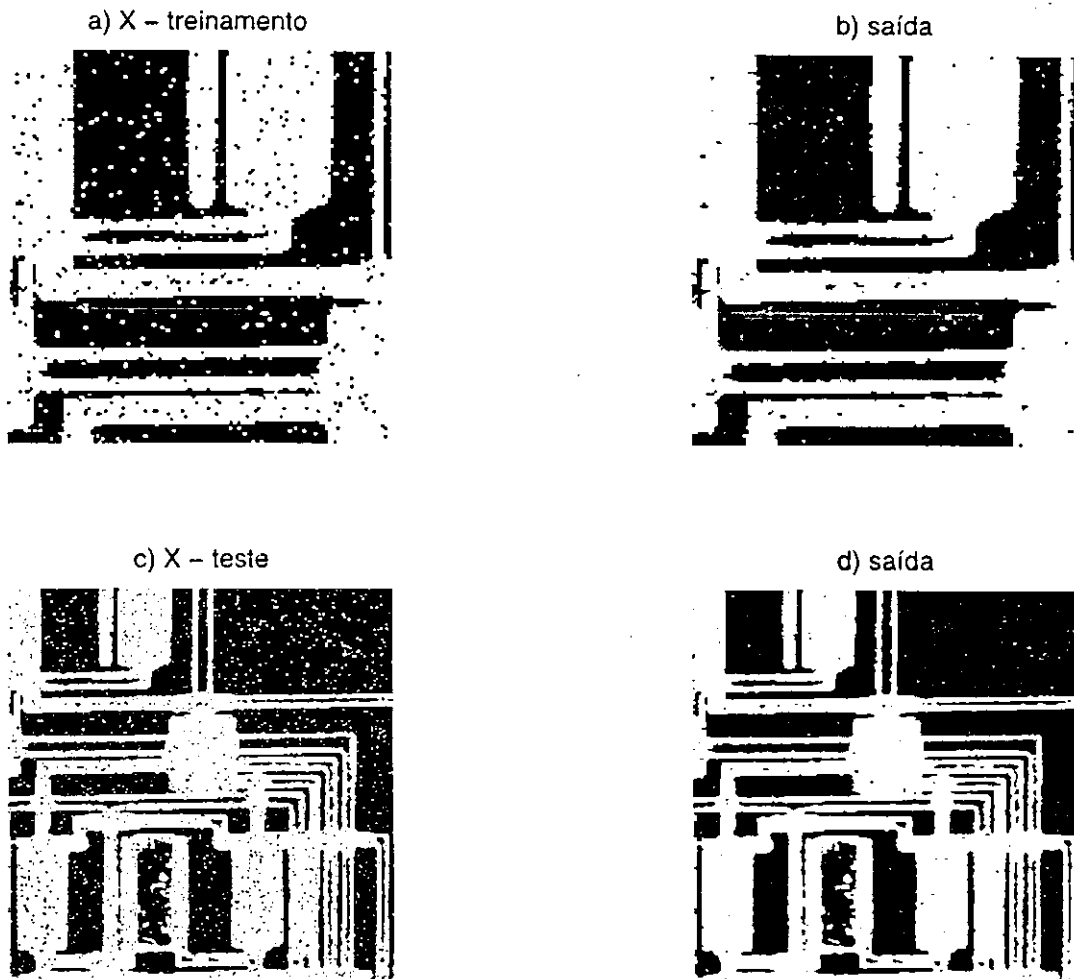


Figura 4.9: Imagens de treinamento e teste da RNMM via decomposição por 8 erosões: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por RP.

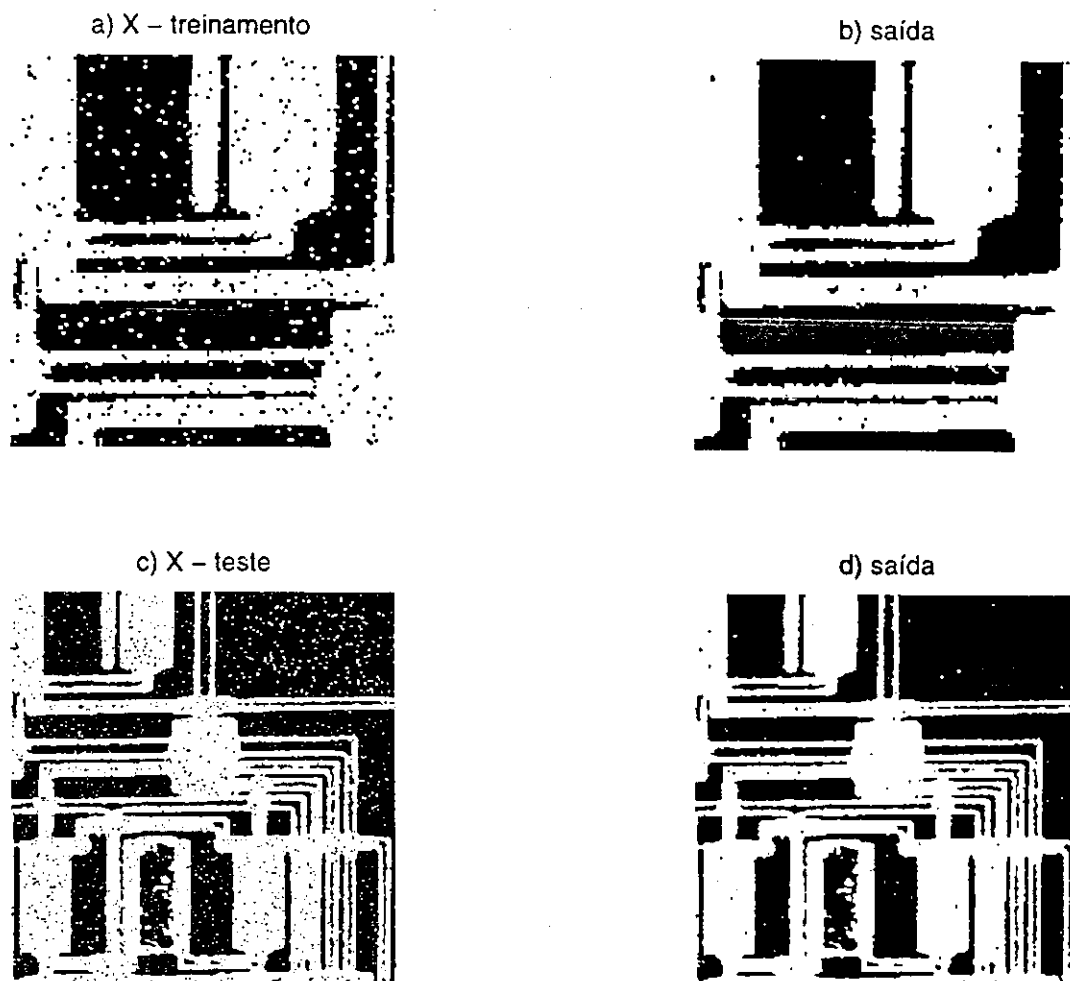


Figura 4.10: Imagens de treinamento e teste da RNMM via decomposição por 8 dilatações: (a) entrada para o treinamento, (b) saída ao final do treinamento, (c) entrada de teste, (d) saída do teste (imagem filtrada). Filtro projetado via treinamento da RNMM por AG.

cinza. Este tipo de complexidade computacional independe do tipo de máquina utilizada no treinamento da RNMM. Neste caso, mediu-se o número de operações por ponto flutuante (função *flops* do MATLAB) realizadas para os dois tipos de treinamento (RP e AG) da RNMM.

Denominação	Tipo de treinamento	N^0 de Épocas (Geração)	Tipo de imagem	N^0 de operações por ponto flutuante (<i>flops</i>)
RNMM-ERO(8)	AG	50	B	9,699667E+09
RNMM-DIL(8)	AG	50	B	9,772510E+09
RNMM-ERO(8)	RP	50	B	5,524500E+09
RNMM-DIL(8)	RP	50	B	5,534020E+09
RNMM-ERO(25)	AG	100	NC	5,053843E+010
RNMM-DIL(25)	AG	100	NC	5,061216E+010
RNMM-ERO(25)	RP	100	NC	1.4799E+11
RNMM-DIL(25)	RP	100	NC	1,4804E+011

Tabela 4.3: Esforço computacional: Número de operações realizadas entre os dois tipos de treinamento (RP e AG).

4.5 Aplicação em extração de Bordas (decomposição de Banon e Barrera)

Faremos agora uma aplicação da RNMM em problemas de extração de bordas de objetos contidos na imagem, contaminadas com ruído do tipo sal e pimenta. O treinamento da rede será via algoritmo genético, para decomposição via sup-geradora. O objetivo é conjugar em um único operador as operações de extração de bordas e remoção de ruído não Gaussiano.

4.5.1 Simulação

Em todas as simulações realizadas foram utilizadas 25% da imagem ruidosa X (figura 4.11) para treinar a RNMM via algoritmo genético. Para testar o projeto do operador que retira ruídos e extrai as bordas da imagem X , utilizou-se a imagem X total do treinamento da RNMM. A arquitetura da rede para este experimento foi definida com 8 sup-geradores. Os parâmetros do algoritmo genético são os mesmos dos experimentos anteriores. A figura 4.12 nos mostra a saída final do teste realizado com a imagem total do treinamento. O resultado é muito bom, conforme pode ser observado, o operador projetado com a solução da RNMM foi bem eficiente, tanto na extração de bordas bem como na extração de ruídos. Treinamos a RNMM via retro-propagação do erro via decomposição de Banon e Barrera (8 sup-geradores), com a mesma imagem deste experimento obtendo o resultado mostrado na figura 4.13. Observamos que o AG teve melhor desempenho que o RP. Este exemplo mostra toda a generalidade da decomposição de Banon e Barrera [8], haja visto que simultaneamente a rede aprende o operador extrator de borda bem como o operador removedor de ruído, possibilitando assim a viabilidade de projetos de operadores invariantes à translação tanto do tipo crescente como do tipo não crescente.



Figura 4.11: Conjunto de treinamento da RNMM para o experimento conjugado, isto é, filtragem de ruído e extração de bordas.

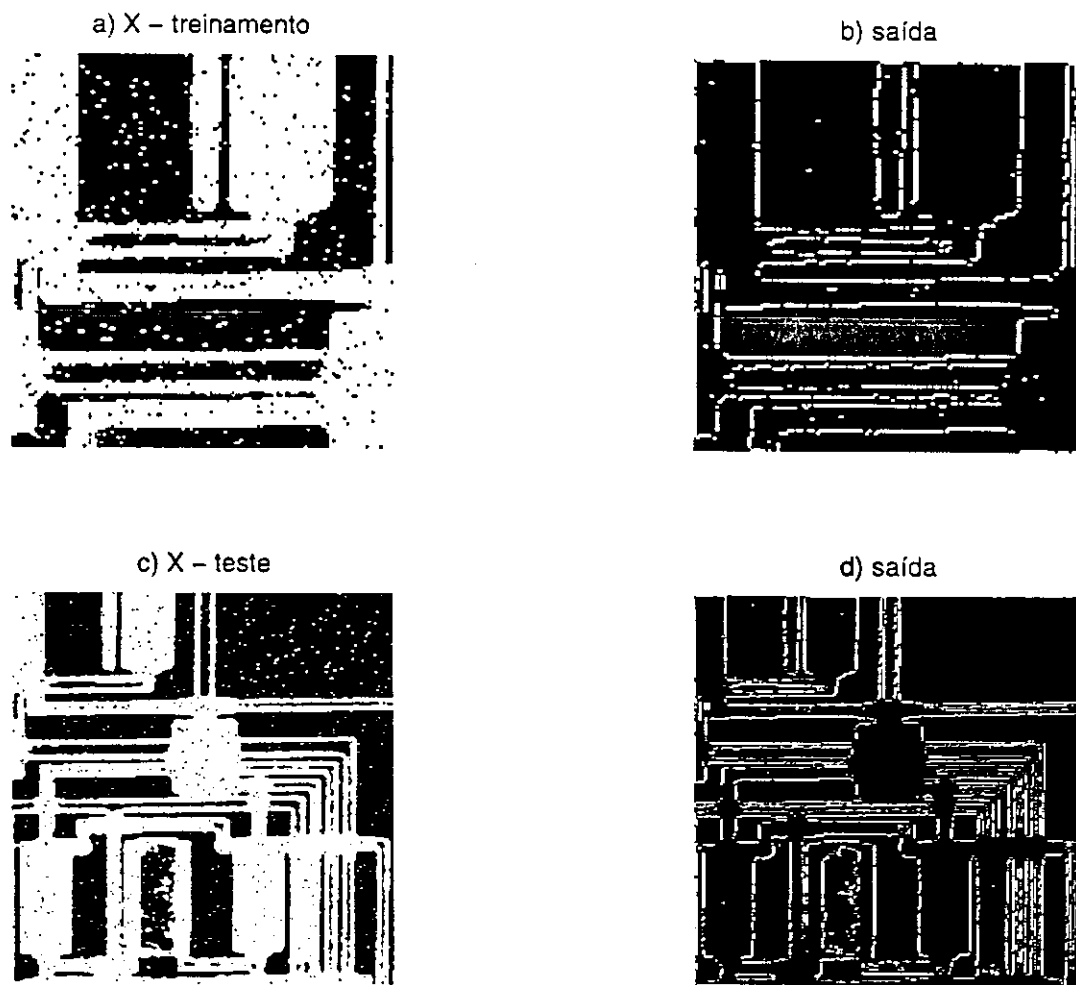


Figura 4.12: Imagens de treinamento da RNMM para o problema de extração de bordas e ruído, decomposição por 8 sup-geradores: (a) imagem binária de entrada, (b) imagem binária desejada, (c) imagens de teste, (d) saída do operador extrator de borda e ruído. Treinamento via AG.

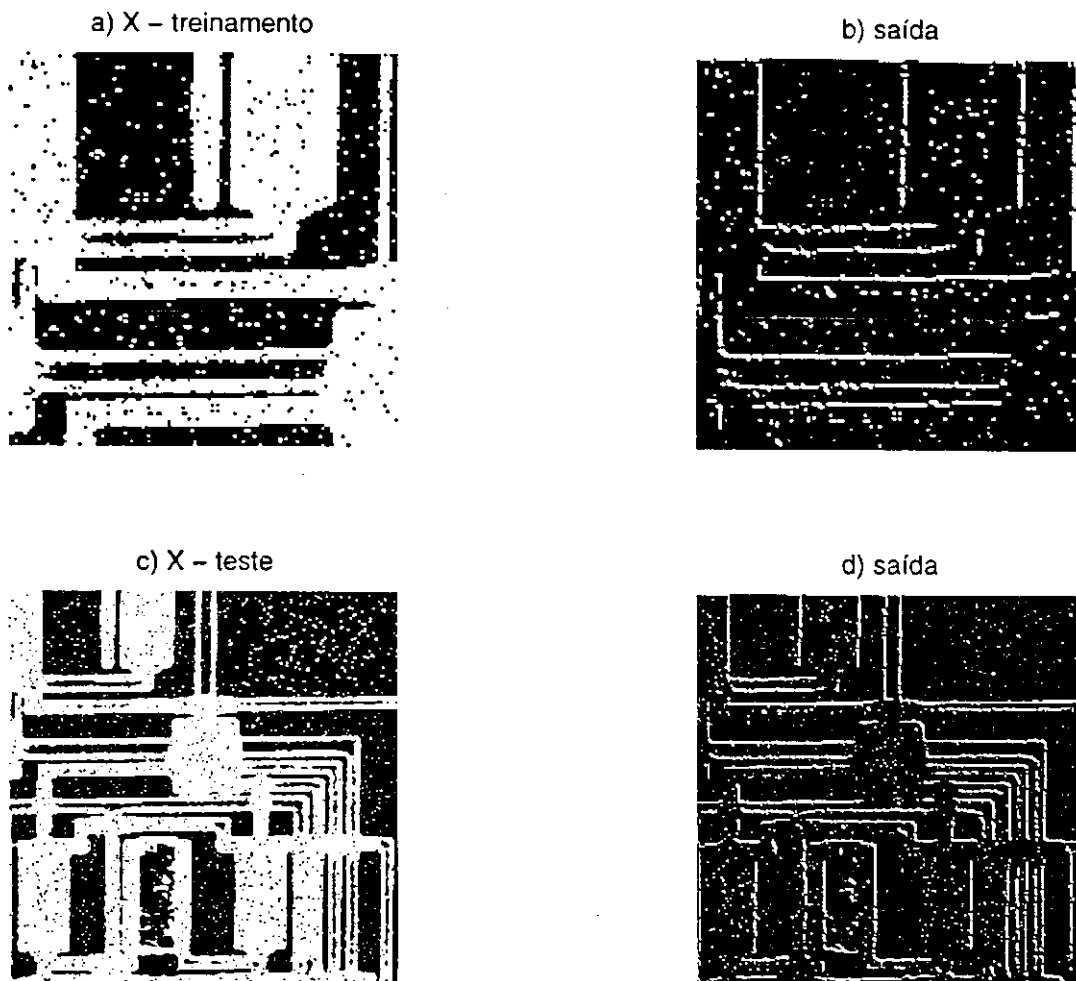


Figura 4.13: Imagens de treinamento da RNMM para o problema de extração de bordas e ruído, decomposição por 8 sup-geradores: (a) imagem binária de entrada, (b) imagem binária desejada, (c) imagens de teste, (d) saída do operador extrator de borda e ruído. Treinamento via RP.

4.6 Considerações Finais

Os resultados obtidos são significativos, mostrando que o treinamento da RNMM com imagens em níveis de cinza via retro-propagação do erro, é mais eficiente que o treinamento da RNMM via AG. Talvez o pior desempenho do filtro projetado com a RNMM via treinamento AG se deva ao fato de se utilizar elementos estruturantes planos para imagens em níveis de cinza. No caso de imagens binárias, os valores se inverteram, mostrando melhor desempenho para o filtro projetado via treinamento da RNMM por algoritmo genético. Isto sugere que as diferenças de desempenho se devem a diferentes tipos de elementos estruturantes usados. Isto é, a RNMM via treinamento retro-propagação do erro usa elementos estruturantes em níveis de cinza, enquanto que a RNMM via treinamento AG usa elementos estruturantes binários. Mostramos também toda a generalidade da decomposição de Banon e Barrera através do exemplo de extração de bordas e ruído de imagem binária. É interessante notar que o treinamento da RNMM via AG teve melhor desempenho que o treinamento via RP para o experimento conjugado.

No próximo capítulo serão apresentadas as conclusões desta tese, ressaltando as contribuições assim como apresentando sugestões para futuras pesquisas.

Capítulo 5

Contribuições Desta Tese

Nesta tese apresentamos uma nova arquitetura de rede neural morfológica, baseada na rede neural modular, a qual denominamos de Rede Neural Modular Morfológica (RNMM). Esta rede resolve o problema de projeto de operadores invariantes à translação, de acordo com o teorema da decomposição de Banon e Barrera [8], e da decomposição de Matheron [2]. De forma original, apresentamos dois algoritmos de treinamento para a RNMM. O primeiro algoritmo de treinamento da RNMM é baseado no algoritmo de retro-propagação do erro (RP) e o segundo é baseado em algoritmos genéticos (AG).

Em sua tese de doutorado, Oliveira [9] se preocupou com a viabilidade da implementação da decomposição de Banon e Barrera via AG. Nesta tese, a nossa preocupação fundamental com relação ao AG foi a sua utilização para treinar a RNMM, para as decomposições via sup-geradores, inf-geradores, erosões e dilatações. Utilizamos imagens binárias e em níveis de cinza nos testes realizados. Comparamos os resultados obtidos com o treinamento da rede via AG com um método clássico e com a nova RNMM treinada via retro-propagação do erro, em um problema de remoção de ruído.

No capítulo 3, o algoritmo de treinamento da RNMM via retro-propagação do erro foi devidamente formalizado, sendo mostrado que a atualização dos pesos da rede (elementos estruturante) é realizada independentemente em cada módulo da rede. O problema do cálculo das derivadas do gradiente foi resolvido utilizando a função in-

-
- [8] G. J. F. Banon and J. Barrera. Minimal representation for translation invariant set mappings by mathematical morphology. *SIAM J. Appl. Math.*, 51(6):1782–1798, 1991.
- [9] João Ricardo de Freitas Oliveira. *O Uso de Algoritmos Genéticos na Decomposição Morfológica de Operadores Invariantes Em Translação Aplicados a Imagens Digitais*. PhD thesis, INPE, 1998.
- [10] Francisco de Assis Tavares Ferreira Da Silva. *Rede Morfológica Não Supervisionada - RMNS*. PhD thesis, Instituto Nacional de Pesquisas Espaciais - INPE, 1998.
- [11] N. J. Nilsson. *Learning Machines: Foundations of Trainable Pattern Classifying Systems*. McGraw Hill, New York, 1965.
- [12] James McClelland and David Rumelhart. *Parallel distributed Processing*. MIT Press, Cambridge - MA, 1986.
- [13] Jennifer L. Davidson and Frank Hummer. Morphology neural networks: An introduction with applications. *Circuits System Signal Process*, 12(2):179–210, 1993.
- [14] G. X. Ritter. Recent developments in image algebra. In *Advance in Eletronics and Eletron Physics*, volume 80. Academic Press, 1991.
- [15] Christoph B. Herwing and Robert J. Shalkoff. Morphological image processing using artificial neural networks. In C. T. Leondes, editor, *Control and Dynamic Systems, Vol. 67*, pages 319–379. Academic Press, 1994.
- [16] Bernad Widrow and Michael A. Lehr. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.
- [17] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, New Jersey, 1998.
- [18] Michael I. Jordan and Robert A. Jacobs. Hierarchies of adaptive experts. In Steve J. Hanson John E. Moody and Richard P. Lippmann, editors, *Neural Information*

Processing Systems, volume 4, chapter XIII, pages 985–992. Morgan Kaufmann Publishers, 1992.

- [19] Eric Menu. *Approche Modulaire des Réseaux de Neurones en Fusion d'informations*. PhD thesis, Université de Paris-SUD - Centre d'Orsay, 1994.
- [20] Lucio F. C. Pessoa and Petros Maragos. MRL-filters: A general class of nonlinear systems and their optimal design for image processing. *IEEE Transactions on Image Processing*, 7:966–978, 1998.
- [21] Lúcio F. C. Pessoa. *Nonlinear Systems and Neural Networks with Hybrid Morphological/Rank/Linear Nodes: Optimal Design and Applications to Image and Pattern Recognition*. PhD thesis, Georgia Institute of Technology, 1997.
- [22] Lúcio F. C. Pessoa and Petros Maragos. Morphological rank neural networks and their adaptive optimal design for image processing. In *Proc. of the 1996 IEEE Intl Conference on Acoustics, Speech and Signal Processing*, Atlanta, 1996. IEEE.
- [23] J. Serra. *Image Analysis And Mathematical Morphology*. Academic Press, London, 1982.
- [24] S. R. Sternberg. Cellular computers and biomedical image processing. In J. Sklansky and J. C. Bisconte, editors, *Lecture Notes in Medical Information, Proc. on Biomedical Images and Computers*, volume 17, pages 274–319. Springer- Verlag, Berlin, 1980.
- [25] Henk J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.
- [26] Petros Maragos. A representation theory for morphological image and signal processing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:586–599, 1989.
- [27] Gerald Jean Francis Banon and Junior Barrera. Decomposição of mappings between complete lattices by mathematical morphology, part i. general lattices. *Signal Processing*, 30:299–327, 1993.

- [28] Gerald J. F. Banon e Junior Barrera. *Bases da Morfologia Matemática Para Análise de Imagens Binárias*. MCT/INPE, São José dos Campos - SP, segunda edição, 1998.
- [29] H. Minkowski. *Gesammelte Abhandlungen*. Teubner Verlag, Leipzig-Berlin, 1911.
- [30] Gerald Jean Francis Banon. *Formal Introduction to Digital Image Processing*. INPE-6969-PUD/034, São José dos Campos - SP, 1998.
- [31] Tom M. Apostol. *Cálculo*. Editorial Reverté, S. A., Rio de Janeiro, 1979.
- [32] Ping Xue Joseph B. Evans and Bede Liu. Analysis and implementation of variable step size adaptative algorithms. *IEEE Transactions On Signal Processing*, 41(8):2517–2535, August 1993.
- [33] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, New York, 1992.
- [34] Robson Pequeno de Sousa. Classificação automática de ostracodes utilizando atributos qualitativos. Dissertação de mestrado, UFPE - DEE, Recife - Pe, 1991.
- [35] Proc. 11th Int. Conf. Artif. Intell. *Training Feedforward Networks Using Genetic Algorithms*, Detroit, 1989.
- [36] S. A. Hard and T. Samad. Genetic synthesis of neural networks architecture. In Lawrence Davis, editor, *Handbook of Genetic Algorithms*, pages 202–221. Van Nostrand Reinhold, New York, 1991.
- [37] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan, 1975.
- [38] Chin-Teng Lin and C.S. George Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall, London, 1996.
- [39] Samir W. Mahfoud and David E. Goldberg. Parallel recombinative simulated annealing. *In Parallel Computing*, 21:1–28, 1995.

2. Referencial Teórico

Ao trabalharmos com um pequeno aspecto de um dos principais elementos que constituem a identidade étnica Fulni-ô, o Tolê em seu aspecto sonoro, procuraremos entender como se realiza a construção conceitual de um grupo étnico bem como o conceito de identidade étnica trabalhados na antropologia. Com estes conceitos em mãos, traremos os trabalhos de alguns antropólogos entre os índios no Nordeste, para entendermos onde queríamos chegar ao utilizar a antropologia como base para nossa percepção do que vêm a ser as facetas musicais como elemento adscritivo dos Fulni-ô. Ainda, neste capítulo, traremos algumas tendências teóricas da etnomusicologia que possam, também, nos ajudar frente aos problemas existentes na observação das manifestações sonoras musicais Fulni-ô.

2.1. Contribuições Antropológica

De acordo com Brandão (1986, 145) um grupo étnico consiste de

Uma categoria de articulação de tipos de pessoas que, por estarem historicamente unidas por laços próprios de relações realizadas como famílias, redes de parentes, clãs, metades, aldeias e tribos, e por viverem e se reconhecerem vivendo em comum um mesmo modo peculiar de vida e representação de vida social, estabelecem para eles próprios e para os outros as suas fronteiras étnicas, os seus limites de etnia. Mergulhados em um sistema de relações regidas pela desigualdade aprendem a pensar a diferença; aprendem a se pensar como diferentes.

Ao estudarmos um grupo étnico, nada mais oportuno trazer para pauta as características que dão caminhos à compreensão do mesmo, se não geradoras ou não, para a compreensão de um grupo étnico. Ao citar uma determinada compreensão dele a partir de uma observação etnológica. Ao citar uma determinada