



Universidade Estadual de Campinas

Instituto de Matemática, Estatística
e Computação Científica

DEPARTAMENTO DE MATEMÁTICA APLICADA



Modelos Modificados de Redes Neurais Morfológicas

Autor: Estevão Esmi Laureano

Orientador: Prof. Dr. Peter Sussner

16 de junho de 2010

Este trabalho teve o apoio financeiro da Fapesp. Processo 2006/05868-5.



Modelos Modificados de Redes Neurais Morfológicas

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Estevão Esmi Laureano e aprovada pela comissão julgadora.

Campinas, 30 de Abril de 2010



Prof. Dr. Peter Sussner.

Orientador

Banca Examinadora:

1. Álvaro Rodolfo De Pierro, Dr. DMA / IMECC / Unicamp
2. Fernando José Von Zuben, Dr. DCA / FEEC / Unicamp
3. Marcos Eduardo Ribeiro do Valle Mesquita, Dr. DM / CCE / UEL
4. Peter Sussner, Dr. DMA / IMECC / Unicamp

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Matemática Aplicada.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Laureano, Estevão Esmi
L373m Modelos modificados de redes neurais morfológicas/Estevão Esmi
Laureano -- Campinas, [S.P. : s.n.], 2010.

Orientador : Peter Sussner

Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Matemática, Estatística e Computação Científica.

1.Teoria dos reticulados. 2.Morfologia matemática. 3.Conjuntos difusos. 4.Redes neurais (Computação). 5.Memória associativa. 6.Reconhecimento de padrões . I. Sussner, Peter. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Título em inglês: Modified models of morphological neural networks

Palavras-chave em inglês (Keywords): 1.Lattice theory. 2.Mathematical morphology. 3.Fuzzy sets. 4.Neural networks (Computer science). 5.Associative memory. 6.Pattern recognition.

Área de concentração: Matemática Aplicada

Titulação: Mestre em Matemática Aplicada

Banca examinadora: Prof. Dr. Peter Sussner (IMECC-UNICAMP)
Prof. Dr. Álvaro Rodolfo de Pierro (IMECC-UNICAMP)
Prof. Dr. Fernando José Von Zuben (FEEC-UNICAMP)
Prof. Dr. Marcos Eduardo Ribeiro do Valle Mesquita (CCE-UEL)

Data da defesa: 26/04/2010

Programa de Pós-Graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 26 de abril de 2010 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



Prof.(a). Dr(a). PETER SUSSNER



Prof. (a). Dr (a). ALVARO RODOLFO DE PIERRO



Prof. (a). Dr (a). MARCOS EDUARDO RIBEIRO DO VALLE MESQUITA



Prof. (a). Dr (a). FERNANDO JOSÉ VON ZUBEN

Resumo

Redes neurais morfológicas (MNN) são redes neurais artificiais cujos nós executam operações elementares da morfologia matemática (MM). Vários modelos de MNNs e seus respectivos algoritmos de treinamentos têm sido propostos nos últimos anos, incluindo os perceptrons morfológicos (MPs), o perceptron morfológico com dendritos, as memórias associativas morfológicas (fuzzy), as redes neurais morfológicas modulares e as redes neurais de pesos compartilhados e regularizados. Aplicações de MNNs incluem reconhecimento de padrão, previsão de séries temporais, detecção de alvos, auto-localização e processamento de imagens hiperespectrais.

Nesta tese, abordamos dois novos modelos de redes neurais morfológicas. O primeiro consiste em uma memória associativa fuzzy denominada KS-FAM, e o segundo representa uma nova versão do perceptron morfológico para problemas de classificação de múltiplas classes, denominado perceptron morfológico com aprendizagem competitiva (MP/CL).

Para ambos modelos, investigamos e demonstramos várias propriedades. Em particular para a KS-FAM, caracterizamos as condições para que uma memória seja perfeitamente recordada, assim como a forma da saída produzida ao apresentar um padrão de entrada qualquer. Provamos ainda que o algoritmo de treinamento do MP/CL converge em um número finito de passos e que a rede produzida independe da ordem com que os padrões de treinamento são apresentados. Além disso, é garantido que o MP/CL resultante classifica perfeitamente todos os dados de treinamento e não produz regiões de indecisões.

Finalmente, comparamos os desempenhos destes modelos com os de outros modelos similares em uma série de experimentos, que inclui reconhecimento de imagens em tons de cinza, para a KS-FAM, e classificação de vários conjuntos de dados disponíveis na internet, para o MP/CL.

Palavras-chave: Reticulado, Morfologia Matemática, Conjuntos Fuzzy, Rede Neural (Morfológica), Memória Associativa (Fuzzy), Reconhecimento de Padrão.

Abstract

Morphological neural networks (MNN) are artificial neural networks whose hidden neurons perform elementary operations of mathematical morphology (MM). Several particular models of MNNs have been proposed in recent years, including morphological perceptrons (MPs), morphological perceptrons with dendrites, (fuzzy) morphological associative memories, modular morphological neural networks as well as morphological shared-weight and regularization neural networks. Applications of MNNs include pattern recognition, time series prediction, target detection, self-location, and hyper-spectral image processing.

In this thesis, we present two new models of morphological neural networks. The first one consists of a fuzzy associative memory called KS-FAM. The second one represents a novel version of the morphological perceptron for classification problems with multiple classes called morphological perceptron with competitive learning (MP/CL).

For both KS-FAM and MP/CL models, we investigated and showed several properties. In particular, we characterized the conditions for perfect recall using the KS-FAM as well as the outputs produced upon presentation of an arbitrary input pattern. In addition, we proved that the learning algorithm of the MP/CL converges in a finite number of steps and that the results produced after the conclusion of the training phase do not depend on the order in which the training patterns are presented to the network. Moreover, the MP/CL is guaranteed to perfectly classify all training data without generating any regions of indecision.

Finally, we compared the performances of our new models and a range of competing models in terms of a series of experiments in gray-scale image recognition (in case of the KS-FAM) and classification using several well-known datasets that are available on the internet (in case of the MP/CL).

Keywords: Lattice Theory, Mathematical Morphology, Fuzzy Set, (Morphological) Neural Network, (Fuzzy) Associative Memory, Pattern Recognition.

*Aos meu pais,
Brasil Alves Laureano e,
Tânia Esmi Laureano*

Agradecimentos

À todos que me apoiaram e me ajudaram durante este projeto de mestrado. Em especial, para meus pais, Brasil Alves Laureano e Tânia Esmi Laureano, a minha irmã Renata Esmi Laureano, e aos demais entes queridos que tanto torceram por mim.

À Damaris, por todos os momentos de compreensão, carinho e amor.

Também aos meus grandes amigos que tiveram comigo durante este tempo: Willian Eloi, Gustavo, Marcio, Marcos e Luciana, Willian Oliveira, Nolmar, Celso, Ney e Flávia, Eduardo Miqueles, José Carlos e vários outros.

Ao meu orientador, Peter Sussner, pela orientação, paciência e, principalmente, por acreditar em minha capacidade para desenvolver este projeto.

Agradeço por fim, todos professores tanto da Unicamp quanto da Unip que contribuíram para minha formação acadêmica nos cursos de pós-graduação e bacharelado.

Tabela de siglas

<i>Sigla</i>	<i>Significado</i>
MNN	Redes neurais morfológicas
MP	Perceptron morfológico
MPD	Perceptron morfológico com dendritos
MM	Morfologia matemática
FMNN	Redes neural morfológica fuzzy
FMAM	Memória associativa morfológica fuzzy
MP/CL	Perceptron morfológico com aprendizagem competitiva
MLP	Perceptron de múltiplas camadas
kNN	k vizinhos mais próximos
ANN	k Redes neural artificial
MAM	Memória associativa morfológica
SE	Elemento estruturante
FMM	Morfologia matemática fuzzy
AM	Memória associativa
FAM	Memória associativa fuzzy
AMM	Memória morfológica auto-associativa
HMM	Memória morfológica heteroassociativa
IFAM	Memória associativa fuzzy implicativa
GFAM	Memória associativa fuzzy generalizada
KS-FAM	Memória associativa fuzzy baseada na medida de subsethood de Kosko
NRMSE	Raiz quadrada do erro médio quadrático normalizado

DAM	Memória associativa dinâmica
BAM	Memória associativa bidirecional
KAM	“Kernel associative memory”
BSB	“Brain-state-in-a-box”
OLAM	Memória associativa linear óptima
FLN	Neurocomputação em reticulados fuzzy
FLNN	“Fuzzy lattice neural network”
FLR	“Fuzzy lattice reasoning”
MSE	Erro médio quadrático

Sumário

1	Introdução	1
2	Fundamentos teóricos das redes neurais morfológicas	5
2.1	Estrutura de reticulados completos na morfologia matemática . . .	7
2.2	Motivação e conceitos básicos da morfologia matemática	18
2.3	Conceitos básicos da morfologia matemática fuzzy	20
3	Memórias associativas morfológicas	25
3.1	Modelos clássicos de memórias associativas morfológicas	26
3.2	Memórias associativas morfológicas fuzzy	28
3.3	FAM baseada na medida de <i>subsethood</i> de Kosko (KS-FAM) . . .	31
3.3.1	Medida de <i>subsethood</i> de Kosko	31
3.3.2	FAM baseada na medida de <i>subsethood</i> de Kosko para o caso binário	34
3.3.3	KS-FAM: Definição e propriedades	35
3.4	Experimentos com KS-FAM	41
3.4.1	Imagens binárias	42
3.4.2	Imagens em tons de cinza	43
3.4.2.1	Experimentos usando padrões incompletos	45
3.4.2.2	Variações de claridade e orientação	46
3.4.2.3	Padrões ruidosos	48
4	Redes neurais morfológicas construtivas	53
4.1	Neurocomputação em reticulados fuzzy	53
4.2	Perceptron morfológico	57
4.3	Perceptron morfológico com aprendizagem competitiva (MP/CL)	60
4.3.1	Algoritmo do MP/CL para problemas de classificação binários	63
4.3.2	Propriedades do algoritmo do MP/CL para problemas de classificação binários	67

4.3.3	Algoritmo do MP/CL para problemas de classificação com múltiplas classes	81
4.4	Experimentos com redes neurais morfológicas construtivas	82
4.4.1	Problema sintético de Ripley	85
4.4.2	Problema de segmentação de imagem	87
4.4.3	Problema de classificação de flores íris	88
4.4.4	Problema de câncer de mama de Wisconsin	90
5	Conclusões e considerações finais	93

Introdução

Redes neurais morfológicas (MNNs) surgiram da fusão de idéias da morfologia matemática e redes neurais artificiais. Vários modelos particulares de MNNs e seus respectivos algoritmos de treinamentos foram propostos nos últimos anos, incluindo os perceptrons morfológicos (MPs) [72, 83], o perceptron morfológico com dendritos (MPDs) [74], as memórias associativas morfológicas (fuzzy) [92, 93, 95, 99], as redes neurais morfológicas modulares [1] e as redes neurais de pesos compartilhados e regularizadas [37, 45]. Os modelos de neuro-computação fuzzy desenvolvidos por Kaburlasos, Petridis et al., também podem ser classificados como redes neurais morfológicas, mesmo tendo sido baseados e inspirados em outras idéias, tais como as provindas dos modelos ART [11, 12, 13]. Redes neurais híbridas “morphological-rank-linear” e morfológicas têm sido aplicadas com sucesso em uma variedade de problemas, tais como reconhecimento de padrão [40, 42, 43, 46, 63], previsão de séries temporais [2, 44, 95], detecção de alvos [46], reconhecimento de caracteres [62], auto-localização e análise de imagem hiperespectral [30, 66].

O perceptron morfológico está entre os primeiros modelos de redes neurais morfológicas que apareceram na literatura [69, 72, 83]. Este modelo foi baseado na álgebra minimax, que corresponde a uma álgebra de reticulado originada para problemas em escalonamento de máquinas e pesquisa operacional [4, 14, 19]. A álgebra minimax possui algumas semelhanças com a álgebra linear, por exemplo, ela dispõe de produtos de matrizes e lida com estruturas matemáticas que recordam espaços vetoriais e anéis [18]. O fato dos cálculos ocorridos na ca-

mada de uma rede neural tradicional poderem ser expressos como um produto matriz-vetor, seguido da aplicação de uma função de ativação não linear, inspirou pesquisadores a desenvolver MNNs substituindo os produtos matriciais convencionais pelos definidos na álgebra minimax [21, 28, 29]. J.D. Davidson mostrou que a teoria da morfologia matemática clássica pode ser encaixada na álgebra minimax, por meio de um isomorfismo que mapeia certos produtos matriciais em operações elementares da morfologia matemática [20]. Dessa maneira, para este tipo de rede, a álgebra minimax tem concedido aos pesquisadores um acesso fácil ao problema de definir pesos que, geralmente, na terminologia da morfologia matemática, correspondem a certos elementos estruturantes.

A *morfologia matemática* (MM), por sua vez, é uma teoria que usa conceitos das teorias de conjuntos, geometria e topologia para análise de estruturas geométricas em objetos [36, 58, 76, 77] e tem sido utilizada em várias áreas, tais como processamento de imagens, reconhecimento de padrões e visão computacional [9, 32, 33, 47, 64, 78, 79]. Inicialmente, essa teoria surgiu para processamento de imagens binárias [58, 76] e, subsequentemente, várias abordagens foram propostas para generalizá-la para imagens em tons de cinza, tais como as abordagens a *umbra* [76, 80] e *morfologia matemática fuzzy* [23, 60]. Está última, consiste de uma abordagem à MM a partir da teoria de conjuntos fuzzy [23, 60, 96, 101]. Em geral, a estrutura de reticulados completo é a estrutura amplamente aceita para se conduzir à MM [36, 75].

A MM contempla quatro operadores elementares: erosão, dilatação, anti-erosão e anti-dilatação. Posto isto, uma rede neural morfológica é definida, mais precisamente, como sendo uma rede neural artificial cujos nós executam uma das quatro operações elementares da MM, possivelmente seguidos de uma função de ativação [99].

Uma classe de MNNs de especial importância para este trabalho é constituído pelas *redes neurais morfológicas fuzzy* (FMNNs), que consistem em MNNs baseadas na morfologia matemática fuzzy. A primeira classe geral de MNN fuzzy que surgiu na literatura corresponde às memórias associativas morfológicas fuzzy (FMAMs) [99]. Em termos gerais, uma *memória associativa* é um modelo projetado para armazenar pares de entradas e saídas, que adicionalmente exhibe uma certa tolerância a entradas distorcidas ou incompletas [34, 49]. No nosso contexto, uma memória associativa morfológica (fuzzy) é uma memória associativa que, ao mesmo tempo, também é uma MNN (fuzzy).

Nesta tese, abordamos dois novos modelos de redes neurais morfológicas. O primeiro consiste em uma memória associativa fuzzy denominada KS-FAM. O segundo representa uma versão do perceptron morfológico para problemas de classificação de múltiplas classes, denominada perceptron morfológico com aprendizagem competitiva (MP/CL). O último modelo pertence a uma subclasse das MNNs denominada “redes neurais morfológicas construtivas”, cujo nome vem do fato de que, durante a fase de aprendizagem, os algoritmos de treinamentos dessas redes constroem e ajustam automaticamente sua arquitetura para lidar com um determinado problema.

Tanto para KS-FAM quanto para o MP/CL, investigamos e demonstramos várias propriedades, por exemplo, mostramos que o algoritmo de treinamento do MP/CL independe da ordem com que os padrões de treinamento são apresentados à rede. Adicionalmente, comparamos os desempenhos destes modelos com os de outros modelos similares em uma série de experimentos. Como fruto da pesquisa desenvolvidas sobre os dois modelos, o autor e seu orientador, em conjunto, publicaram dois artigos em anais de conferência [89, 87] e um capítulo de livro [88]. Além disso, eles também tiveram um artigo aceito para publicação em uma revista internacional com árbitro [91], e mais dois artigos aceitos para publicação em proceedings de dois congressos internacionais [25, 90].

Esta tese é organizada em cinco capítulos.

No Capítulo 2, apresentaremos todos os conceitos matemáticos fundamentais das redes neurais morfológicas que serão abordadas no decorrer deste trabalho. Nele, distinguiremos entre as abordagens à morfologia matemática no sentido algébrico (Seção 2.1) e no sentido intuitivo (Seção 2.2). Adicionalmente, indicaremos os conceitos básicos da abordagem fuzzy à MM na Seção 2.3.

No Capítulo 3, abordaremos a classe das memórias associativas morfológicas. Inicialmente apresentando os primeiros modelos de MAMs que surgiram na literatura e, em seguida, abordando a classe das memórias associativas morfológicas fuzzy, onde foi revista uma estratégia geral para treinamento de certas FMAMs, baseada na relação de adjunção entre operadores da MM sobre estrutura de reticulados completos. Na terceira seção, introduziremos o modelo KS-FAM, demonstrando algumas das suas propriedades. Por fim, encerraremos este capítulo comparando a KS-FAM com outros modelos de memórias associativas em uma série de experimentos, envolvendo tanto imagens binárias quanto imagens em tons de cinza. Nestas simulações, sondamos a capacidade de tolerância da

KS-FAM com respeito à entradas incompletas e presença de ruídos.

No Capítulo 4, trataremos de maneira geral dos modelos de redes neurais morfológicas construtivas, tais como os modelos FLNs de Kaburlasos et al. Contudo, uma atenção especial será dada ao modelo MP/CL e seu respectivo algoritmo de treinamento, pois, boa parte do texto deste capítulo será dedicado à investigação e demonstração de várias propriedades exibidas pelo algoritmo de treinamento do MP/CL. Também vamos conduzir testes e comparações quanto ao desempenho do MP/CL em quatro problemas bem conhecidos de classificação, disponíveis na internet, com vários classificadores, incluindo o tradicional perceptron de múltiplas camadas (MLP), árvore de decisão, k vizinhos mais próximos (kNN) e outros classificadores morfológicos.

Por último, no Capítulo 5, apresentaremos as conclusões e as considerações finais.

Fundamentos teóricos das redes neurais morfológicas

As redes neurais morfológicas (MNNs) constituem uma classe de redes neurais artificiais (ANNs), cujo elemento de processamento (ou neurônio) executa uma operação elementar da morfologia matemática (erosão, dilatação, anti-erosão ou anti-dilatação) seguida, possivelmente, de uma função de ativação [97].

Os fundamentos matemáticos das redes neurais morfológicas podem ser encontrados na teoria da morfologia matemática (MM) junto à estrutura de reticulados completos [70, 71]. As primeiras redes neurais morfológicas, como as memórias associativas morfológicas (MAMs) e o perceptron morfológico (MP), foram definidas por operadores de uma sub-álgebra da álgebra de imagens, especificamente a álgebra minimax, que consiste em um caso particular da morfologia matemática [20, 21, 73]. Entretanto, a partir da caracterização algébrica dos operadores fundamentais da MM, foi verificado que cada unidade de processamento dessas redes executavam operações da MM.

A morfologia matemática surgiu inicialmente com a morfologia matemática binária, como uma ferramenta para a análise de imagens binárias [56, 57, 58, 76, 77]. Posteriormente, várias abordagens estenderam a teoria inicial para imagens em tons de cinza [36, 76, 81, 75].

A teoria da morfologia matemática inclui dois operadores básicos: a erosão e a dilatação. Estes operadores são definidos em termos de um elemento estruturante. Intuitivamente, uma erosão calcula o grau de inclusão do elemento estruturante

numa imagem dada nos pontos do domínio, enquanto, uma dilatação calcula o grau de intersecção do elemento estruturante refletido pela origem com a imagem dada em cada ponto do domínio. Além da erosão e da dilatação, há ainda dois outros operadores elementares, a anti-dilatação e a anti-erosão, que são produzidos pela combinação de um conceito de negação com a erosão ou dilatação. A Figura 2.1 exibe exemplos de uma erosão e uma dilatação de imagem binária por um elemento estruturante dado por um disco binário de raio igual a 5. Similarmente, a Figura 2.2 exibe exemplos de uma erosão fuzzy e uma dilatação fuzzy de uma imagem fuzzy, i.e., um conjunto fuzzy que identifica uma imagem em tons de cinza.

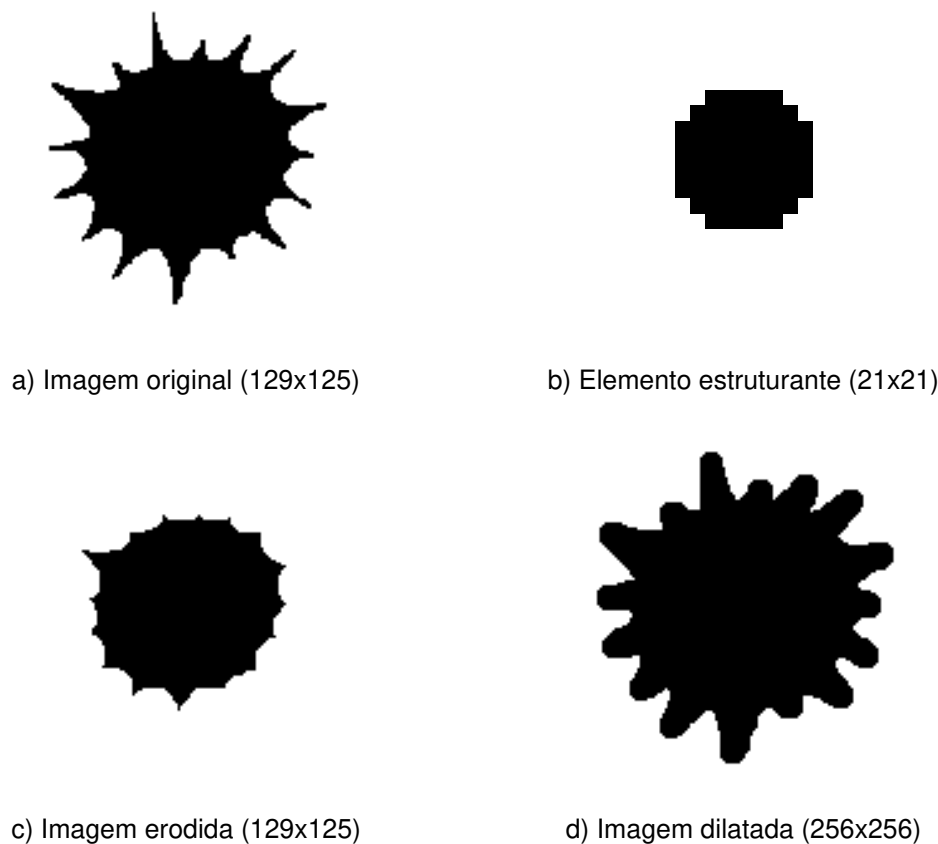
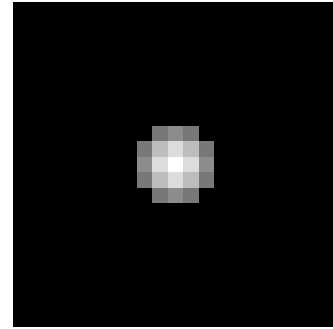


Figura 2.1: (a) Imagem original binária, (b) Elemento estruturante dado por um disco com raio 5, (c) Imagem original erodida pelo elemento estruturante, (d) Imagem original dilatada pelo elemento estruturante.

Além do sentido intuitivo, os operadores elementares da MM podem ser car-



a) Imagem original (256x256)



b) Elemento estruturante (21x21)



c) Imagem erodida (256x256)



d) Imagem dilatada (256x256)

Figura 2.2: (a) Imagem original fuzzy, (b) Elemento de estruturação dado por uma bola de raio 3, (c) Imagem original erodida pelo elemento estruturante, (d) Imagem original dilatada pelo elemento estruturante.

acterizados, equivalentemente, através de um sentido algébrico baseado na teoria de reticulados. Na Seção 2.1 será apresentada a generalização da morfologia matemática em estrutura de reticulados completos a partir das propriedades algébricas de seus operadores.

2.1 Estrutura de reticulados completos na morfologia matemática

A teoria de reticulados se refere a estruturas algébricas que impõem algum tipo de ordenação sobre um conjunto [7, 36, 55, 75]. As fundações matemáticas

da forma mais geral da MM podem ser encontradas nesta teoria [36, 75, 77].

Um conjunto *parcialmente ordenado* \mathbb{L} (poset) é um conjunto tal que, uma relação binária reflexiva, antisimétrica e transitiva “ \leq ” é definida. Se para todo $x, y \in \mathbb{L}$ tem-se $x \leq y$ ou $y \leq x$ então \mathbb{L} é dito uma *corrente*. Neste trabalho, vamos supor, para simplificar, que um poset não é vazio [31].

Um conjunto parcialmente ordenado \mathbb{L} é denominado *reticulado* se, e somente se, todo subconjunto finito e não vazio de \mathbb{L} possuir ínfimo e supremo em \mathbb{L} [7, 55]. O ínfimo do subconjunto $Y \subseteq \mathbb{L}$ é denotado pelo símbolo $\bigwedge Y$ ou, alternativamente, por $\bigwedge_{j \in J} y_j$ para $Y = \{y_j : j \in J\}$ e J um conjunto de índices. Similarmente, o supremo do subconjunto $Y \subseteq \mathbb{L}$ é denotado pelo símbolo $\bigvee Y$ ou, alternativamente, por $\bigvee_{j \in J} y_j$ para $Y = \{y_j : j \in J\}$ e J um conjunto de índices.

Um *reticulado limitado* \mathbb{L} possui um elemento mínimo denotado por $0_{\mathbb{L}}$ (ou alternativamente por $-\infty$) e um elemento máximo por $1_{\mathbb{L}}$ (ou alternativamente por $+\infty$). Em um reticulado limitado definimos $\bigwedge \emptyset = 1_{\mathbb{L}}$ e $\bigvee \emptyset = 0_{\mathbb{L}}$. Note que todo reticulado \mathbb{L} pode ser convertido em um reticulado limitado $\bar{\mathbb{L}}$ pela adição de um elemento mínimo $0_{\mathbb{L}}$ e um elemento de máximo $1_{\mathbb{L}}$.

O conjunto \mathbb{L} é dito *reticulado completo* se todo subconjunto (finito ou infinito) de \mathbb{L} possui um ínfimo e um supremo em \mathbb{L} , o intervalo $[0, 1]$ e o conjunto $\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{+\infty, -\infty\}$ representam exemplos de reticulado completo [7, 55]. Note que todo reticulado completo \mathbb{L} é limitado. Se um reticulado \mathbb{L} possui a propriedade de que todo subconjunto não vazio e limitado possui um ínfimo e um supremo em \mathbb{L} , então \mathbb{L} é *condicionalmente completo*. Por exemplo, o conjunto dos reais \mathbb{R} é condicionalmente completo, mas o conjunto dos números racionais \mathbb{Q} não é condicionalmente completo.

Lema 1. *Seja \mathbb{L} um reticulado condicionalmente completo, Se \mathbb{L} for limitado, então \mathbb{L} é um reticulado completo.*

Demonstração. Seja $Y \subseteq \mathbb{L}$ tal que $Y \neq \emptyset$. Temos que Y é limitado pois $0_{\mathbb{L}} \leq y \leq 1_{\mathbb{L}}$ para todo $y \in Y$. Como \mathbb{L} é condicionalmente completo, temos que existem $\bigwedge Y \in \mathbb{L}$ e $\bigvee Y \in \mathbb{L}$. Logo, \mathbb{L} é reticulado completo \square

Sejam $\mathbb{L}_1, \dots, \mathbb{L}_n$ reticulados, então, uma ordem parcial sobre $\mathbb{L} = \mathbb{L}_1 \times \dots \times \mathbb{L}_n$ pode ser definida de acordo com a Eq. 2.1.

$$(x_1, \dots, x_n) \leq (y_1, \dots, y_n) \Leftrightarrow x_i \leq y_i, \quad \forall i = 1, \dots, n \quad (2.1)$$

O poset \mathbb{L} resultante é também um reticulado e é chamado de um *produto de reticulados* com constituintes $\mathbb{L}_1, \dots, \mathbb{L}_n$. O símbolo \mathbb{L}^n é utilizado para denotar o produto de n cópias de \mathbb{L} . Além disso, se \mathbb{L}_i é reticulado completo para $i = 1, \dots, n$, então \mathbb{L} herda a propriedade de completude se as seguintes regras para qualquer conjunto de índice J forem consideradas:

$$\bigwedge_{j \in J} \mathbf{x}^j = \bigwedge_{j \in J} (\mathbf{x}_1^j, \dots, \mathbf{x}_n^j) = (\bigwedge_{j \in J} \mathbf{x}_1^j, \dots, \bigwedge_{j \in J} \mathbf{x}_n^j), \quad (2.2)$$

$$\bigvee_{j \in J} \mathbf{x}^j = \bigvee_{j \in J} (\mathbf{x}_1^j, \dots, \mathbf{x}_n^j) = (\bigvee_{j \in J} \mathbf{x}_1^j, \dots, \bigvee_{j \in J} \mathbf{x}_n^j). \quad (2.3)$$

Para \mathbf{a} e \mathbf{b} pertencentes a um reticulado \mathbb{L} , o intervalo $[\mathbf{a}, \mathbf{b}]$ denota o conjunto $\{\mathbf{x} \in \mathbb{L} : \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\}$ correspondente à hipercaixa com vértice inferior \mathbf{a} e vértice superior \mathbf{b} . Note que, se $\mathbf{a} > \mathbf{b}$, então o conjunto $[\mathbf{a}, \mathbf{b}] = \emptyset$. Para $X \subseteq \mathbb{L}$, a notação $\text{box}(X)$ é utilizada para denotar o menor intervalo ou hipercaixa que contém X . Se $X = \emptyset$, então, $\text{box}(X) = \emptyset$, caso contrário, $\text{box}(X)$ é dado por $[\bigwedge X, \bigvee X]$. Se $\mathcal{H}(\mathbb{L})$ denota o conjunto de todas as hipercaixas no reticulado completo \mathbb{L} , então, $\mathcal{H}(\mathbb{L})$ forma um reticulado completo com a relação de ordem parcial dada por \subseteq , onde o maior e o menor elemento são \emptyset e \mathbb{L} , respectivamente.

O conjunto $\tau(\mathbb{L}) = \{(\mathbf{a}, \mathbf{b}); \mathbf{a}, \mathbf{b} \in \mathbb{L}\}$ com a ordem parcial dada por $(\mathbf{a}, \mathbf{b}) \leq (\mathbf{c}, \mathbf{d}) \Leftrightarrow \mathbf{c} \leq_{\mathbb{L}} \mathbf{a}$ e $\mathbf{b} \leq_{\mathbb{L}} \mathbf{d}$, onde $\leq_{\mathbb{L}}$ denota a ordem parcial do reticulado \mathbb{L} , também representa um reticulado. Além disso, se \mathbb{L} é completo, então $\tau(\mathbb{L})$ herda a propriedade de completude de \mathbb{L} . Os elementos de $\tau(\mathbb{L})$ são chamados de intervalos generalizados. Aqui, preferimos denotar por (\mathbf{a}, \mathbf{b}) ao invés de $[\mathbf{a}, \mathbf{b}]$ para distinguir das hipercaixas, porém, note que se $(\mathbf{a}, \mathbf{b}) \in \tau(\mathbb{L})$ for tal que $\mathbf{a} \leq \mathbf{b}$, então, podemos identificar (\mathbf{a}, \mathbf{b}) com uma única hipercaixa $[\mathbf{a}, \mathbf{b}] \neq \emptyset$.

O conjunto das funções de um conjunto \mathcal{U} em $[0, 1]$, denotado por $[0, 1]^{\mathcal{U}}$ (ou alternativamente por $\mathcal{F}(\mathcal{U})$), herda a estrutura de reticulado completo de $[0, 1]$ em termos da seguinte ordem parcial: Para todo $\mathbf{a}, \mathbf{b} \in [0, 1]^{\mathcal{U}}$, temos que

$$\mathbf{a} \leq \mathbf{b} \Leftrightarrow \mathbf{a}(u) \leq \mathbf{b}(u), \quad \forall u \in \mathcal{U}. \quad (2.4)$$

Sejam \mathbb{L} e \mathbb{M} reticulados, uma função $\varphi : \mathbb{L} \rightarrow \mathbb{M}$, que satisfaz às seguintes equações para todo $\mathbf{x} \in \mathbb{L}$ e todo $\mathbf{y} \in \mathbb{M}$, é chamada *homomorfismo de reticulados*:

$$\varphi(x \vee y) = \varphi(x) \vee \varphi(y) \quad \text{e} \quad \varphi(x \wedge y) = \varphi(x) \wedge \varphi(y). \quad (2.5)$$

Um homomorfismo de reticulado bijetivo é dito um *isomorfismo de reticulado*. Equivalentemente, a função $\varphi : \mathbb{L} \rightarrow \mathbb{M}$ é um isomorfismo de reticulado se φ é bijetiva e *isótona*, isto é, $\varphi(x) \leq \varphi(y)$ para todo $x \leq y$. No caso especial onde $\mathbb{L} = \mathbb{M}$, temos um *automorfismo de reticulado*.

A função $\nu : \mathbb{L} \rightarrow \mathbb{L}$ é chamada de um automorfismo dual se ela inverte a ordem parcial. Em particular, um automorfismo dual involutivo $\nu : \mathbb{L} \rightarrow \mathbb{L}$ representa uma *negação* em \mathbb{L} . Em outras palavras, uma negação em \mathbb{L} é uma bijeção involutiva que inverte a ordem parcial de \mathbb{L} [36]. Por exemplo, $N_S(x) = 1 - x$ é uma negação no reticulado completo $[0, 1]$.

No contexto da MM, uma erosão (algébrica) é um mapeamento ε de um reticulado completo \mathbb{L} para um reticulado completo \mathbb{M} que comuta com o operador ínfimo. Em outras palavras, um operador ε representa uma erosão se, e somente se, a seguinte igualdade valer para todo subconjunto $Y \subseteq \mathbb{L}$:

$$\varepsilon\left(\bigwedge Y\right) = \bigwedge_{y \in Y} \varepsilon(y) \quad (2.6)$$

Similarmente, um operador $\delta : \mathbb{L} \rightarrow \mathbb{M}$ representa uma dilatação (algébrica) se, e somente se, para todo subconjunto $Y \subseteq \mathbb{L}$ valer a seguinte igualdade:

$$\delta\left(\bigvee Y\right) = \bigvee_{y \in Y} \delta(y) \quad (2.7)$$

Um operador $\bar{\varepsilon} : \mathbb{L} \rightarrow \mathbb{M}$ é uma anti-erosão (algébrica) e um operador $\bar{\delta} : \mathbb{L} \rightarrow \mathbb{M}$ é uma anti-dilatação (algébrica) se, e somente se, para todo subconjunto $Y \subseteq \mathbb{L}$ valer a Eq. 2.8 e Eq. 2.9, respectivamente [99].

$$\bar{\varepsilon}\left(\bigwedge Y\right) = \bigvee_{y \in Y} \bar{\varepsilon}(y) \quad (2.8)$$

$$\bar{\delta}\left(\bigvee Y\right) = \bigwedge_{y \in Y} \bar{\delta}(y) \quad (2.9)$$

Alguns pesquisadores adicionam o sufixo “algébrico” aos nomes dos operadores morfológicos apresentados acima, devido ao fato dos mesmos serem definidos a partir de certas propriedades algébricas, e assim, distinguir estas definições

daquelas baseadas em medidas de inclusão e intersecção, como na MM binária [8].

O próximo lema indica que certas composições de operações elementares da MM também representam operações morfológicas elementares. A prova deste lema é bastante óbvia e segue diretamente a aplicação das propriedades algébricas dos respectivos operadores elementares, tal como foram definidas acima.

Lema 2. *Sejam $\varepsilon, \delta, \bar{\varepsilon}, \bar{\delta}$ respectivamente uma erosão, uma dilatação, uma anti-erosão e uma anti-dilatação de um reticulado completo \mathbb{L} para um reticulado completo \mathbb{M} . Dado um operador $f : \mathbb{M} \rightarrow \mathbb{K}$ onde \mathbb{K} é um reticulado completo, as seguintes afirmações são verdadeiras:*

1. *Se f é uma erosão, então, $f \circ \varepsilon$ é uma erosão e $f \circ \bar{\delta}$ é uma anti-dilatação.*
2. *Se f é uma dilatação, então, $f \circ \delta$ é uma dilatação e $f \circ \bar{\varepsilon}$ é uma anti-erosão.*

Banon e Barrera mostraram um resultado importante para a MM, que consiste em que todo mapeamento Ψ entre reticulados completos \mathbb{L} e \mathbb{M} pode ser expresso em termos de supremos e ínfimos das quatro operações básicas da morfologia matemática [5]. Precisamente, todo mapeamento $\Psi : \mathbb{L} \rightarrow \mathbb{M}$ pode ser expresso como um supremo de pares de ínfimos entre erosões e anti-dilatações, isto é, existem erosões ε^i e anti-dilatações $\bar{\delta}^i$ para algum conjunto de índices I tal que

$$\Psi = \bigvee_{i \in I} (\varepsilon^i \wedge \bar{\delta}^i) \quad (2.10)$$

Alternativamente, $\Psi : \mathbb{L} \rightarrow \mathbb{M}$ pode ser expresso como um ínfimo de pares de supremos entre dilatações e anti-erosões, e consequentemente uma versão dual da Eq. 2.10 pode ser obtida. Esta decomposição está de acordo com o modelo de processamento de várias redes neurais morfológicas importantes, tais como o perceptron morfológico e o MP/CL (que serão abordadas no Capítulo 4) [18, 19, 88]. A prova da Equação 2.10 é construtiva e requer somente a estrutura de reticulado completo de \mathbb{L} e \mathbb{M} [5].

A MM pode agregar duas importantes noções de dualidade: *negação* e *adjunção*. Anti-dilatações e anti-erosões podem ser construídas pela combinações de negações com erosões ou dilatações, conforme o teorema a seguir [99].

Teorema 1. *Sejam \mathbb{L} e \mathbb{M} reticulados completos com negações $\mathbf{v}_{\mathbb{L}}$ e $\mathbf{v}_{\mathbb{M}}$, respectivamente.*

1. *Um operador $\bar{\delta} : \mathbb{L} \rightarrow \mathbb{M}$ é uma anti-dilatação $\Leftrightarrow \bar{\delta} = \varepsilon \circ \mathbf{v}_{\mathbb{L}}$ e $\bar{\delta} = \mathbf{v}_{\mathbb{M}} \circ \delta$, onde δ é uma dilatação e ε é uma erosão.*
2. *Um operador $\bar{\varepsilon} : \mathbb{M} \rightarrow \mathbb{L}$ é uma anti-erosão $\Leftrightarrow \bar{\varepsilon} = \delta \circ \mathbf{v}_{\mathbb{M}}$ e $\bar{\varepsilon} = \mathbf{v}_{\mathbb{L}} \circ \varepsilon$, onde ε é uma erosão e δ é uma dilatação.*

Sejam o mapeamento $\Psi : \mathbb{L} \rightarrow \mathbb{M}$, e $\mathbf{v}_{\mathbb{L}}$, $\mathbf{v}_{\mathbb{M}}$ negações nos respectivos reticulados completos \mathbb{L} e \mathbb{M} . O operador Ψ^v , chamado *negação de Ψ* , é dado por:

$$\Psi^v(x) = \mathbf{v}_{\mathbb{M}}(\Psi(\mathbf{v}_{\mathbb{L}}(x))) \quad \forall x \in \mathbb{L} \quad (2.11)$$

Os operadores de erosão e dilatação podem ser ligados pelo conceito de negação, conforme a proposição a seguir [36].

Proposição 1. *A negação de uma erosão é uma dilatação, e vice-versa.*

A seguir, será apresentada a noção de dualidade com respeito a adjunção. Considere dois operadores arbitrários $\delta : \mathbb{L} \rightarrow \mathbb{M}$ e $\varepsilon : \mathbb{M} \rightarrow \mathbb{L}$. Dizemos que o par (ε, δ) é uma adjunção se, e somente se, a seguinte expressão valer:

$$\delta(x) \leq y \Leftrightarrow x \leq \varepsilon(y) \quad \forall x \in \mathbb{L}, y \in \mathbb{M} \quad (2.12)$$

Note que uma consequência direta da Proposição 1 é que, se o par (ε, δ) representa uma adjunção, então, o par $(\delta^v, \varepsilon^v)$ também forma uma adjunção.

A próxima proposição garante que existe uma única erosão que pode ser associada com uma certa dilatação, e vice-versa, em termos adjunção [77, 36].

Proposição 2. *Sejam \mathbb{L} e \mathbb{M} reticulados completos. Considere os mapeamentos $\delta : \mathbb{L} \rightarrow \mathbb{M}$ e $\varepsilon : \mathbb{M} \rightarrow \mathbb{L}$.*

1. *Se (ε, δ) é uma adjunção, então, δ é uma dilatação e ε é uma erosão.*
2. *Para uma dilatação δ existe uma única erosão ε tal que (ε, δ) é uma adjunção, e a erosão adjunta é dada por*

$$\varepsilon(x) = \bigvee \{y \in \mathbb{L} : \delta(y) \leq x\}, \quad \forall x \in \mathbb{M} \quad (2.13)$$

3. Para uma erosão ε existe uma única dilatação δ tal que (ε, δ) é uma adjunção, e a dilatação adjunta é dada por

$$\delta(x) = \bigwedge \{y \in \mathbb{M} : \varepsilon(y) \geq x\}, \quad \forall x \in \mathbb{L} \quad (2.14)$$

As operações elementares da morfologia matemática, de fato, podem ser produzidas em um reticulado se uma certa operação adicional estiver definida além das operações de ínfimo e supremo. Se \mathbb{L} é um reticulado arbitrário, dizemos que \mathbb{L} é um *reticulado com ordem de grupo* ou um *l-grupo* [7] se \mathbb{L} também forma um grupo com uma certa operação “+” (geralmente, a notação de adição é usada para notações de grupo) e se toda translação de grupo é isótona, i.e., a seguinte condição é satisfeita para todo $x, y \in \mathbb{L}$ tal que $x \leq y$:

$$a + x + b \leq a + y + b \quad \forall a, b \in \mathbb{L}. \quad (2.15)$$

Se um l-grupo \mathbb{L} também for um reticulado condicionalmente completo, então dizemos que \mathbb{L} é um l-grupo condicionalmente completo.

De maneira mais geral, um *grupo parcialmente ordenado* é um grupo que também constitui um conjunto parcialmente ordenado e satisfaz a Equação 2.15. O único grupo parcialmente ordenado limitado, i.e., equipado com fronteiras universais é o grupo trivial $\{0\}$, onde 0 denota o elemento neutro com respeito a adição. Portanto, o grupo trivial $\{0\}$ é também o único reticulado com ordem de grupo que representa um reticulado limitado [7]. Todavia, R. Cuninghame-Green considera, na teoria matemática da álgebra minimax, estruturas algébricas denominadas “*bounded lattice ordered groups*” (ou *blogs*), que são dadas por um reticulado limitado \mathbb{G} cujo conjunto dos elementos finitos $\mathbb{F} = \mathbb{G} \setminus \{+\infty, -\infty\}$ forma um l-grupo (lembre que $-\infty = \bigwedge \mathbb{G}$ e $+\infty = \bigvee \mathbb{G}$ representam, respectivamente, o menor e maior elemento de \mathbb{G}). Contudo, o nome **blog** sugere a existência de estruturas de grupos em reticulados limitados diferente de $\{0\}$, por isso, neste trabalho, preferimos chamar essa estrutura de *extensão de reticulado limitado com ordem de grupo* (ou extensão de l-grupo) ao invés de blog. No caso especial onde a extensão de l-grupo \mathbb{G} é um reticulado completo nos referimos a \mathbb{G} como *extensão de l-grupo completo*.

Para que uma extensão de l-grupo \mathbb{G} seja consistente, é necessário que o operador “+” também seja definido em $(\mathbb{G} \times \mathbb{G}) \setminus (\mathbb{F} \times \mathbb{F})$. A fim de simplicidade,

estamos interessados apenas em \mathbb{G} diferente do grupo trivial $\{0\}$. Como toda translação de grupo é um automorfismo de reticulado e para todo $x \in \mathbb{F}$ tem-se $-\infty \leq x \leq +\infty$, as seguintes definições são utilizadas para manter a isotonia, i.e., preservação da ordem:

$$a + (+\infty) = +\infty + a = +\infty, \forall a \in \mathbb{F} \cup \{+\infty\}, \quad (2.16)$$

$$a + (-\infty) = -\infty + a = -\infty, \forall a \in \mathbb{F} \cup \{-\infty\}. \quad (2.17)$$

Ainda falta definir “+” em $\{(+\infty, -\infty), (-\infty, +\infty)\}$. Para isto, existem duas boas opções: $(-\infty) + (+\infty) = (+\infty) + (-\infty) = +\infty$ que é coerente com a Equação 2.16 e $(-\infty) + (+\infty) = (+\infty) + (-\infty) = -\infty$ que é coerente com a Equação 2.17. Posto isto, uma nova operação binária $+$ que coincide com a operação $+$ em $\mathbb{G} \times \mathbb{G} \setminus \{(+\infty, -\infty), (-\infty, +\infty)\}$ pode ser definida e suas diferenças são dadas pelas seguintes regras:

$$(-\infty) + (+\infty) = (+\infty) + (-\infty) = -\infty, \quad (2.18)$$

$$(-\infty) + ' (+\infty) = (+\infty) + ' (-\infty) = +\infty. \quad (2.19)$$

Observe que $(\mathbb{G}, +)$ e $(\mathbb{G}, +')$ não formam grupos já que nem $-\infty$ e nem $+\infty$ possuem inverso aditivo.

Dessa maneira, uma extensão de \mathbf{l} -grupo pode ser denotada pelo sistema $(\mathbb{G}, \vee, \wedge, +, +')$ (denotaremos por apenas \mathbb{G} quando não houver ambiguidade sobre os operadores). No caso especial onde \mathbb{G} é extensão de \mathbf{l} -grupo (completo) e também uma corrente, nos referiremos à \mathbb{G} como *extensão de \mathbf{l} -grupo (completo) totalmente ordenado*. Exemplos de extensões de \mathbf{l} -grupos completos totalmente ordenados são $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$ e $(\mathbb{Z}_{\pm\infty}, \vee, \wedge, +, +')$ onde $+$, $+$ coincidem com a operação usual de adição em \mathbb{R} , e $(\mathbb{R}_{+\infty}^{\geq 0}, \vee, \wedge, \cdot, \cdot')$ onde \cdot e \cdot' correspondem a multiplicação usual nos números reais positivos. Note que $(\mathbb{Q}_{\pm\infty}, \vee, \wedge, +, +')$ representa uma extensão de \mathbf{l} -grupo totalmente ordenado mas não uma extensão de \mathbf{l} -grupo completo totalmente ordenado, pois que \mathbb{Q} não é reticulado completo.

Por fim, observe que as operações $\vee, \wedge, +, +'$ definidas em \mathbb{G} podem ser entendidas para \mathbb{G}^n , onde $n > 1$, tomando coordenada a coordenada.

Além disso, o Lema 1 por ser utilizado para produzir uma extensão de \mathbf{l} -grupo completo a partir de produtos de \mathbf{l} -grupo *condicionalmente* completos. Em

particular, seja $\bar{\mathbb{R}}^n = \mathbb{R}^n \cup \{+\infty, -\infty\}$ e $+, +'$ definidas fora de $\mathbb{R}^n \times \mathbb{R}^n$ como nas Equações 2.16, 2.17, 2.18 e 2.19, o sistema $(\bar{\mathbb{R}}^n, \vee, \wedge, +, +')$ forma uma extensão de l-grupo completo que não é uma corrente quando $n > 1$. De maneira analoga obtemos a extensão de l-grupo completo $(\mathbb{Z}^n \cup \{+\infty, -\infty\}, \vee, \wedge, +, +')$.

A existência do inverso aditivo em um estrutura de l-grupo permite definir o conceito de elemento conjugado em uma extensão de l-grupo [18, 19]. Seja \mathbb{G} uma extensão de l-grupo. Na álgebra minimax o conjugado de um elemento $x \in \mathbb{G}$ é denotado por x^* e é definido por:

$$x^* = \begin{cases} -x & \text{se } x \in \mathbb{G} \setminus \{+\infty, -\infty\} \text{ (inverso aditivo)} \\ -\infty & \text{se } x = +\infty \\ +\infty & \text{se } x = -\infty \end{cases} \quad (2.20)$$

Note que o mapeamento $v_* : \mathbb{G} \rightarrow \mathbb{G}$, dado por $v_*(x) = x^*$ representa uma negação no reticulado \mathbb{G} . Uma matriz $A \in \mathbb{G}^{m \times n}$ corresponde a uma matriz cojugada $A^* \in \mathbb{G}^{n \times m}$, onde cada componente $[a_{ij}^*]$ de A^* é dada por $[a_{ij}^*] = [a_{ji}]^*$. Obviamente, segue que $(A^*)^* = A$.

A operação de máximo e mínimo, de duas matrizes de dimensões compatíveis, é executada em cada componente, por exemplo, $C = A \vee B$ é dada por $[c_{ij}] = [a_{ij} \vee b_{ij}]$, para $A, B \in \mathbb{G}^{m \times n}$.

Os modelos de redes neurais morfológicas, discutidos neste trabalho, utilizam dois tipos de produtos de matrizes. Sejam as matrizes A e B com entradas em \mathbb{G} de dimensões, respectivamente, $m \times p$ e $p \times n$, a matriz $C = A \boxtimes B$ é o *produto max* de A e B e a matriz $D = A \boxdot B$ é o *produto min* de A e B , e são definidas pelas seguintes equações:

$$c_{ij} = \bigvee_{k=1}^p (a_{ik} + b_{kj}) \quad (2.21)$$

$$d_{ij} = \bigwedge_{k=1}^p (a_{ik} +' b_{kj}). \quad (2.22)$$

A partir das definições anteriores, uma elegante relação de dualidade entre operações de matrizes [18, 19] é dada por

$$(A \wedge B)^* = A^* \vee B^* \text{ e } (A \boxdot C)^* = C^* \boxtimes A^* \quad (2.23)$$

para $A, B \in \mathbb{G}^{m \times n}$ e $C \in \mathbb{G}^{n \times p}$.

O lema enunciado a seguir é útil para demonstrar que, de fato, erosões e dilatações podem ser obtidas através, respectivamente, de produtos min e produtos max. Lembre que, em uma extensão de \mathbf{l} -grupo, os operadores $+$ e $+'$ possuem prioridade em relação as operações de \vee e \wedge do reticulado. Entretanto, o uso de parênteses pode ser utilizado para alterar o significado da expressão, por exemplo $(x \vee y) + z$.

Lema 3. [91] *Sejam \mathbb{G} uma extensão de \mathbf{l} -grupo completo e J um conjunto arbitrário de índices. As seguintes igualdades valem para todo $w, x^j \in \mathbb{G}$ onde $j \in J$:*

$$w +' (\bigwedge_{j \in J} x^j) = \bigwedge_{j \in J} w +' x^j, \quad (2.24)$$

$$w + (\bigvee_{j \in J} x^j) = \bigvee_{j \in J} w + x^j. \quad (2.25)$$

Teorema 2. [91] *Seja \mathbb{G} uma extensão de \mathbf{l} -grupo completo. Considere os seguintes operadores $\varepsilon_A, \delta_A : \mathbb{G}^n \rightarrow \mathbb{G}^m$ para $A \in \mathbb{G}^{n \times m}$:*

$$\varepsilon_A(\mathbf{x}) = A^t \boxtimes \mathbf{x}, \quad (2.26)$$

$$\delta_A(\mathbf{x}) = A^t \boxdot \mathbf{x}. \quad (2.27)$$

Temos que ε_A representa uma erosão e δ_A representa uma dilatação do reticulado completo \mathbb{G}^n para o reticulado completo \mathbb{G}^m .

Os operadores ε_A e δ_A são utilizados por vários modelos de redes neurais morfológicas, tais como o perceptron morfológico [72] e as memórias associativas morfológicas [73]. O próximo teorema mostra como verificar a relação de ordem entre dois elementos de uma extensão de \mathbf{l} -grupo, através de operações morfológicas elementares baseadas em produto min ou produto max.

Teorema 3. *Sejam \mathbb{G} uma extensão de \mathbf{l} -grupo e $\mathbf{x}, \mathbf{a}, \mathbf{b} \in \mathbb{G}^n$. As seguintes afirmações são válidas:*

1. *A desigualdade $\mathbf{a} \leq \mathbf{x}$ vale se, e somente se, $\varepsilon_{\mathbf{v}}(\mathbf{x}) \geq \mathbf{0}$, onde $\mathbf{v} = \mathbf{a}^*$, ou se $\bar{\varepsilon}_{\mathbf{w}}(\mathbf{x}) \leq \mathbf{0}$, onde $\mathbf{w} = \mathbf{a}$.*
2. *A desigualdade $\mathbf{x} \leq \mathbf{b}$ vale se, e somente se, $\delta_{\mathbf{v}}(\mathbf{x}) \leq \mathbf{0}$, onde $\mathbf{v} = \mathbf{b}^*$, ou se $\bar{\delta}_{\mathbf{w}}(\mathbf{x}) \geq \mathbf{0}$, onde $\mathbf{w} = \mathbf{b}$.*

O Teorema 3 implica que os elementos de uma hipercaixa $[\mathbf{a}, \mathbf{b}] = \{\mathbf{x} \in \mathbb{G}^n : \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\}$ podem ser determinados pela verificação da seguinte igualdade

$$\varepsilon_{\mathbf{a}^*}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{b}}(\mathbf{x}) \geq \mathbf{0}. \quad (2.28)$$

Em especial, esse fato é usado na estratégia de classificação de algumas redes neurais morfológicas, tal como o perceptron morfológico. A Figura 2.3 provê uma ilustração desta situação para a hipercaixa $[\mathbf{a} = 1, \mathbf{b} = 9]$ na extensão de \mathbb{l} -grupo $\mathbb{R}_{\pm\infty}$.

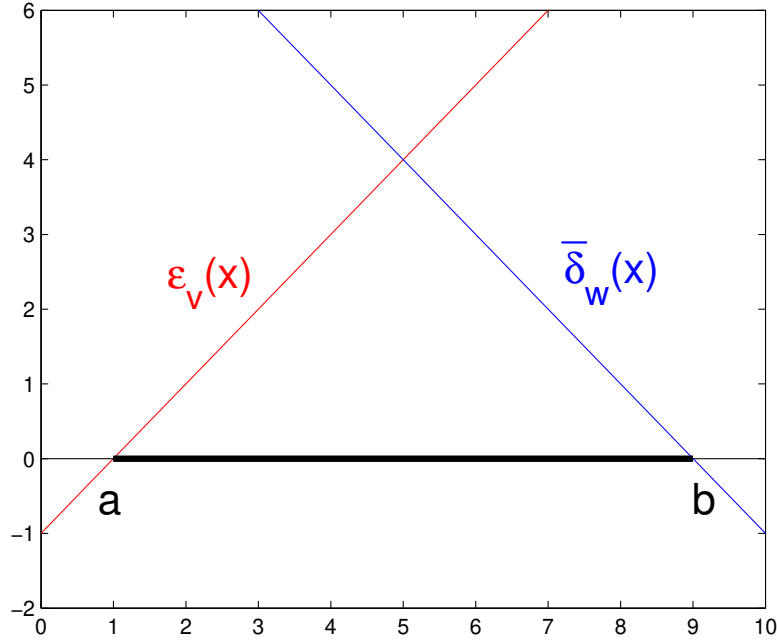


Figura 2.3: Exemplo de uma erosão ε_v e uma anti-dilatação $\bar{\delta}_w$ na extensão de \mathbb{l} -grupo \mathbb{R} , onde $v = -a$ e $w = b$ para $a, b \in \mathbb{R}$.

Nesta seção apresentamos a caracterização da MM sobre a estrutura de reticulados completos numa forma algébrica. Contudo, o desenvolvimento inicial da MM, especificamente a MM binária, *não* foi motivada por esse sentido algébrico, apesar de satisfazê-la. A próxima seção objetiva-se a esclarecer o sentido intuitivo com o qual surgiu a MM binária.

2.2 Motivação e conceitos básicos da morfologia matemática

A morfologia matemática binária compreende dois operadores fundamentais, a *erosão* e a *dilatação*, que podem ser descritos, respectivamente, em termos de inclusões e intersecções entre conjuntos [56, 57, 58, 76, 77]. A seguir, serão apresentadas as definições básicas da MM binária junto a alguns exemplos e a uma análise intuitiva das operações de erosão e dilatação.

Seja \mathbf{X} o símbolo que denota o espaço euclidiano \mathbb{R}^d ou o espaço digital \mathbb{Z}^d (onde d é um inteiro positivo), então uma imagem binária \mathbf{A} é identificada como um subconjunto de \mathbf{X} , isto é, $\mathbf{A} \subseteq \mathbf{X}$. As operações de erosão binária E_B e a dilatação binária D_B são associadas a um subconjunto \mathbf{S} de \mathbf{X} chamado de *elemento estruturante* (SE). Note que o elemento estruturante pode ser visto como uma imagem de acordo com a definição anterior.

Sejam a translação de uma imagem $\mathbf{A} \subseteq \mathbf{X}$ por $\mathbf{x} \in \mathbf{X}$ denotada por $\mathbf{A}_\mathbf{x}$ e dada por $\mathbf{A}_\mathbf{x} = \{\mathbf{a} + \mathbf{x} : \mathbf{a} \in \mathbf{A}\}$, e a reflexão de \mathbf{A} em torno da origem é denotada por $\bar{\mathbf{A}}$ e dada por $\bar{\mathbf{A}} = \{-\mathbf{a} \in \mathbf{X} : \mathbf{a} \in \mathbf{A}\}$. A erosão $E_B(\mathbf{A}, \mathbf{S})$ e a dilatação $D_B(\mathbf{A}, \mathbf{S})$ de uma imagem binária \mathbf{A} por um elemento estruturante \mathbf{S} , são definidas pelas Equações 2.29 e 2.30 que correspondem, respectivamente, à subtração e à adição de Minkowski [77, 76].

$$E_B(\mathbf{A}, \mathbf{S}) = \{\mathbf{x} \in \mathbf{X} : \mathbf{S}_\mathbf{x} \subseteq \mathbf{A}\} = \bigcap_{\mathbf{s} \in \bar{\mathbf{S}}} \mathbf{A}_\mathbf{s} \quad (2.29)$$

$$D_B(\mathbf{A}, \mathbf{S}) = \{\mathbf{x} \in \mathbf{X} : \bar{\mathbf{S}}_\mathbf{x} \cap \mathbf{A} \neq \emptyset\} = \bigcup_{\mathbf{s} \in \bar{\mathbf{S}}} \mathbf{A}_\mathbf{s} = \bigcup_{\mathbf{a} \in \mathbf{A}} \mathbf{S}_\mathbf{a} \quad (2.30)$$

No sentido intuitivo, uma erosão (Eq. 2.29) calcula em quais pontos \mathbf{x} do espaço \mathbf{X} um elemento estruturante \mathbf{S} está *incluso* em uma imagem \mathbf{A} , enquanto uma dilatação (Eq. 2.30) mede quais pontos o elemento estruturante refletido $\bar{\mathbf{S}}$ se *intersepta* com a imagem \mathbf{A} . As Figuras 2.4 e 2.5, ilustram a operação de erosão e dilatação de uma imagem \mathbf{A} por um elemento estruturante \mathbf{S} , respectivamente.

Note que, para um elemento estruturante \mathbf{S} fixo, as Equações 2.4 e 2.5 também satisfazem, respectivamente, às definições de erosão algébrica e dilatação algébrica, pois, os operadores \cap e \cup representam operações de ínfimo e supremo no reticulado completo $\mathcal{P}(\mathbf{X})$ (o conjunto de todos os subconjuntos de \mathbf{X}). Portanto,

utilizada para definir uma morfologia matemática fuzzy cujos operadores básicos estejam baseados em medidas de inclusão e conjunção fuzzy [96], conforme será elucidado na Seção 2.3. Numa forma algébrica, a morfologia matemática pode ser conduzida sobre a estrutura de reticulados completos como visto na Seção 2.1 [36, 75, 77].

2.3 Conceitos básicos da morfologia matemática fuzzy

Originalmente a MM foi desenvolvida como uma abordagem à teoria de conjuntos para análise de imagens binárias [58, 76]. Extensões para morfologia matemática de tons de cinza incluem as abordagens de *umbra* e *threshold* [76, 81], além de várias abordagens na direção da morfologia matemática fuzzy (FMM) [60, 96].

Em poucas palavras, a idéia inicial que motivou a extensão da MM binária a partir da teoria dos conjuntos fuzzy, consiste na substituição dos operadores de conjuntos (inclusão, intersecção e complemento) nas equações de erosão e dilatação binárias, pelas respectivas extensões na lógica fuzzy. Especificamente, uma abordagem à MM fuzzy, em geral, considera certas medidas de inclusão fuzzy e intersecção fuzzy, assim como, uma certa negação fuzzy para definir operações morfológicas elementares: erosão fuzzy, dilatações fuzzy, anti-erosão fuzzy e anti-dilatação fuzzy. Esse tipo de abordagem é consistente com a motivação intuitiva com a qual surgiu e se desenvolveu a MM, pois, só posteriormente é que a mesma foi conduzida sobre uma estrutura de reticulados completos e seus operadores elementares caracterizados através de propriedades algébricas. Uma outra maneira de se conduzir uma abordagem à morfologia matemática fuzzy é a partir das definições algébricas conforme as Equações 2.6, 2.7, 2.8 e 2.9.

Neste trabalho, denotaremos um conjunto fuzzy A como sendo uma função \mathbf{a} de um domínio \mathbf{X} em $[0, 1]$, onde o valor $\mathbf{a}(\mathbf{x})$ é o *grau de pertinência* de \mathbf{x} em A . O símbolo $\mathcal{F}(\mathbf{X})$ denota a classe de todos os conjuntos fuzzy em \mathbf{X} . No caso particular onde $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^k]$ denotaremos o conjunto fuzzy $\mathbf{a} \in \mathcal{F}(\mathbf{X})$ como um vetor coluna $\mathbf{a} = [\mathbf{a}_1, \dots, \mathbf{a}_k]^T \in [0, 1]^k$ onde $\mathbf{a}_i = \mathbf{a}(\mathbf{x}^i)$ para todo $i = 1, \dots, k$. Lembre que $\mathcal{F}(\mathbf{X})$ é um reticulado completo com a ordem parcial dada pela Eq. 2.4.

Seja \mathbf{X} um grupo, tipicamente o l-grupo \mathbb{R}^n ou \mathbb{Z}^n . Para $\mathbf{x} \in \mathbf{X}$ e $\mathbf{s} \in \mathcal{F}(\mathbf{X})$ denotamos por $\mathbf{s}_{\mathbf{x}}, \bar{\mathbf{s}} \in \mathcal{F}(\mathbf{X})$ a translação de \mathbf{s} por \mathbf{x} dada pela Eq. 2.31 e a reflexão de \mathbf{s} em torno da origem dada por Eq. 2.32.

$$\mathbf{s}_{\mathbf{x}}(\mathbf{y}) = \mathbf{s}(\mathbf{y} - \mathbf{x}) \quad \forall \mathbf{y} \in \mathbf{X} \quad (2.31)$$

$$\bar{\mathbf{s}}(\mathbf{y}) = \mathbf{s}(-\mathbf{y}) \quad \forall \mathbf{y} \in \mathbf{X} \quad (2.32)$$

Seja $\text{Inc}_{\mathcal{F}} : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \rightarrow [0, 1]$ uma inclusão fuzzy. A erosão fuzzy $\varepsilon_{\mathcal{F}}(\mathbf{a}, \mathbf{s})$ de um conjunto fuzzy \mathbf{a} por um elemento estruturante $\mathbf{s} \in \mathcal{F}(\mathbf{X})$ é dada por

$$\varepsilon_{\mathcal{F}}(\mathbf{a}, \mathbf{s})(\mathbf{x}) = \text{Inc}_{\mathcal{F}}(\mathbf{s}_{\mathbf{x}}, \mathbf{a}) \quad \forall \mathbf{x} \in \mathbf{X} \quad (2.33)$$

Similarmente, se $\text{Sec}_{\mathcal{F}} : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \rightarrow [0, 1]$ denota uma intersecção fuzzy então a dilatação fuzzy $\Delta_{\mathcal{F}}(\mathbf{a}, \mathbf{s})$ de um conjunto fuzzy \mathbf{a} por um elemento estruturante $\mathbf{s} \in \mathcal{F}(\mathbf{X})$ é dada por

$$\Delta_{\mathcal{F}}(\mathbf{a}, \mathbf{s})(\mathbf{x}) = \text{Sec}_{\mathcal{F}}(\bar{\mathbf{s}}_{\mathbf{x}}, \mathbf{a}) \quad \forall \mathbf{x} \in \mathbf{X} \quad (2.34)$$

Uma grande variedade de medidas de inclusão fuzzy e intersecção fuzzy denominadas, respectivamente, por Inf-I e Sup-C (ou por Min-I e Max-C no caso de \mathbf{X} finito), podem ser construídas em termos de implicações $I_{\mathcal{F}}$ e conjunções fuzzy $C_{\mathcal{F}}$ [96] como se segue:

$$\text{Inc}_{\mathcal{F}}(\mathbf{a}, \mathbf{b}) = \bigwedge_{\mathbf{x} \in \mathbf{X}} I_{\mathcal{F}}(\mathbf{a}(\mathbf{x}), \mathbf{b}(\mathbf{x})) \quad (2.35)$$

$$\text{Sec}_{\mathcal{F}}(\mathbf{a}, \mathbf{b}) = \bigvee_{\mathbf{x} \in \mathbf{X}} C_{\mathcal{F}}(\mathbf{a}(\mathbf{x}), \mathbf{b}(\mathbf{x})) \quad (2.36)$$

Alguns pesquisadores em FMM tem concebido medidas de inclusão fuzzy que relaxam a noção da medida de inclusão e intersecção crisp, permitindo a definição de medidas que não satisfazem a propriedade de herança [16], isto é, que as restrições de $\text{Inc}_{\mathcal{F}}$ e $\text{Sec}_{\mathcal{F}}$ à $\mathcal{P}(\mathbf{X}) \times \mathcal{P}(\mathbf{X})$ não coincidem com a inclusão e intersecção de conjuntos crisp, respectivamente. Por exemplo, a medida de subsethood de Kosko utilizada no modelo KS-FAM discutido no próximo capítulo deste trabalho representa uma medida de inclusão fuzzy que não satisfaz a propriedade de

herança [86, 87].

As definições de erosões fuzzy (Eqs 2.33) e dilatações fuzzy (Eqs 2.34) dadas acima obedecem à regra intuitiva com a qual foram definidos seus correspondentes no caso binário. Apesar disso, muitos autores preferem (ou apenas consideram) utilizar as definições algébricas desses operadores, propiciada pela estrutura de reticulado completo formada pela classe dos conjuntos fuzzy, aproveitando-se, assim, das propriedades e teoremas estabelecidos para estes operadores, tal como a relação de adjunção e a decomposição de Banon e Barrera.

Formalmente, dizemos que uma função $\varepsilon_{\mathcal{F}}(\cdot, \mathbf{s}) : \mathcal{F}(\mathbf{X}) \rightarrow \mathcal{F}(\mathbf{X})$ é uma *erosão fuzzy no sentido algébrico* se, e somente se, $\varepsilon_{\mathcal{F}}$ satisfaz a Eq. 2.6 para todo $\mathbf{s} \in \mathcal{F}(\mathbf{X})$. Similarmente, dizemos que uma função $\delta_{\mathcal{F}}(\cdot, \mathbf{s}) : \mathcal{F}(\mathbf{X}) \rightarrow \mathcal{F}(\mathbf{X})$ é uma *dilatação fuzzy no sentido algébrico* se, e somente se, $\delta_{\mathcal{F}}$ satisfaz a Eq. 2.7 para todo $\mathbf{s} \in \mathcal{F}(\mathbf{X})$ [96].

Observe na Equação 2.35 que, se a implicação fuzzy $I_{\mathcal{F}}(\mathbf{b}, \cdot)$ comutar com o operador de ínfimo para todo $\mathbf{b} \in [0, 1]$, então, $\text{Inc}_{\mathcal{F}}(\mathbf{s}, \cdot)$ também comutará com o ínfimo para todo $\mathbf{s} \in \mathcal{F}(\mathbf{X})$ e, portanto, a função resultante $\varepsilon_{\mathcal{F}}$ satisfará a definição de uma erosão fuzzy, tanto no sentido algébrico quanto no sentido intuitivo. Exemplos de implicações fuzzy que produzem erosões fuzzy em ambos os sentidos:

$$I_M(x, y) = \begin{cases} 1, & x \leq y \\ y, & x > y \end{cases} \quad (\text{Gödel}) \quad (2.37)$$

$$I_P(x, y) = \begin{cases} 1, & x \leq y \\ \frac{y}{x}, & x > y \end{cases} \quad (\text{Goguen}) \quad (2.38)$$

$$I_W(x, y) = 1 \wedge (y - x + 1) \quad (\text{Lukasiewicz}) \quad (2.39)$$

$$I_K(x, y) = (1 - x) \vee y \quad (\text{Kleene}). \quad (2.40)$$

Similarmente, se a conjunção fuzzy $C_{\mathcal{F}}$ da Equação 2.36 for tal que $C_{\mathcal{F}}(\mathbf{b}, \cdot)$ é uma dilatação algébrica $[0, 1] \rightarrow [0, 1]$, isto é, comuta com o operador supremo para todo $\mathbf{b} \in [0, 1]$, então, a função correspondente $\Delta_{\mathcal{F}}(\cdot, \mathbf{s})$ é tanto uma dilatação fuzzy $\mathcal{F}(\mathbf{X}) \rightarrow \mathcal{F}(\mathbf{X})$ no sentido intuitivo quanto no sentido algébrico. Exemplos de conjunções que representam dilatações algébricas em ambos os argumentos:

$$T_M(x, y) = x \wedge y \quad (\text{Mínimo}) \quad (2.41)$$

$$T_P(x, y) = x \cdot y \quad (\text{Produto}) \quad (2.42)$$

$$T_W(x, y) = 0 \vee (x + y - 1) \quad (\text{Lukasiewicz}) \quad (2.43)$$

$$C_K(x, y) = \begin{cases} 0, & x + y \leq 1 \\ x, & x + y > 1 \end{cases} \quad (\text{Kleene}) \quad (2.44)$$

Note que os operadores T_M , T_P e T_L correspondem a t-normas [48]. De fato, toda t-norma *contínua* representa uma dilatação algébrica $[0, 1] \rightarrow [0, 1]$ em ambos os argumentos [99].

Além disso, conjunções fuzzy e implicações fuzzy podem estar relacionadas pelo conceito de adjunção. Especificamente, dizemos que uma conjunção fuzzy C e uma implicação fuzzy I formam uma adjunção se, e somente se, $C(\cdot, z)$ e $I(z, \cdot)$ formarem uma adjunção para todo $z \in [0, 1]$, isto é, se C e I satisfizerem a seguinte equação para todo $x, y, z \in [0, 1]$ [99]:

$$C(x, z) \leq y \Leftrightarrow x \leq I(z, y) \quad (2.45)$$

Pela proposição 2, $C(\cdot, z)$ e $I(z, \cdot)$ representam, respectivamente, uma dilatação algébrica e uma erosão algébrica. Além disso, uma implicação fuzzy I e uma conjunção C podem ser obtidas de acordo com as seguintes equações:

$$I(y, x) = \bigvee \{z \in [0, 1] : C(z, y) \leq x\} \text{ e } C(x, y) = \bigwedge \{z \in [0, 1] : I(y, z) \geq x\} \quad (2.46)$$

Os pares (T_M, I_M) , (T_P, I_P) , (T_W, I_W) e (C_K, I_K) constituem exemplos de operadores adjuntos.

A partir das Equações 2.35 e 2.36 podemos definir dois produtos matriciais na lógica fuzzy. Sejam $\mathbf{A} \in [0, 1]^{m \times k}$ e $\mathbf{B} \in [0, 1]^{k \times n}$, os *produtos max-C* e *min-I* de \mathbf{A} por \mathbf{B} são denotados, respectivamente, por $\mathbf{D} = \mathbf{A} \circ \mathbf{B}$ e $\mathbf{E} = \mathbf{A} \otimes \mathbf{B}$ e definidos pelas seguintes Equações [61, 99], onde C e I correspondem a uma conjunção fuzzy e a uma implicação fuzzy:

$$d_{ij} = \bigvee_{\xi=1}^k C(a_{i\xi}, b_{\xi j}) \quad \forall i = 1, \dots, m \text{ e } \forall j = 1, \dots, n \quad (2.47)$$

$$e_{ij} = \bigwedge_{\xi=1}^k I(b_{\xi j}, a_{i\xi}) \quad \forall i = 1, \dots, m \text{ e } \forall j = 1, \dots, n \quad (2.48)$$

Existem ainda outros produtos matriciais definidos na teoria de lógica fuzzy, tal como o produto Inf-D (ou Min-D) derivado de uma certa disjunção fuzzy D. Porém estes dois são suficientes para o propósito deste trabalho. Para mais informações indicamos [61].

Vários modelos de FMAMs são baseados nos produtos max-C e min-I [99]. Além disso, uma certa estratégia de aprendizagem para FMAMs pode ser adotada quando esses produtos representam certas operações elementares da MM fuzzy [99]. Este tema será abordado com mais detalhes no próximo capítulo.

Memórias associativas morfológicas

Uma *memória associativa* (AM) é um modelo projetado para armazenar pares de entrada e saída. Uma propriedade desejável em uma AM, é que dada uma versão incompleta ou distorcida de um padrão de entrada ela seja capaz de recordar a saída desejada. Matematicamente falando, dado k pares de vetores (ou memórias fundamentais) $(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^k, \mathbf{y}^k)$, o projeto de uma AM consiste em encontrar um mapeamento g tal que $g(\mathbf{x}^\xi) = \mathbf{y}^\xi$ para todo $\xi = 1, \dots, k$ e que seja equipado com uma certa tolerância a ruídos, isto é, para uma versão corrompida $\tilde{\mathbf{x}}^\xi$ de \mathbf{x}^ξ tenha-se $g(\tilde{\mathbf{x}}^\xi) = \mathbf{y}^\xi$. Uma *memória associativa fuzzy* (FAM) é uma AM que armazena pares de vetores que representam conjuntos fuzzy.

Uma memória associativa morfológica (MAM) [73] é um modelo de AM representada por uma rede neural morfológica. Os modelos clássicos de MAMs de tons de cinza e seus principais resultados serão discutidos na Seção 3.1. Em seguida serão abordadas as memórias associativas morfológicas fuzzy (FMAMs) que consistem em AMs descritas por redes neurais cujos neurônios executam operações elementares da morfologia matemática fuzzy, e que armazenam pares de entradas e saídas dados por conjuntos fuzzy. Note que a determinação de FMAMs depende de como (ou em que sentido) os operadores morfológicos fuzzy elementares são definidos, isto é, no sentido intuitivo, no sentido algébrico ou por ambos. Também será discutido um método de aprendizagem não iterativo, baseado na relação de dualidade com respeito a adjunção dos operadores da MM, que pode ser empregado na fase de armazenamento para uma certa classe de FMAMs [99].

Na Seção 3.3 será apresentado um modelo de AM fuzzy baseado na medida

de *subsethood* de Kosko [50, 51] que pode ser classificado como uma rede neural morfológica no sentido intuitivo. Este modelo generaliza a memória associativa binária introduzida em [87]. Por fim, encerraremos este capítulo apresentando alguns resultados de experimentos envolvendo a AM fuzzy baseada na medida de *subsethood* de Kosko.

3.1 Modelos clássicos de memórias associativas morfológicas

Suponha que se queira armazenar k pares de vetores (ou memórias fundamentais) $(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^k, \mathbf{y}^k)$ usando uma *memória associativa morfológica* (MAM). Sejam $X = [\mathbf{x}^1, \dots, \mathbf{x}^k] \in \mathbb{R}^{n \times k}$ a matriz cujas colunas são os vetores $\mathbf{x}^\xi \in \mathbb{R}^n$ e $Y = [\mathbf{y}^1, \dots, \mathbf{y}^k] \in \mathbb{R}^{m \times k}$ a matriz cujas colunas são os vetores $\mathbf{y}^\xi \in \mathbb{R}^m$, para $\xi = 1, \dots, k$. Aqui, foi considerado $(\mathbf{x}^\xi, \mathbf{y}^\xi) \in \mathbb{R}^n \times \mathbb{R}^m$ ao invés de $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ pertencente ao reticulado completo $\bar{\mathbb{R}}^n \times \bar{\mathbb{R}}^m$, visto que essa restrição é suficiente para todos os propósitos práticos. Relembre que $\bar{\mathbb{R}}^p = \mathbb{R}^p \cup \{-\infty, +\infty\}$, $p \geq 1$, representa, pelo Lema 1, uma extensão de l-grupo completo, pois, \mathbb{R}^p é um reticulado condicionalmente completo.

Considere as operações matriciais definidas na Seção 2.1, $X \in \mathbb{R}^{n \times k}$ e $Y \in \mathbb{R}^{m \times k}$. Apresentaremos dois modelos básicos de MAM: o primeiro consiste na construção de uma matriz W_{XY} de dimensão $m \times n$ dada pela Equação 3.1, e o segundo consiste em um esquema dual de construção de uma matriz M_{XY} de dimensão $m \times n$ conforme a Equação 3.2. Note ainda que se pode obter a seguinte igualdade $W_{XY} = (M_{YX})^*$ a partir da Equação 2.23.

$$W_{XY} = X \boxtimes Y^* = \bigwedge_{\xi=1}^k \mathbf{y}^\xi \boxtimes (\mathbf{x}^\xi)^* \quad (3.1)$$

$$M_{XY} = X \boxtimes Y^* = \bigvee_{\xi=1}^k \mathbf{y}^\xi \boxtimes (\mathbf{x}^\xi)^* \quad (3.2)$$

Dado um padrão de entrada $\mathbf{x} \in \mathbb{R}^n$ o processo de recordação de um padrão $\mathbf{y} \in \mathbb{R}^m$ no modelo W_{XY} é dado pela Eq. 3.3. Analogamente, uma saída $\mathbf{z} \in \mathbb{R}^m$ é produzida no modelo dual M_{XY} conforme a Eq. 3.4.

$$\mathbf{y} = \delta_{W_{XY}^t}(\mathbf{x}) = W_{XY} \boxtimes \mathbf{x} \quad (3.3)$$

$$\mathbf{z} = \varepsilon_{M_{XY}^t}(\mathbf{x}) = M_{XY} \boxdot \mathbf{x} \quad (3.4)$$

Note que as Equações 3.3 e 3.4 formam, respectivamente, uma dilatação algébrica e uma erosão algébrica, conforme o Teorema 2.

No caso onde $X = Y$ (isto é, $\mathbf{y}^\xi = \mathbf{x}^\xi$, para $\xi = 1, \dots, k$) tem-se W_{XX} e M_{XX} e são chamadas de *memórias morfológicas auto-associativas* (AMMs), caso contrário obtem-se memórias heteroassociativas (HMM). O exemplo 1 ilustra duas AMMs (W_{XX} e M_{XX}) que armazenaram com sucesso todas as memórias fundamentais (i.e., $W_{XX} \boxtimes \mathbf{x}^\xi = \mathbf{x}^\xi = M_{XX} \boxdot \mathbf{x}^\xi$ para $\xi = 1, \dots, 4$).

Exemplo 1.

$$X = \begin{pmatrix} 9 & 4 & 4 & 3 \\ 1 & 8 & 0 & 5 \\ 5 & 7 & 7 & 2 \end{pmatrix}, \quad W_{XX} = \begin{pmatrix} 0 & -4 & -3 \\ -8 & 0 & -7 \\ -4 & -3 & 0 \end{pmatrix}, \quad M_{XX} = \begin{pmatrix} 0 & 8 & 4 \\ 4 & 0 & 3 \\ 3 & 7 & 0 \end{pmatrix}$$

Para o caso binário onde $X \in \{0, 1\}^{n \times k}$, Sussner mostrou que cada componente do produto $\min M_{XX} \boxdot \mathbf{x}$ equivale ao cálculo de uma inclusão crisp de um certo elemento estruturante $\mathbf{s} \in \{0, 1\}^n$ em \mathbf{x} , para todo $\mathbf{x} \in \{0, 1\}^n$. Seja \mathbf{m}_i^t a i -ésima linha de M_{XX} e a negação (fuzzy) $N_S : [0, 1]^n \rightarrow [0, 1]^n$ definida para todo \mathbf{n} como

$$N_S(\mathbf{x}) = \bar{\mathbf{x}} = (1 - x_1, \dots, 1 - x_n), \quad \forall \mathbf{x} \in [0, 1]^n. \quad (3.5)$$

Especificamente, foi verificada a seguinte igualdade para todo $i = 1, \dots, n$ [86]:

$$\mathbf{m}_i^t \boxdot \mathbf{x} = \begin{cases} 1 & \text{se } (\bar{\mathbf{m}}_i^t)^t \leq \mathbf{x} \\ 0 & \text{caso contrário} \end{cases} \quad (3.6)$$

Por isso, o operador $\varepsilon_{M_{XX}^t}$ também representa uma erosão no sentido intuitivo.

Seja $\tilde{\mathbf{x}}$ uma versão ruidosa de \mathbf{x} , dizemos que $\tilde{\mathbf{x}}$ é uma versão erodida de \mathbf{x} se, e somente se, $\tilde{\mathbf{x}} \leq \mathbf{x}$, analogamente, dizemos que $\tilde{\mathbf{x}}$ é uma versão dilatada de \mathbf{x} se, e somente se, $\tilde{\mathbf{x}} \geq \mathbf{x}$.

As AMMs exibem duas propriedades desejáveis em uma AM. Uma se refere

a sua capacidade de absoluta de armazenamento, ou seja, pode-se armazenar quantas memórias fundamentais se queira. Outra se refere à convergência da rede em um único passo. Esta última característica pode ser formulada em termos dos pontos fixos das AMMs. Um padrão \mathbf{x} é chamado ponto fixo de uma AMM W_{XX} se, e somente se, $W_{XX} \boxtimes \mathbf{x} = \mathbf{x}$. Similarmente, um padrão \mathbf{x} é chamado ponto fixo de uma AMM M_{XX} se, e somente se, $M_{XX} \boxtimes \mathbf{x} = \mathbf{x}$. Dado um vetor de entrada \mathbf{y} a saída $W_{XX} \boxtimes \mathbf{y}$ é ponto fixo do modelo W_{XX} , similarmente, o vetor de saída $M_{XX} \boxtimes \mathbf{y}$ é ponto fixo de M_{XX} .

Todavia, AMMs armazenam uma grande quantidade de memórias falsas ou espúrias além de exibirem uma tolerância a erro limitada. Especificamente, a memória W_{XX} exibe tolerância apenas a ruídos erosivos, enquanto, a memória M_{XX} exibe tolerância a ruídos dilativos [73, 84, 92]. Uma análise das propriedades de HMMs pode ser encontrada em [73, 95].

3.2 Memórias associativas morfológicas fuzzy

Memórias associativas morfológicas fuzzy (FMAMs) são memórias associativas morfológicas cujos padrões de entrada e saída e/ou pesos sinápticos representam conjuntos fuzzy [99]. Recentemente, as FMAMs foram aplicadas com sucesso em problemas de reconstrução de imagens e previsão de séries temporais [94, 95].

Em um sentido mais abrangente do que a FMAM, chamamos de *memória associativa fuzzy* (FAM) uma AM descrita por uma rede neural que armazena pares de vetores que representam conjuntos fuzzy.

Se o neurônio de uma rede neural executa um produto do tipo max-C então ele é dito um *neurônio max-C*. Uma FMAM composta por neurônios max-C é denominada como *FMAM max-C* [99]. Valle e Sussner mostraram que vários modelos de FAMs pertencem às classes das FMAMs max-C, tais como a famosa FAM de Kosko, FAM implicativa (IFAM), FAM generalizada (GFAM) de Chung e Lee, FAM de Junbo e max-min FAM com limiar de Liu [6, 15, 51, 54, 93, 99].

Considere uma FMAM max-C de uma única camada com m neurônios max-C. Podemos representar este modelo pela seguinte equação:

$$\mathbf{y} = W(\mathbf{x}) = W \circ \mathbf{x} \quad (3.7)$$

onde $W \in [0, 1]^{m \times n}$ representa a matriz de pesos sinápticos e $\mathbf{x} \in [0, 1]^n$ e $\mathbf{y} \in$

$[0, 1]^m$ são os padrões de entrada e saída fuzzy, respectivamente. Referimo-nos a tal rede como FMAM max-C \mathcal{W} .

Seja uma FMAM max-C \mathcal{W} tal que a correspondente conjunção fuzzy C executa uma dilatação fuzzy algébrica no segundo argumento. Recorde da Seção 2.3 que $\mathcal{W}(\mathbf{x})$ também representará uma dilatação fuzzy algébrica.

Em vista da Proposição 2, uma erosão $\mathcal{A} : [0, 1]^m \rightarrow [0, 1]^n$ pode ser construída, tal que $(\mathcal{A}, \mathcal{W})$ formam uma adjunção. A FMAM adjunta de \mathcal{W} é definida como um modelo de AM que corresponde ao mapeamento \mathcal{A} . O teorema a seguir caracteriza as FMAMs adjuntas às FMAMs max-C [99].

Teorema 4. *Se \mathcal{W} denota a FMAM max-C \mathcal{W} dada pela Eq. 3.7. Seja N uma negação fuzzy. Dado um padrão de entrada $\mathbf{y} \in [0, 1]^m$, a FMAM adjunta de \mathcal{W} computa o padrão de saída $\mathbf{x} \in [0, 1]^n$ em termos da seguinte equação, onde $M = [m_{ij}] = [N(w_{ji})] = N(W)$:*

$$\mathbf{x} = \mathcal{A}(\mathbf{y}) = M^T \bullet \mathbf{y} \quad (3.8)$$

Um resultado similar ao Teorema 4 pode ser produzido para FMAMs de uma camada que representam uma erosão fuzzy algébrica, tais que cada neurônio da rede executa um produto min-D. Estas memórias associativas fuzzy pertencem à classe das FMAMs min-D. A demonstração deste resultado e mais informações sobre as FMAMs min-D podem ser encontradas em [99].

A seguir será apresentada uma classe geral de estratégia de treinamento (ou armazenamento) para FMAMs max-C chamada *aprendizagem por adjunção* (ou *aprendizagem fuzzy implicativo*).

Sejam $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \dots, k\}$ o conjunto de memórias fundamentais, e $X \in [0, 1]^{n \times k}$ e $Y \in [0, 1]^{m \times k}$ as matrizes cujas colunas são os vetores \mathbf{x}^ξ e \mathbf{y}^ξ , respectivamente. Seja ainda $\mathcal{D}_X : [0, 1]^{m \times n} \rightarrow [0, 1]^{m \times k}$ o operador definido por:

$$\mathcal{D}_X(W) = W \circ X \quad (3.9)$$

O operador \mathcal{D}_X representa uma dilatação para todo $X \in [0, 1]^{n \times k}$ se, e somente se, o operador $C(\cdot, x)$ do produto max-C for uma dilatação para todo $x \in [0, 1]$, conforme o Teorema 19 de [99].

Considere \mathcal{D}_X sendo uma dilatação e a FMAM max-C dada pela Eq. 3.7. Pela Proposição 2 existe uma única erosão $\mathcal{E}_X : [0, 1]^{m \times k} \rightarrow [0, 1]^{m \times n}$ que forma

uma adjunção com \mathcal{D}_X . Seja a matriz de pesos sinápticos da FMAM max-C dada por $W = \mathcal{E}_X(Y)$, então, W representa o supremo do conjunto das matrizes $V \in [0, 1]^{m \times n}$ tal que $V \circ X \leq Y$, pelo fato de W satisfazer a seguinte Equação [99]:

$$W = \bigvee \{V \in [0, 1]^{m \times n} : \mathcal{D}_X(V) = V \circ X \leq Y\} \quad (3.10)$$

Note que a matriz de pesos sinápticos $W = \mathcal{E}_X(Y)$ é calculada pela erosão \mathcal{E}_X adjunta do operador \mathcal{D}_X , conforme a Proposição 2. Daí o nome de aprendizagem por adjunção.

A Equação 3.10 implica que W é a melhor aproximação abaixo de Y em termos do produto max-C. Note ainda que, se existe $V \in [0, 1]^{m \times n}$ tal que $V \circ X = Y$ como $V \leq W$ temos que $W \circ X = Y$ pela Eq. 3.10 e pelo fato do operador $C(\cdot, x)$ do produto max-C ser crescente. No caso auto-associativo, a FMAM max-C obtida pela aprendizagem por adjunção sempre armazenará e recuperará todas as memórias fundamentais.

O próximo teorema mostra que a matriz de pesos sinápticos $W = \mathcal{E}_X(Y)$ pode ser facilmente computada por um único produto min-I, e sua demonstração pode ser encontrada em [99].

Teorema 5. *Seja $X = [x^1, \dots, x^k] \in [0, 1]^{n \times k}$ e $Y = [y^1, \dots, y^k] \in [0, 1]^{m \times k}$. Considere o operador \mathcal{D}_X dado pela Eq. 3.9 baseado em uma conjunção fuzzy C que representa uma dilatação nos dois argumentos. A matriz de pesos sinápticos $W = \mathcal{E}_X(Y)$ é dada pelo seguinte produto min-I, onde I é a implicação fuzzy que é adjunta a C .*

$$W = Y \circledast X^T \quad (3.11)$$

Em particular, o conjunto das FMAMs max-C produzidas pelo esquema de aprendizagem por adjunção onde C é um t-norma contínua, coincide com a classe das IFAMs [93].

Neste trabalho, falaremos em FMAM de Lukasiewicz para nos referir a FMAM max-C (ou min-D) gerada por aprendizagem adjunta baseada na conjunção (ou disjunção) fuzzy de Lukasiewicz.

3.3 FAM baseada na medida de *subsethood* de Kosko (KS-FAM)

Nesta seção apresentaremos um modelo de memória associativa fuzzy de duas camadas denominada *KS-FAM* [87]. O processamento dos neurônios da primeira camada é baseado na medida de *subsethood* de Kosko [50, 51], que pode ser vista como uma medida de inclusão entre conjuntos fuzzy que não satisfaz a propriedade de herança, isto é, não generaliza o caso crisp. Considerando as definições dos operadores morfológicos fuzzy no sentido intuitivo discutidos na Seção 2.3, se nenhuma restrição quanto a propriedade de herança for assumida, então, o modelo KS-FAM representa uma FMAM. A seguir, será apresentada a medida de *subsethood* de Kosko e algumas análises sobre suas propriedades.

3.3.1 Medida de *subsethood* de Kosko

Kosko propôs uma medida *subsethood* [50, 51] definida em um universo finito Ω , que pode ser vista como uma medida de inclusão que não generaliza o caso crisp. Sejam A e B conjuntos fuzzy definidos em um universo finito Ω , tal que $\text{card}(A) = \sum_{x \in \Omega} \mathbf{a}(x) > 0$, então, a medida de *subsethood* $S(A, B)$ é definida por

$$S(A, B) = 1 - \frac{\sum_{x \in \Omega} 0 \vee (\mathbf{a}(x) - \mathbf{b}(x))}{\sum_{x \in \Omega} \mathbf{a}(x)} = \frac{\sum_{x \in \Omega} \mathbf{a}(x) \wedge \mathbf{b}(x)}{\sum_{x \in \Omega} \mathbf{a}(x)} \quad (3.12)$$

Note que a Equação 3.12 pode ser facilmente generalizada para um domínio contínuo Ω com a ressalva de que $\int_{\Omega} \mathbf{a}(x) dx$ seja finito e maior que zero, ou seja, $0 < \int_{\Omega} \mathbf{a}(x) dx < \infty$:

$$S(A, B) = \frac{\int_{\Omega} (\mathbf{a}(x) \wedge \mathbf{b}(x)) dx}{\int_{\Omega} \mathbf{a}(x) dx} \quad (3.13)$$

Pela Equação 3.13 fica evidente que a medida de *subsethood* calcula o grau de inclusão de um conjunto fuzzy A em um conjunto fuzzy B . Porém, S não mantém a propriedade de herança. Sejam $A, B \subseteq \Omega$ e Inc a medida de inclusão crisp dada por

$$\text{Inc}(A, B) = \begin{cases} 1 & \text{se } A \subseteq B \\ 0 & \text{caso contrário} \end{cases} \quad (3.14)$$

O exemplo 2 ilustra a diferença entre a medida de inclusão crisp (*Inc*) com a de *subsethood*

Exemplo 2. *Sejam*

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

A medida de inclusão crisp $\text{Inc}(A, B) = 0$ pois $A \not\subseteq B$, enquanto $S(A, B) = \frac{2}{3}$.

Note pelo exemplo anterior que todas as coordenadas, *exceto a última* do conjunto binário A , estão contidas em B , portanto, num sentido fuzzy, é razoável pensar que o grau de inclusão de A em B seja $\frac{2}{3}$. Todavia, ao se aplicar a função limiar f definida conforme a Eq. 3.15 obtém-se a seguinte igualdade $f(S(A, B)) = \text{Inc}(A, B)$ no caso crisp.

$$f(x) = \begin{cases} 1 & \text{se } x \geq 1 \\ 0 & \text{caso contrário} \end{cases} \quad (3.15)$$

A partir dessa observação e da Equação 3.6, Sussner verificou que o *produto min* executado por memórias morfológicas auto-associativas M_{XX} pode ser expresso em termos da medida de *subsethood* de Kosko [86]. Sejam $M_{XX} \in \{0, 1\}^{n \times n}$ e $\mathbf{y} \in \{0, 1\}^n$, onde \mathbf{m}_i^t denota a i -ésima linha de M_{XX} . Para simplificarmos, denotaremos $(\mathbf{m}_i^t)^t$ por \mathbf{m}_i . O símbolo $\bar{\mathbf{m}}_i$ denota a negação fuzzy padrão de \mathbf{m}_i definida como na Equação 3.5. Seja f definida como na Eq. 3.15, a seguinte igualdade é obtida [86]:

$$(M_{XX} \boxtimes \mathbf{y})_i = \mathbf{m}_i^t \boxtimes \mathbf{y} = f(S(\bar{\mathbf{m}}_i, \mathbf{y})) \quad \text{para } i = 1, \dots, n. \quad (3.16)$$

De maneira mais geral, pode ser mostrado que $1 \wedge (\mathbf{m}^t \boxtimes \mathbf{y}) = f(S(\bar{\mathbf{m}}, \mathbf{y}))$ para todo $\mathbf{m}, \mathbf{y} \in \{0, 1\}^n$. Basta verificar, de maneira completamente análoga ao exposto por Sussner em [86], as seguintes equivalências:

$$\mathbf{m}^t \boxtimes \mathbf{x} \geq 1 \Leftrightarrow \mathbf{m}_i + \mathbf{x}_i \geq 1 \quad \forall i = 1, \dots, n \quad (3.17)$$

$$\Leftrightarrow \mathbf{x}_i \geq 1 - \mathbf{m}_i \quad \forall i = 1, \dots, n \quad (3.18)$$

$$\Leftrightarrow \mathbf{x} \geq \bar{\mathbf{m}} \quad (3.19)$$

Levando em conta a relação exposta acima da medida de *subsethood* de Kosko

com o produto min, vamos definir o *produto min fuzzy* $\tilde{\boxtimes} : [0, 1]^{m \times n} \times [0, 1]^{n \times p} \rightarrow [0, 1]^{m \times p}$. Sejam $M \in [0, 1]^{m \times n}$ e $X \in [0, 1]^{n \times p}$. A matriz Y formada pelo produto min fuzzy de M com X é dada por

$$y_{ij} = S(\tilde{\mathbf{m}}_i, \mathbf{x}^j) \text{ para todo } i = 1, \dots, m \text{ e } j = 1, \dots, p \quad (3.20)$$

onde $\tilde{\mathbf{m}}_i$ denota a transposta da i -ésima linha de M e \mathbf{x}^j a j -ésima coluna de X .

Exemplo 3. Considere a matriz $X \in \{0, 1\}^{4 \times 2}$ e a matriz M_{XX} gerada pelas colunas de X como abaixo. Este exemplo ilustra tanto a ação do produto min quanto a do produto min fuzzy entre a matriz M_{XX} com o vetor $\mathbf{x} \in \{0, 1\}^{4 \times 1}$ abaixo, onde $\mathbf{y} = M_{XX} \tilde{\boxtimes} \mathbf{x}$ e $\mathbf{z} = M_{XX} \tilde{\boxtimes} \mathbf{x}$.

$$X = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, M_{XX} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \mathbf{z} = \begin{bmatrix} 0.667 \\ 0.667 \\ 1.000 \\ 1.000 \end{bmatrix}$$

Exemplo 4. Uma visualização do efeito do produto min fuzzy é dado neste outro exemplo. Considere as oito imagens binárias de tamanho 64×64 , lidas da esquerda para direita e de cima para baixo, exibidas na Figura 3.1. Estas imagens foram convertidas respectivamente nos padrões binários $\mathbf{x}^1, \dots, \mathbf{x}^8 \in \{0, 1\}^{4096}$ e utilizadas na construção da matriz de pesos sinápticos $M_{XX} \in \{0, 1\}^{4096 \times 4096}$. Seja $\tilde{\mathbf{x}}^1$ o padrão corresponde à versão corrompida de \mathbf{x}^1 (cameraman) retratada na Figura 4(a), a interpretação visual do produto min fuzzy de M_{XX} e $\tilde{\mathbf{x}}^1$ é dada pela Figura 4(b).

Baseado no produto min fuzzy e num algoritmo de pós-processamento \mathcal{T} que executa uma defuzzificação, Sussner propôs uma abordagem bem sucedida para uma AMM binária (\mathcal{T} MAM) que exhibe maior tolerância a ruídos arbitrários do que a MAM M_{XX} [86]. Considere $\mathbf{x}^1, \dots, \mathbf{x}^k \in \{0, 1\}^{n \times k}$ os padrões binários utilizados na construção da matriz M_{XX} . Dada uma entrada $\mathbf{x} \in \{0, 1\}^n$, a fase de recordação da \mathcal{T} MAM é dada pelo seguinte esquema:

$$\text{entrada } \mathbf{x} \rightarrow M_{XX} \tilde{\boxtimes} \mathbf{x} \rightarrow \text{Pós-proc. } \mathcal{T} \rightarrow \text{saída } \mathbf{y}.$$



Figura 3.1: Oito imagens binárias que foram usadas na construção da matriz M_{XX} .

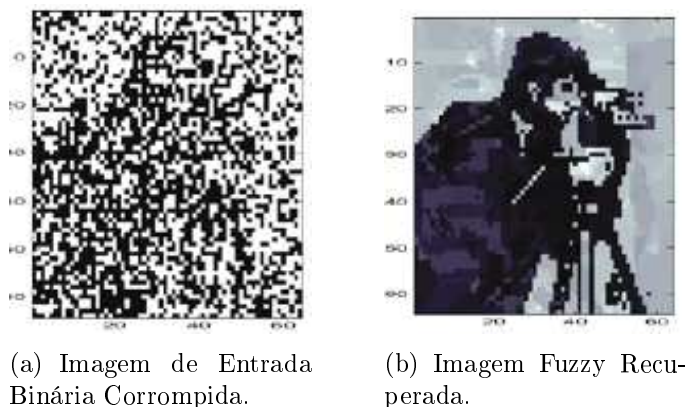


Figura 3.2: A imagem fuzzy na direita foi produzida a partir do vetor referente da imagem corrompida à esquerda através do produto min fuzzy com M_{XX} .

Uma outra abordagem consiste de uma MAM binária de duas camadas aplicada tanto para o caso auto-associativo quanto heteroassociativo, que reduz o número de memória espúrias comparado às MAMs clássicas [85]. Uma FAM baseada na medida de *subsethood* que será apresentada a seguir, foi proposta inicialmente para tentar combinar as vantagens das duas abordagens anteriores.

3.3.2 FAM baseada na medida de *subsethood* de Kosko para o caso binário

Como já mencionado, as memórias W_{XX} e M_{XX} exibem duas características indesejáveis: armazenam uma grande quantidade de memórias espúrias e possuem uma

tolerância a erro limitada. Recentemente alguns modelos de MAMs foram propostos no intuito de superar essas dificuldades. Por exemplo, Sussner introduziu uma memória associativa morfológica de duas camadas que exhibe uma quantidade reduzida de memórias espúrias [86]. Outra abordagem utilizou o *produto min fuzzy* junto a um algoritmo de pós-processamento que prove maior tolerância a ruído arbitrários [85].

A seguir, será apresentado o modelo original baseado na medida de *subsethood* de Kosko proposto para o caso binário que, em experimentos preliminares com problemas de reconstrução de imagens binárias, obteve bons resultados [87].

Sejam $X \in \{0, 1\}^{n \times k}$, $Y \in \{0, 1\}^{m \times k}$ e $Z = [z^1, \dots, z^k] \in \{0, 1\}^{p \times k}$ tal que $\bigvee_{\xi=1}^k z^\xi = \mathbf{1} = (1, \dots, 1)^t \in [0, 1]^p$ e cada coluna z^ξ satisfaz as condições de $z^\xi \not\leq z^\gamma$ e $z^\xi \wedge z^\gamma = \mathbf{0} = (0, \dots, 0)^t \in [0, 1]^p$ para todo $\gamma \neq \xi$. Dado uma entrada \mathbf{x} , uma saída \mathbf{y} é produzida de acordo com as seguintes equações:

$$\mathbf{w} = \mathbf{h}(M_X^{XZ} \boxtimes \mathbf{x}) \quad (3.21)$$

$$\mathbf{y} = W_{ZY} \boxtimes \mathbf{w} \quad (3.22)$$

onde $M_X^{XZ} = M_{XZ} \boxtimes M_{XX}$ e a função $\mathbf{h} : \mathbb{R}^p \rightarrow \{0, 1\}^p$ é definida como $\mathbf{h}(\mathbf{x}) = (h(x_1), \dots, h(x_p))^t$ tal que

$$h(x_i) = \begin{cases} 1 & \text{se } x_i = \bigvee_{j=1}^p x_j \\ 0 & \text{caso contrário} \end{cases}, \text{ para } i = 1, \dots, p \quad (3.23)$$

Esse modelo pode ser visto como uma rede neural de duas camadas, cuja primeira calcula um mapeamento fuzzy da entrada por meio de um *produto min fuzzy*, seguida de uma função \mathbf{h} que executa uma competição entre os neurônios desta camada de modo que apenas os neurônios com a maior saída são ativados.

3.3.3 KS-FAM: Definição e propriedades

Nesta subseção será generalizado o modelo original baseado na medida de *subsethood* de Kosko de modo que o mesmo seja capaz de armazenar pares de entrada e saída que correspondem a conjuntos fuzzy.

O lema a seguir simplifica consideravelmente o modelo em aspectos computacionais permitindo uma generalização consistente do modelo para o caso fuzzy.

Lema 4. *Sejam $X = [\mathbf{x}^1, \dots, \mathbf{x}^k] \in [0, 1]^{n \times k}$ e $Z = [\mathbf{z}^1, \dots, \mathbf{z}^k] \in \{0, 1\}^{m \times k}$ tal que $\bigvee_{\xi=1}^k \mathbf{z}^\xi = \mathbf{1}$ e cada coluna \mathbf{z}^ξ satisfaz as condições de $\mathbf{z}^\xi \not\leq \mathbf{z}^\gamma$ e $\mathbf{z}^\xi \wedge \mathbf{z}^\gamma = \mathbf{0}$ para todo $\gamma \neq \xi$. Então $M_X^{XZ} = M_{XZ}$.*

Demonstração. Primeiramente, será mostrado que $(M_X^{XZ})_{ij} \leq (M_{XZ})_{ij}$ para $i = 1, \dots, m$ e $j = 1, \dots, n$. Usando o fato que $(M_{XX})_{kk} = 0$ para $k = 1, \dots, n$, segue que

$$(M_X^{XZ})_{ij} = \bigwedge_{p=1}^n (M_{XZ})_{ip} + (M_{XX})_{pj} \leq (M_{XZ})_{ij} + (M_{XX})_{jj} = (M_{XZ})_{ij}$$

Logo $(M_X^{XZ})_{ij} \leq (M_{XZ})_{ij}$.

Em seguida será verificado que $(M_X^{XZ})_{ij} \geq (M_{XZ})_{ij}$ donde será concluído o resultado.

Como $\bigvee_{\xi=1}^k \mathbf{z}^\xi = \mathbf{1}$ e $\mathbf{z}^\xi \wedge \mathbf{z}^\gamma = \mathbf{0}$ para todo $\gamma \neq \xi$ então para cada $i = 1, \dots, m$ existe ξ^i tal que $z_i^{\xi^i} = 1$ e $z_i^\gamma = 0$ para $\gamma \neq \xi^i$, o que implica que $(M_{XZ})_{ip} = \bigvee_{\gamma=1}^k (z_i^\gamma - x_p^\gamma) = 1 - x_p^{\xi^i}$ para todo $p = 1, \dots, n$. Daí segue que

$$\begin{aligned} (M_X^{XZ})_{ij} &= \bigwedge_{p=1}^n (M_{XZ})_{ip} + (M_{XX})_{pj} \\ &= \bigwedge_{p=1}^n 1 - x_p^{\xi^i} + \left(\bigvee_{\gamma=1}^k x_p^\gamma - x_j^\gamma \right) \\ &\geq \bigwedge_{p=1}^n 1 - x_p^{\xi^i} + x_p^{\xi^i} - x_j^{\xi^i} \\ &\geq \bigwedge_{p=1}^n 1 - x_j^{\xi^i} \\ &\geq 1 - x_j^{\xi^i} \\ &\geq (M_{XZ})_{ij} \end{aligned}$$

Portanto $(M_X^{XZ})_{ij} = (M_{XZ})_{ij}$ para todos $i = 1, \dots, m$ e $j = 1, \dots, n$. □

Suponha que se deseja armazenar o seguinte conjunto de pares de entrada e saída $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) \in [0, 1]^n \times [0, 1]^m : \xi = 1, \dots, k\}$. Sejam $X = [\mathbf{x}^1, \dots, \mathbf{x}^k]$ a matriz cujas colunas são os vetores de entrada $\mathbf{x}^\xi \in [0, 1]^n$, $Y = [\mathbf{y}^1, \dots, \mathbf{y}^k]$ a matriz cujas colunas são os vetores de saída $\mathbf{y}^\xi \in [0, 1]^m$ e a matriz $Z \in \{0, 1\}^{p \times k}$ dada

como no lema 4. O modelo original pode ser generalizado da seguinte maneira: dada uma entrada $\mathbf{x} \in [0, 1]^n$, uma saída $\mathbf{y} \in [0, 1]^m$ é produzida de acordo com as seguintes equações:

$$\mathbf{w} = \mathbf{h}(M_{XZ} \boxtimes \mathbf{x}) \quad (3.24)$$

$$\mathbf{y} = W_{ZY} \boxtimes \mathbf{w}. \quad (3.25)$$

Note que o lema 4 garante que o modelo generalizado estará bem definido, uma vez que $M_{XZ} \in [0, 1]^{p \times n}$, pois $(M_{XZ})_{ij} = 1 - x_j^{\xi^i} \in [0, 1]$, onde ξ^i é tal que $z_i^{\xi^i} = 1$ e $z_i^\gamma = 0$ para $\gamma \neq \xi^i$. Portanto o *produto min fuzzy*, entre M_{XZ} e uma entrada $\mathbf{x} \in [0, 1]^n$ qualquer, pode ser computado.

O Teorema 6 garante que a memória W_{ZY} é capaz de recordar perfeitamente a saída \mathbf{y}^ξ quando apresentando o vetor \mathbf{z}^ξ como entrada, para $i = 1, \dots, k$.

Teorema 6. *Sejam $Z \in \{0, 1\}^{p \times k}$ como no lema 4 e $Y \in [0, 1]^{m \times k}$. Então $W_{ZY} \boxtimes \mathbf{z}^\xi = \mathbf{y}^\xi$ para todo $\xi = 1, \dots, k$.*

Demonstração. Por um lado, segue que $(W_{ZY} \boxtimes \mathbf{z}^\xi)_i \leq y_i^\xi$ pois

$$\begin{aligned} (W_{ZY} \boxtimes \mathbf{z}^\xi)_i &= \bigvee_{t=1}^p (W_{ZY})_{it} + z_t^\xi \\ &= \bigvee_{t=1}^p \left(\bigwedge_{\gamma=1}^k y_i^\gamma - z_t^\gamma \right) + z_t^\xi \\ &\leq \bigvee_{t=1}^p y_i^\xi - z_t^\xi + z_t^\xi \\ &\leq y_i^\xi \end{aligned}$$

Por outro lado, temos que $(W_{ZY} \boxtimes \mathbf{z}^\xi)_i \geq y_i^\xi$, pois como existe t^* tal que $z_{t^*}^\xi = 1$ e $z_{t^*}^\gamma = 0$ para $\gamma \neq \xi$ pelas hipóteses de que $\mathbf{z}^\xi \not\leq \mathbf{z}^\gamma$ e $\mathbf{z}^\xi \wedge \mathbf{z}^\gamma = \mathbf{0}$ para $\xi \neq \gamma$ então

$$\begin{aligned}
(W_{ZY} \boxtimes \mathbf{z}^\xi)_i &= \bigvee_{t=1}^p (W_{ZY})_{it} + z_t^\xi \\
&= \bigvee_{t=1}^p \left(\bigwedge_{\gamma=1}^k y_i^\gamma - z_t^\gamma \right) + z_t^\xi \\
&\geq \left(\bigwedge_{\gamma=1}^k y_i^\gamma - z_{t^*}^\gamma \right) + z_{t^*}^\xi \\
&\geq \left(\left(\bigwedge_{\gamma=1; \gamma \neq \xi}^k y_i^\gamma \right) \wedge (y_i^\xi - 1) \right) + 1 \\
&\geq (y_i^\xi - 1) + 1 \\
&\geq y_i^\xi
\end{aligned}$$

Como $(W_{ZY} \boxtimes \mathbf{z}^\xi)_i \leq y_i^\xi$ e $(W_{ZY} \boxtimes \mathbf{z}^\xi)_i \geq y_i^\xi$ para todo $i = 1, \dots, m$ conclui-se que $(W_{ZY} \boxtimes \mathbf{z}^\xi) = \mathbf{y}^\xi$ \square

O Corolário 1 expressa uma consequência direta do Teorema 6, cuja demonstração utiliza apenas o fato do produto max (\boxtimes) comutar com o supremo (\vee) (lembre que $\delta_{W_{ZY}^t}$ é uma dilatação no sentido algébrico).

Corolário 1. *Sejam Y e Z como no Teorema 6, então $W_{ZY} \boxtimes (\bigvee_{\xi \in \Theta} \mathbf{z}^\xi) = \bigvee_{\xi \in \Theta} \mathbf{y}^\xi$, onde Θ é um subconjunto de $\{1, \dots, k\}$ não vazio.*

O Teorema 7 caracteriza a capacidade de armazenamento da memória associativa fuzzy baseada na medida de *subsethood* de Kosko (KS-FAM).

Teorema 7. *Sejam $Y = [\mathbf{y}^1, \dots, \mathbf{y}^k] \in [0, 1]^{m \times k}$, $X = [\mathbf{x}^1, \dots, \mathbf{x}^k] \in [0, 1]^{n \times k}$ tal que $\mathbf{x}^\gamma \not\leq \mathbf{x}^\xi$ para todo $\xi \neq \gamma$ e $Z = [\mathbf{z}^1, \dots, \mathbf{z}^k] \in \{0, 1\}^{p \times k}$ como no lema 4. Então $W_{ZY} \boxtimes \mathbf{h}(M_{XZ} \tilde{\boxtimes} \mathbf{x}^\xi) = \mathbf{y}^\xi$ para todo $\xi = 1, \dots, k$.*

Demonstração. Dado $\xi \in \{1, \dots, k\}$. Seja $i \in \{1, \dots, p\}$ temos que $z_i^\xi = 1$ ou $z_i^\xi = 0$.

No caso onde $z_i^\xi = 1$ temos que $z_j^\gamma = 0$ para $\gamma \neq \xi$. Isso implica que $(M_{XZ})_{ij} = z_i^\xi - x_j^\xi = 1 - x_j^\xi \Rightarrow \bar{m}_{ij} = x_j^\xi$ para todo $j = 1, \dots, n$. Daí $(M_{XZ} \tilde{\boxtimes} \mathbf{x}^\xi)_i = 1$, pois

$$(M_{XZ} \tilde{\boxtimes} \mathbf{x}^\xi)_i = \frac{\sum_{j=1}^n \bar{m}_{ij} \wedge x_j^\xi}{\sum_{j=1}^n \bar{m}_{ij}} = \frac{\sum_{j=1}^n x_j^\xi \wedge x_j^\xi}{\sum_{j=1}^n x_j^\xi} = 1$$

Agora, se $z_i^\xi = 0$ então existe γ tal que $z_i^\gamma = 1$ pois $\bigvee_{\lambda=1}^k z^\lambda = \mathbf{1}$. Isso implica que $(M_{XZ})_{ij} = z_i^\gamma - x_j^\gamma = 1 - x_j^\gamma \Rightarrow \bar{m}_{ij} = x_j^\gamma$ para todo $j = 1, \dots, n$. Além disso, como $\mathbf{x}^\gamma \not\leq \mathbf{x}^\xi$ então existe j^* tal que $x_{j^*}^\gamma > x_{j^*}^\xi \geq 0$. Logo, $(M_{XZ} \tilde{\boxminus} \mathbf{x}^\xi)_i < 1$, pois

$$\begin{aligned}
 (M_{XZ} \tilde{\boxminus} \mathbf{x}^\xi)_i &= \frac{\sum_{j=1}^n \bar{m}_{ij} \wedge x_j^\xi}{\sum_{j=1}^n \bar{m}_{ij}} \\
 &= \frac{\sum_{j=1}^n x_j^\gamma \wedge x_j^\xi}{\sum_{j=1}^n x_j^\gamma} \\
 &= \frac{(\sum_{j=1; j \neq j^*}^n x_j^\gamma \wedge x_j^\xi) + x_{j^*}^\xi}{\sum_{j=1}^n x_j^\gamma} \\
 &\leq \frac{(\sum_{j=1; j \neq j^*}^n x_j^\gamma) + x_{j^*}^\xi}{\sum_{j=1}^n x_j^\gamma} \\
 &< \frac{\sum_{j=1}^n x_j^\gamma}{\sum_{j=1}^n x_j^\gamma} = 1
 \end{aligned}$$

Como existe i^* tal que $z_{i^*}^\xi = 1$, pois, $z^\xi \not\leq z^\gamma$ para $\xi \neq \gamma$, então, do resultado acima segue que $\bigvee_{i=1}^p (M_{XZ} \tilde{\boxminus} \mathbf{x}^\xi)_i = 1$ e, portanto, $\mathbf{h}(M_{XZ} \tilde{\boxminus} \mathbf{x}^\xi) = z^\xi$ pela Equação 3.23.

Pelo Teorema 6 segue que $W_{ZY} \boxtimes z^\xi = \mathbf{y}^\xi$, o que conclui a demonstração. \square

O próximo exemplo ilustra a capacidade de armazenamento do modelo KS-FAM, assim como, sua tolerância a presença de ruído e a padrões incompletos.

Exemplo 5. *Seja $\{\mathbf{x}^1, \dots, \mathbf{x}^6\}$, respectivamente, o conjunto das imagens da primeira linha da Figura 3.3 lidas da esquerda para direita. Utilizamos uma KS-FAM para armazenar os pares $\{(\mathbf{x}^\xi, \mathbf{x}^\xi) : \xi = 1, \dots, 6\}$. A memória resultante armazenou perfeitamente cada uma das memórias fundamentais, conforme o Teorema 7. Na segunda e terceira linha da Figura 3.3, cada coluna contém exemplos de versões de um determinado \mathbf{x}^ξ , respectivamente, com adição de ruído gaussiano e incompleto, que produziram a saída desejada ao serem apresentados a rede.*

O último resultado apresentado nesta subseção vem do Teorema 8 que caracteriza a saída produzida por uma KS-FAM dada uma entrada \mathbf{x} qualquer. Note que o número de memórias espúrias armazenadas é bem reduzido se comparado aos modelos clássicos de MAMs. Outro fato importante a observar é que esse resultado foi propiciado devido à aplicação da função \mathbf{h} na primeira camada.

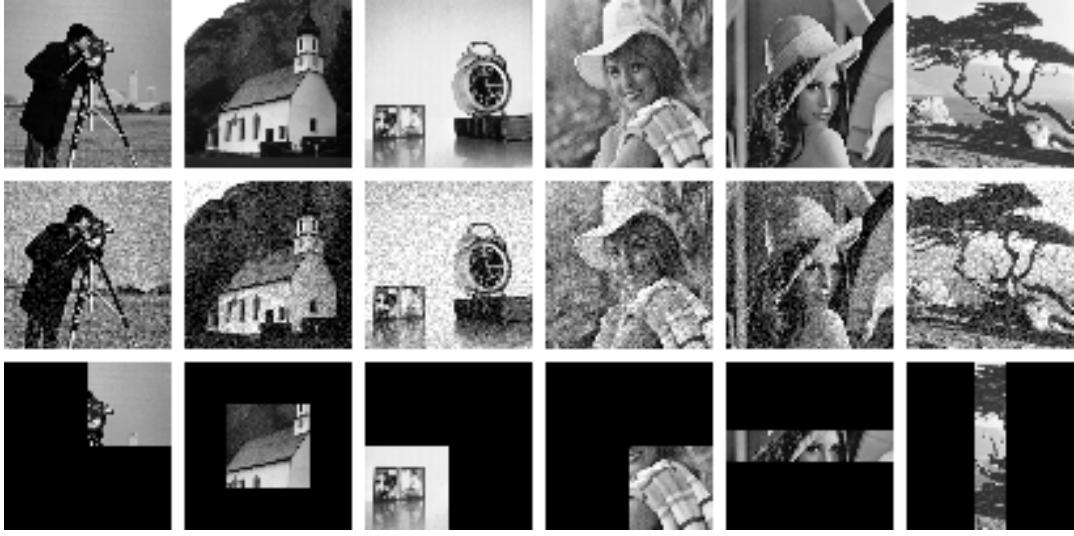


Figura 3.3: A primeira linha exibe as imagens utilizadas como memórias fundamentais para gerar uma KS-FAM. Para cada coluna, a segunda e terceira linha contêm entradas apresentadas à memória KS-FAM cuja saída é exibida na primeira linha.

Teorema 8. *Sejam $X \in [0, 1]^{n \times k}$, $Y \in [0, 1]^{m \times k}$ e $Z = [z^1, \dots, z^k] \in \{0, 1\}^{p \times k}$ tal que $\bigvee_{\xi=1}^k z^\xi = \mathbf{1}$ e cada coluna z^ξ satisfaz as condições de $z^\xi \not\leq z^\gamma$ e $z^\xi \wedge z^\gamma = \mathbf{0}$ para todo $\gamma \neq \xi$. Então, para cada $\mathbf{x} \in [0, 1]^n$ existe um único subconjunto de índices $\Theta_x \subseteq \{1, \dots, k\}$, $\Theta_x \neq \emptyset$, tal que $\mathbf{h}(M_{XZ} \tilde{\nabla} \mathbf{x}) = \bigvee_{\xi \in \Theta_x} z^\xi$. Além disso, $W_{ZY} \boxtimes \mathbf{h}(M_{XZ} \tilde{\nabla} \mathbf{x}) = \bigvee_{\xi \in \Theta_x} y^\xi$.*

Demonstração. Sejam $\alpha = \bigvee_{j=1}^n (M_{XZ} \tilde{\nabla} \mathbf{x})_j$ o maior valor do produto min fuzzy $M_{XZ} \tilde{\nabla} \mathbf{x}$, $T = \{j : (\mathbf{h}(M_{XZ} \tilde{\nabla} \mathbf{x}))_j = 1\}$ (o subconjunto de $\{1, \dots, m\}$ que satisfaz o maior valor α), $\Theta_x = \{\xi : z_j^\xi = 1 \text{ para } j \in T\}$ e $S = \{j : (\bigvee_{\xi \in \Theta_x} z^\xi)_j = 1\}$.

Lembre que $[0, 1]$ representa uma corrente. Como $(M_{XZ} \tilde{\nabla} \mathbf{x})_j \in [0, 1]$ para todo $j = 1, \dots, m$, segue que existe um índice $j \in \{1, \dots, m\}$ tal que $\alpha = (M_{XZ} \tilde{\nabla} \mathbf{x})_j$ [7]. Logo tem-se $T \neq \emptyset$. A prova da existência de Θ_x segue do fato que para cada $j \in T$ existe um índice ξ_j tal que $z_j^{\xi_j} = 1$ pela hipótese que $\bigvee_{\xi=1}^k z^\xi = \mathbf{1}$ e, portanto, $\Theta_x \neq \emptyset$.

Para provar a unicidade de Θ_x basta então verificar que $S = T$. A sequência de implicações a seguir conclui que $T \subseteq S$.

$$\xi_j \in \Theta_x \Rightarrow (\bigvee_{\xi \in \Theta_x} z^\xi)_j = 1 \Rightarrow j \in S \Rightarrow T \subseteq S$$

Por outro lado, cada $j \in S$ resulta na existência de um índice $i^* \in T$ de acordo com a sequência de implicações dadas por

$$j \in S \Rightarrow \exists \xi_j \in \Theta_x \text{ com } z_j^{\xi_j} = 1 \Rightarrow \exists i^* \in T \text{ com } z_{i^*}^{\xi_j} = 1 \text{ e } (M_{XZ} \tilde{\boxtimes} \mathbf{x})_{i^*} = \alpha$$

Note que se $z_i^\xi = 1$ então $(M_{XZ})_{iq} = z_i^\xi - x_q^\xi$ para todo $q = 1, \dots, n$ pelo fato de $z_i^\gamma = 0$ para todo $\gamma \neq \xi$. Logo $(M_{XZ})_{i^*q} = z_{i^*}^{\xi_j} - x_q^{\xi_j} = z_j^{\xi_j} - x_q^{\xi_j} = (M_{XZ})_{jp}$ para todo $q = 1, \dots, n$, donde temos que

$$\alpha = (M_{XZ} \tilde{\boxtimes} \mathbf{x})_{i^*} = (M_{XZ} \tilde{\boxtimes} \mathbf{x})_j \Rightarrow (\mathbf{h}(M_{XZ} \tilde{\boxtimes} \mathbf{x}))_j = 1 \Rightarrow j \in T \Rightarrow S \subseteq T \quad (3.26)$$

Portanto, conclui-se que $S = T$.

A igualdade $S = T$ prova que $\mathbf{h}(M_{XZ} \tilde{\boxtimes} \mathbf{x}) = \bigvee_{\xi \in \Theta_x} \mathbf{z}^\xi$ pelas próprias definições de S e T . Recorrendo ao Colorário 1 conclui-se que $W_{ZY} \boxtimes \mathbf{h}(M_{XZ} \tilde{\boxtimes} \mathbf{x}) = \bigvee_{\xi \in \Theta_x} \mathbf{y}^\xi$.

Por fim, a demonstração da unicidade de Θ_x decorre diretamente da unicidade de T . \square

3.4 Experimentos com KS-FAM

Nesta seção compararemos a KS-FAM com outras memórias associativas. Iniciaremos analisando a porcentagem de acerto do modelo KS-FAM com imagens binárias em um experimento idêntico ao realizado por Sussner em [86]. Uma vez que uma imagem em tons de cinza pode ser identificada com um conjunto fuzzy. Prosseguiremos sondando o desempenho da KS-FAM em algumas simulações utilizando imagens em tons de cinza (fuzzy), e comparando-as com outros modelos de memórias associativas em termos da raiz quadrada do erro médio quadrático normalizado (NRMSE), que é calculado através da equação

$$\frac{\|\mathbf{y} - \bar{\mathbf{y}}\|_2}{\|\mathbf{y}\|_2} \quad (3.27)$$

onde \mathbf{y} e $\bar{\mathbf{y}}$ denotam respectivamente a saída desejada e a saída produzida pelo modelo, e $\|\cdot\|_2$ a norma euclidiana.

3.4.1 Imagens binárias

Nesta subseção repetiremos o experimento conduzido por Sussner em [86] utilizando o modelo KS-FAM, a fim de comparar a porcentagem de acerto deste modelo com as de outros, presentes na mesma publicação. Em particular, considere os padrões $\mathbf{x}^1, \dots, \mathbf{x}^{10} \in \{0, 1\}^{49}$ gerados a partir da aplicação do método “*row scan*” em cada imagem binária exibida na Figura 3.4, tal que $\mathbf{x}^1, \dots, \mathbf{x}^5$ correspondem respectivamente às vogais maiúsculas de A a U, enquanto $\mathbf{x}^6, \dots, \mathbf{x}^{10}$ às vogais minúsculas a, e, i, o, e u. Os pares das vogais maiúsculas $(\mathbf{x}^1, \mathbf{x}^1), \dots, (\mathbf{x}^5, \mathbf{x}^5)$ foram armazenados usando uma MAM W_{xx} , uma rede de Hopfield discreta [38] e uma memória associativa dinâmica com armazenamento por projeção (“projection-recorded DAM”) [34]. Além disso, os pares $(\mathbf{x}^1, \mathbf{x}^6), \dots, (\mathbf{x}^5, \mathbf{x}^{10})$, que associam cada vogal maiúscula a sua respectiva minúscula, foram armazenados utilizando uma memória associativa bidirecional com armazenamento por correlação (“correlation-recorded BAM”) [34], uma \mathcal{T} MAM e uma KS-FAM com $Z = I \in \{0, 1\}^{5 \times 5}$, onde I denota a matriz identidade.

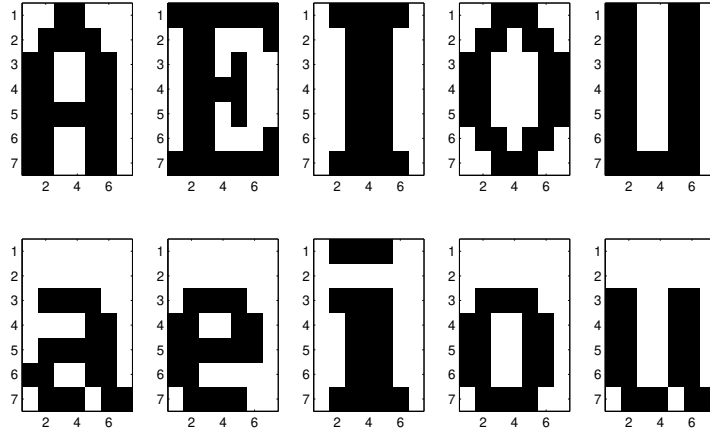


Figura 3.4: Dez imagens binárias de dimensões 7×7 que foram utilizadas como pares de entrada-saída na construção das memórias testadas.

O gráfico ilustrado na Figura 3.5 exibe a porcentagem média de acertos dos modelos mencionados anteriormente ao apresentar entradas $\tilde{\mathbf{x}}^\xi$, que correspondem a versões de \mathbf{x}^ξ com valores de pixels (ou bits) invertidos aleatoriamente por uma certa probabilidade p , em 1000 experimentos para cada $\xi = 1, \dots, 5$.

O gráfico na Figura 3.5 revela que o modelo KS-FAM obteve um ótimo de-

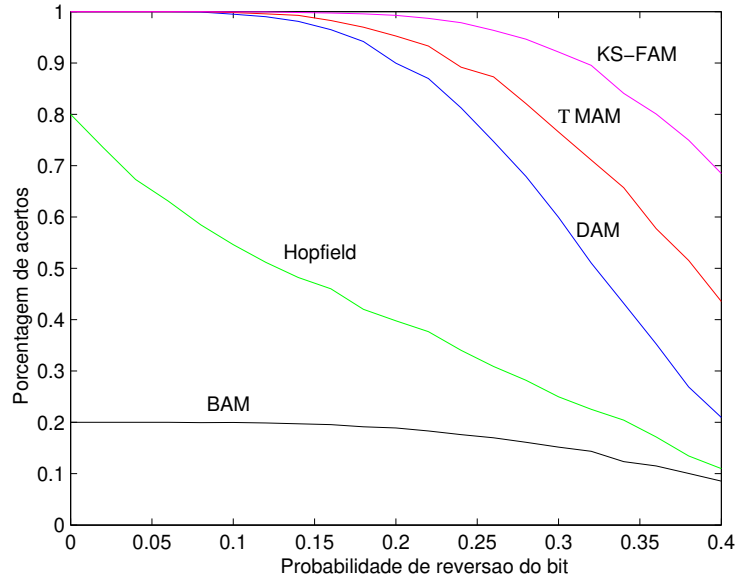


Figura 3.5: Percentual de perfeito reconhecimento ao apresentar 1000 versões corrompidas de cada vogal maiúscula contendo reversões aleatórias de bits com probabilidade p .

sempenho em relação às outras memórias, já que sua média de acertos, para cada probabilidade de reversão de bits p testada, ficou acima das demais, como observado nas curvas traçadas. Avaliando as curvas referentes aos outros modelos, verificamos que neste experimento a T MAM e a “projection-recorded DAM” exibiram tolerância satisfatória com respeito a inversão de bits, em contraste com o modelo “correlation-recorded BAM” que exibiu baixa percentagem de acertos médios para toda probabilidade p e, nem ao menos foi capaz de armazenar todos os pares de memórias fundamentais. A rede de Hopfield discreta armazenou cerca de 80% dos pares das vogais maiúsculas, porém, sua percentagem de acertos médio cai rapidamente conforme a probabilidade p aumenta.

3.4.2 Imagens em tons de cinza

A Figura 3.6 exibe quatro imagens de tamanho 64×64 com 256 tons de cinza, que representam versões reduzidas das que estão contidas na base de dados do Grupo de Visão Computacional da Universidade de Granada na Espanha. Para cada imagem foi gerado um vetor \mathbf{x}^ξ de tamanho 4096, utilizando o método *row-scan*, para $\xi = 1, \dots, 4$. Os vetores resultantes $\mathbf{x}^1, \dots, \mathbf{x}^4$ foram utilizados em todas as

simulações restantes deste capítulo, e correspondem respectivamente às imagens da Figura 3.6 lidas da esquerda para direita.



Figura 3.6: As quatro imagens, da esquerda para direita, correspondem aos padrões $\mathbf{x}^1, \dots, \mathbf{x}^4$ utilizados nos experimentos de reconstrução de imagens.

Na fase de construção da memória KS-FAM, consideramos \mathbf{Z} como sendo a matriz identidade 4×4 . Como sugerido pelo Teorema 7, o modelo KS-FAM armazenou perfeitamente cada um dos padrões \mathbf{x}^ξ para $\xi = 1, \dots, 4$. Contudo, a fim de comparar o desempenho da KS-FAM em termos de NRMSE com outros modelos de memórias associativas, elaboramos um série de experimentos que envolve investigações sobre incompletude, variação de claridade e orientação, assim como presença de vários tipos de ruídos. Em particular, em nossos experimentos consideramos também a rede de Hamming [35, 53], a MAM $\mathbf{W}_{\mathbf{X}\mathbf{X}}$ [73], a FAM max-min de Kosko [51], a FAM de Junbo [39], a IFAM de Lukasiewicz [93], a “kernel associative memory” (KAM) [103], o modelo “brain-state-in-a-box” (BSB) generalizado de Costantini et al. [17], a MAM $\mathbf{W}_{\mathbf{X}\mathbf{X}} + \mathbf{v}$ [92], a memória associativa linear óptima (OLAM) [49], e a rede Hopfield complexa de Tanaka et al. [98].

Lembre que o modelo KAM dispõe de n neurônios escondidos, onde n é o número de protótipos (ou memórias fundamentais) a ser armazenado. Cada neurônio correspondente a um protótipo \mathbf{x}^ξ , para $\xi = 1, \dots, n$, cuja ativação, ao apresentar uma entrada \mathbf{x} , é dada pelo cálculo de uma função “Kernel” $k(\mathbf{x}, \mathbf{x}^\xi)$. Neste trabalho adotamos a popular “Gaussian kernel function” dada pela seguinte expressão:

$$k(\mathbf{x}, \mathbf{x}^\xi) = \exp\left(\frac{\|\mathbf{x} - \mathbf{x}^\xi\|^2}{2\sigma^2}\right) \quad \text{para } \xi = 1, \dots, n. \quad (3.28)$$

Além disso, seguimos o sugerido por Stokbro et al. e escolhemos σ como sendo a distância euclidiana média entre todos os pares de diferentes centros das funções

de base radiais $K(\cdot, \mathbf{x}^\xi)$ [82, 103]:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{\xi \neq \gamma} \|\mathbf{x}^\xi - \mathbf{x}^\gamma\|^2} \quad (3.29)$$

Com respeito ao modelo BSB generalizado, a abordagem de Constantino et al. decompõe uma imagem com L tons de cinza em $B = \log_2 L$ imagens binárias [17], onde cada imagem é armazenada numa rede BSB binária, descritas pelas seguintes equações discretizadas

$$\mathbf{x}(k+1) = g(\mathbf{x}(k) + W\mathbf{x}(k)), \quad k = 1, 2, \dots \quad (3.30)$$

onde $g(\mathbf{x}) = (-\mathbf{1}) \vee (\mathbf{x} \wedge \mathbf{1})$. Neste trabalho, computamos a matriz de pesos sinápticos W seguindo o método iterativo nas equações 5 e 6 de [17] usando $\delta = 100$ e $\eta = 0.1$.

A rede de Hopfield complexa, consiste de uma rede recorrente que armazena memórias fundamentais com entradas complexas. Em particular, cada valor de pixel numa imagem com L tons de cinza é identificado como um ponto no conjunto $J_L = \{\mathbf{e}^{i(L-j)\theta_L}; j = 0, \dots, L-1\}$, onde $\theta_L = \frac{2\pi}{L}$. Na abordagem de Tanaka et al., os pesos da rede são sintetizados utilizando a regra de projeção generalizada [52]. Nas simulações, atualizamos os estados dos neurônios da rede até os mesmos oscilarem na ordem de 10^{-4} ou até atingir um máximo de 10 iterações, onde o esquema de atualização dos estados dos neurônios adotado é dado pela equação 12 em [98] com $\epsilon = 0.2$,

3.4.2.1 Experimentos usando padrões incompletos

Neste primeiro experimento, investigaremos os modelos de memórias associativas apresentando apenas partes das imagens originais como entrada, tais como, ilustradas na Figura 3.7. Uma interpretação visual das saídas deste teste para a imagem incompleta do cameraman é visualizada na Figura 3.8. A Tabela 3.1 lista os NRMSEs resultantes para cada imagem parcial.

Uma breve olhada na Tabela 3.1 revela que o modelo KS-FAM obteve o melhor desempenho entre os modelos testados em termos de NRMSE, completando perfeitamente todas as imagens parciais. Um resultado similar foi atingido pela rede de Hamming, que apenas falhou em recuperar a imagem do cameraman. Os



Figura 3.7: Versões incompletas das imagens originais.



Figura 3.8: Da esquerda para direita e de cima para baixo, a primeira imagem exibe uma versão incompleta da imagem do cameraman e as imagens restantes correspondem, respectivamente, as saídas produzidas pela KS-FAM, rede de Hamming, MAM W_{xx} , FAM de Kosko, FAM de Junbo, IFAM de Lukasiewicz, KAM, BSB generalizada, MAM $W_{xx} + \nu$, OLAM, e rede Hopfield complexa.

NRMSEs produzidos pelas MAM W_{xx} , KAM, e IFAM de Lukasiewicz são comparáveis e relativamente pequenos. Os piores desempenhos ficaram por conta da rede de Hopfield complexa, da MAM $W_{xx} + \nu$ e da FAM de Kosko.

3.4.2.2 Variações de claridade e orientação

Neste experimento modificamos a claridade e a orientação das imagens originais. Especificamente, subtraímos uma constante (0.35 no caso fuzzy e 89 no caso de tons de cinza) na imagem da árvore e adicionamos a mesma constante na imagem

Tabela 3.1: NRMSEs produzido pelos modelos de AMs ao apresentar os padrões incompletos da Figura 3.7

	Árvore	Lena	Igreja	Cameraman
KS-FAM	0	0	0	0
Hamming	0	0	0	0.7599
W_{xx}	0.0088	0.0191	0	0.0412
FAM de kosko	0.4764	0.8695	1.1443	0.6609
FAM de Junbo	0.1658	0.2067	0.0666	0.4184
IFAM de Luk.	0.0088	0.0193	0	0.0448
KAM	0.0905	0.0772	0.0059	0.5359
BSB gen.	0	0.1401	0.2600	0.7452
$W_{xx} + v$	0.6482	0.8042	0.6979	0.7809
OLAM	0.2451	0.4706	0.4538	0.7268
Hopfield	0.0067	0.3324	0.1172	0.8559

da Lena. Os valores dos pixels resultantes foram limitados pelos valores das fronteiras do domínio fuzzy $[0, 1]$ ou domínio de tons de cinza $[0, 255]$, dependendo do caso. Além disso, rotacionamos a imagem da igreja por 10 graus à esquerda e a imagem do cameraman por 10 graus à direita.

As Figuras 3.9, 3.10, 3.11, e 3.12 exibem os resultados deste experimentos para os diversos modelos testados. A KS-FAM conseguiu recuperar perfeitamente todas as quatro imagens originais a partir de suas versões modificadas. Desconsiderando a KS-FAM, o melhor desempenho entre as memórias restantes foi alcançado pelo modelo KAM, já que a rede de Hamming foi incapaz de lidar com variações de claridade. A Tabela 3.2 sumariza os resultados deste experimento em termos de NRMSE.

Analisando as saídas produzidas pelos modelos nas figuras de 3.9 à 3.12 e os erros apresentados na Tabela 3.2, verificamos que os piores desempenhos ficaram por conta das memórias W_{xx} , FAM de Kosko, FAM de Junbo e a IFAM de Lukasiewicz. Além disso, a BSB generalizada de Constantini et al. e a rede de Hopfield complexa de Tanaka et al. obtiveram um comportamento similar a rede de Hamming, produzindo saídas satisfatórias, tanto em termos de NRMSE quanto visual, apenas para as imagens com variações de orientação.

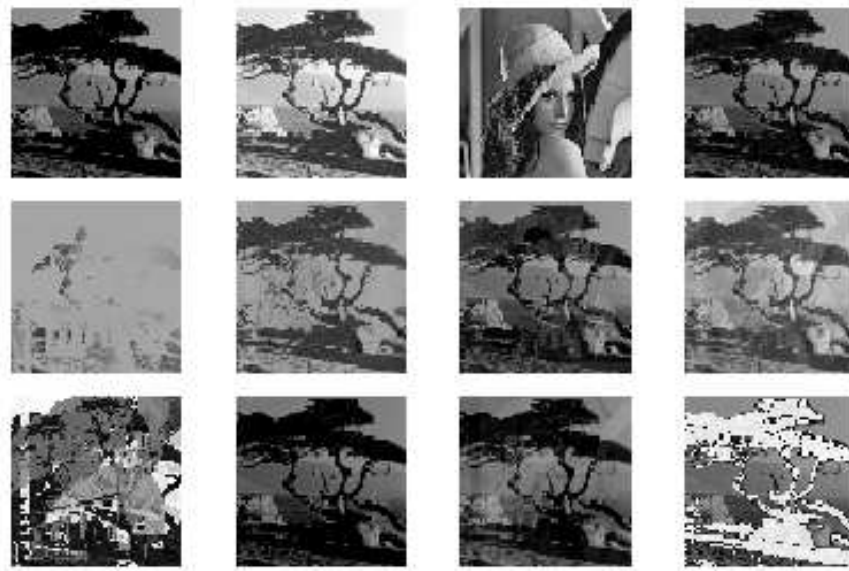


Figura 3.9: A imagem no canto superior esquerdo exibe a versão escurecida da imagem da árvore. As imagens restantes, lidas da direita para esquerda e de cima para baixo, representam as saídas produzidas pela KS-FAM, rede de Hamming, MAM W_{xx} , FAM de Kosko, FAM de Junbo, IFAM de Lukasiewicz, KAM, BSB generalizada, MAM $W_{xx} + \nu$, OLAM, e rede Hopfield complexa.

Tabela 3.2: NRMSEs produzido pelos modelos de AMs com padrões contendo variações de claridade e orientação

	Árvore	Lena	Igreja	Cameraman
KS-FAM	0	0	0	0
Hamming	0.6347	0.8414	0	0
W_{xx}	0.4771	0.7354	1.6015	0.9509
FAM de kosko	0.4302	0.8937	1.1586	0.7300
FAM de Junbo	0.2603	0.7240	1.3803	0.7014
IFAM de Luk.	0.4517	0.7240	1.5378	0.9063
KAM	0.1945	0.1499	0.0566	0.0784
BSB gen.	0.5595	0.9764	0.0272	0.0052
$W_{xx} + \nu$	0.6032	0.4615	0.6168	0.4765
OLAM	0.4986	0.6810	0.2892	0.1937
Hopfield	0.8564	0.8901	0.0338	0.0150

3.4.2.3 Padrões ruidosos

Finalmente, corrompemos as imagens originais introduzindo vários tipos de ruídos: ruído pimenta de densidade 0.25, ruído sal e pimenta de densidade 0.5, e



Figura 3.10: Da esquerda para direita e de cima para baixo, a primeira imagem mostra a versão clareada da imagem da Lena, enquanto que as imagens restantes correspondem às saídas dos modelos testados listados na mesma ordem que na Figura 3.9.

ruído gaussiano com média zero e variância 0.03. Cada uma das Figuras 3.13, 3.14 e 3.15 ilustra as saídas produzidas pelas memórias associativas avaliadas ao apresentar o padrão corrompido visualizado no canto esquerdo superior da própria imagem. Note que na Figura 3.14, as saídas produzidas pelas memórias W_{XX} e a IFAM de Lukasiewicz formam uma imagem quase que completamente branca.

A Tabela 3.3 mostra os NRMSEs obtidos em 100 experimentos para cada padrão \mathbf{x}^ξ , $\xi = 1, \dots, 4$. Tanto a KS-FAM quanto a rede de Hamming produziram exatamente a resposta desejada para cada versão corrompida apresentada e, por isso, os NRMSEs gerados por estes modelos durante tais experimentos foram iguais a zero. Outros modelos que também apresentaram uma tolerância muito satisfatória com respeito aos três tipos de ruídos analisados, foram: KAM, BSB generalizada e a rede Hopfield complexa. A MAM W_{XX} , a FAM de Junbo e a IFAM de Lukasiewicz forão capazes apenas de lidar de maneira satisfatória com o ruído pimenta.



Figura 3.11: Da esquerda para direita e de cima para baixo, a primeira visualiza a versão rotacionada da imagem da igreja. As imagens restantes correspondem, respectivamente, as saídas dos modelos testados na mesma ordem da Figura 3.9.



Figura 3.12: Da esquerda para direita e de cima para baixo, a primeira imagem mostra a versão rotacionada da imagem do cameraman. As demais imagens correspondem às saídas dos modelos testados, dispostas como na Figura 3.9.

Tabela 3.3: NRMSEs produzidos pelos modelos de AMs com padrões ruidosos

	Ruído Pimenta (Densidade 0.25)	Ruído Sal e Pimenta (Densidade 0.5)	Ruído Gaussiano ($\sigma^2 = 0.03$)
KS-FAM	0	0	0
Hamming	0	0	0
W_{xx}	0.0024	1.4388	0.9005
FAM de kosko	0.8068	0.8200	0.8185
FAM de Junbo	0.0399	1.2381	0.6563
IFAM de Luk.	0.0027	1.2764	0.8701
KAM	0.0361	0.1018	0.0137
BSB gen.	0	0.1374	0.1181
$W_{xx} + v$	0.7433	0.5733	0.2770
OLAM	0.2507	0.2437	0.0365
Hopfield	0.0297	0.14075	0.0116



Figura 3.13: Da esquerda para direita e de cima para baixo, a primeira imagem representa uma versão da imagem da Lena contendo ruído pimenta de densidade 0.25. As imagens restantes correspondem, respectivamente, as saídas da KS-FAM, rede Hamming, W_{xx} , FAM de Kosko, FAM de Junbo, IFAM de Lukasiewicz, KAM, BSB generalizada, $W_{xx} + v$, OLAM e rede de Hopfield complexa.

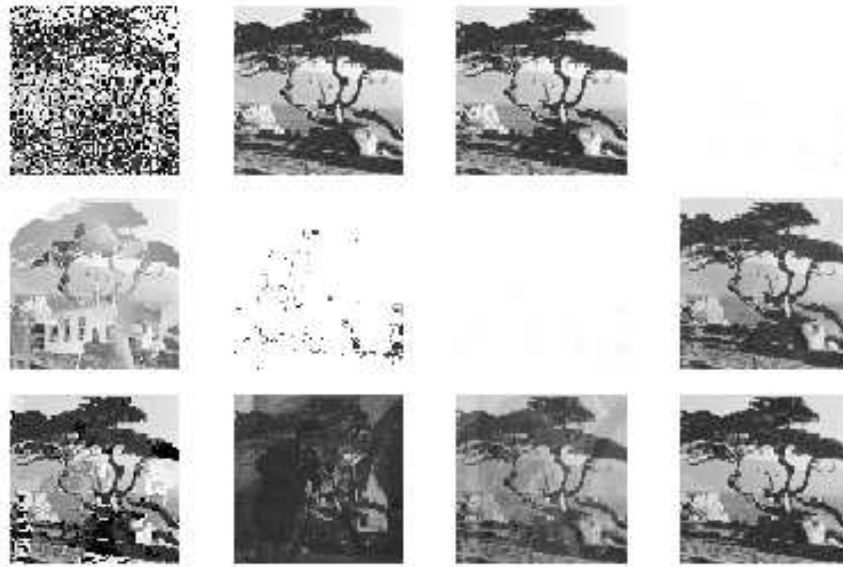


Figura 3.14: A imagem no canto superior esquerdo exibe uma versão corrompida da imagem da árvore contendo ruído sal e pimenta de densidade 0.5. As imagens restantes correspondem as saídas produzidas pelos modelos testados, dispostas na mesma ordem que na Figura 3.13.



Figura 3.15: A imagem no canto superior esquerdo mostra uma versão corrompida da imagem da igreja contendo ruído gaussiano com média zero e variância 0.03. As imagens restantes correspondem às saídas produzidas pelos modelos testados, dispostas na mesma ordem que na Figura 3.13.

Redes neurais morfológicas construtivas

As redes neurais morfológicas que ajustam automaticamente sua arquitetura durante a fase de treinamento são denominadas construtivas. De maneira geral, a estratégia de treinamento das redes morfológicas construtivas consiste da utilização de um certo algoritmo de clusterização e da conversão de cada cluster gerado numa ou mais unidades de processamento. Exemplos de MNN construtivas incluem a classe dos perceptron morfológicos (MPs) [72, 83, 91] e os modelos de neurocomputação em reticulados fuzzy (FLNs) de Kaburlasos et al. [43, 44, 41, 40, 42, 63].

4.1 Neurocomputação em reticulados fuzzy

Kaburlasos et al. define os modelos FLNs sobre uma estrutura matemática denominada *reticulado fuzzy*, que nada mais é do que uma extensão do conceito de reticulado a partir da extensão do conceito de ordem parcial. Por exemplo, se x e y pertencem a um reticulado \mathbb{L} tal que $x < y$, então, ao invés de falar que x é menor que y , num reticulado fuzzy, dizemos que x é menor que y com um grau $\sigma(x, y) \in [0, 1]$, onde σ representa uma extensão da ordem parcial do reticulado \mathbb{L} [43, 63].

Um operador binário $\sigma : \mathbb{L} \times \mathbb{L} \rightarrow [0, 1]$ é dito uma *ordem parcial fuzzy* no reticulado \mathbb{L} se os seguintes axiomas são cumpridos para todo x, y e z pertencentes

à \mathbb{L} :

$$\text{A-1 } \sigma(x, 0_{\mathbb{L}}) = 0 \text{ se } x \neq 0_{\mathbb{L}}$$

$$\text{A-2 } \sigma(x, x) = 1$$

$$\text{A-3 se } x \leq y \text{ então } \sigma(z, x) \leq \sigma(z, y).$$

Apesar de uma ordem parcial fuzzy σ representar uma medida de inclusão com valores no domínio fuzzy $[0, 1]$, optamos por falar de uma ordem parcial fuzzy ao invés de medida de inclusão (fuzzy) pelo fato de σ generalizar a ordem parcial convencional do reticulado. Um par (\mathbb{L}, σ) composto de um reticulado \mathbb{L} e uma ordem parcial fuzzy σ é chamado de um *reticulado fuzzy*.

Ordens parciais fuzzy podem ser construídas a partir de funções de *avaliações positivas*. Seja \mathbb{L} um reticulado qualquer. Uma função de avaliação positiva v é uma função de \mathbb{L} em \mathbb{R} tal que $v(x \vee y) = v(x) + v(y) - v(x \wedge y)$ e $x < y \Rightarrow v(x) < v(y)$ para todo $x, y \in \mathbb{L}$. A seguinte proposição mostra como elaborar ordens parciais fuzzy, utilizando tais funções.

Proposição 3. [40] *Sejam \mathbb{L} um reticulado e $v : \mathbb{L} \rightarrow \mathbb{R}$ uma função de avaliação positiva com $v(0_{\mathbb{L}}) = 0$. As funções σ_k e σ_s dadas por*

$$\sigma_k(x, y) = \begin{cases} 1 & , x \vee y = 0_{\mathbb{L}} \\ \frac{v(y)}{v(x \vee y)} & , c.c. \end{cases} \quad e \quad (4.1)$$

$$\sigma_s(x, y) = \begin{cases} 1 & , x = 0_{\mathbb{L}} \\ \frac{v(x \wedge y)}{v(x)} & , c.c. \end{cases} \quad \forall x, y \in \mathbb{L} \quad (4.2)$$

representam ordens parciais fuzzy.

Exemplos de escolhas convencionais de pares de uma função de avaliação positiva e um automorfismo dual para construções de ordens parciais fuzzy são

$$v_l(x) = x \quad e \quad \theta_l(x) = 1 - x \quad \text{para } x \in [0, 1] \quad (4.3)$$

$$v_e^\zeta(x) = \frac{1}{1 + e^{-\lambda(x-x_m)}} \quad e \quad \theta_e(x) = 2x_m - x, \quad \text{para } x \in [x_{\min}, x_{\max}] \subseteq \mathbb{R} \quad (4.4)$$

onde $x_m = \frac{x_{\max} + x_{\min}}{2}$ e $\lambda = \frac{\zeta}{x_{\max} - x_{\min}}$ para algum $\zeta > 0$.

Considere o produto de reticulados $\mathbb{L} = \mathbb{L}_1 \times \dots \times \mathbb{L}_n$. Se $v_i : \mathbb{L}_i \rightarrow \mathbb{R}$ denota uma função de avaliação positiva em \mathbb{L}_i para $i = 1, \dots, n$, então, $v : \mathbb{L} \rightarrow \mathbb{R}$ dada por $v(\mathbf{x}) = v_1(x_1) + \dots + v_n(x_n)$ para todo $\mathbf{x} \in \mathbb{L}$ é uma avaliação positiva.

A estratégia de computação da maioria dos modelos FLNs na literatura é baseado no cálculo de ordens parciais fuzzy (ou grau de inclusão) entre intervalos generalizados de um certo produto de reticulado. Recorde da Seção 2.1 que o conjunto dos intervalos generalizados de um reticulado também representam um reticulado com a ordem parcial induzida pela ordem do reticulado. Em particular, sejam $(\mathbf{a}, \mathbf{b}), (\mathbf{c}, \mathbf{d})$ intervalos generalizados no produto de reticulados $\mathbb{L} = \mathbb{L}_1 \times \dots \times \mathbb{L}_n$, definimos $(\mathbf{a}, \mathbf{b}) \leq (\mathbf{c}, \mathbf{d})$ se, e somente se, $\mathbf{c} \leq \mathbf{a}$ e $\mathbf{b} \leq \mathbf{d}$. Equivalentemente, se θ é uma negação (ou um automorfismo dual) em \mathbb{L} , então, $(\mathbf{a}, \mathbf{b}) \leq (\mathbf{c}, \mathbf{d})$ se, e somente se, $\theta(\mathbf{a}) \leq \theta(\mathbf{c})$ e $\mathbf{b} \leq \mathbf{d}$. Esta última observação permite produzir ordens parciais fuzzy para o conjunto de intervalos generalizados de \mathbb{L} a partir de funções de avaliação positiva no sentido da Proposição 3.

Se v e θ são, respectivamente, uma avaliação positiva e um automorfismo dual em um reticulado \mathbb{L} , então o operador $v : \tau(\mathbb{L}) \rightarrow \mathbb{R}$ dado por

$$v(\mathbf{a}, \mathbf{b}) = v(\theta(\mathbf{a})) + v(\mathbf{b}) \text{ para todo } (\mathbf{a}, \mathbf{b}) \in \tau(\mathbb{L})$$

representa uma avaliação positiva no reticulado $\tau(\mathbb{L})$ [40].

Apesar dos modelos FLNs serem definidos sobre o conjunto dos intervalos generalizados de um certo reticulado, todo processamento dos modelos específicos de FLNs que surgiram na literatura, até o momento, estão restritos a cálculos de ordens parciais fuzzy entre intervalos generalizados de reticulados da forma (\mathbf{a}, \mathbf{b}) tal que $\mathbf{a} \leq \mathbf{b}$. Note que cada intervalo deste tipo pode ser identificado como uma única hipercaixa $[\mathbf{a}, \mathbf{b}] \neq \emptyset$ contida em \mathbb{L} , conforme mencionado na Seção 2.1. Por isso, os cálculos executados pelas FLNs também representam medidas de inclusão entre hipercaixas.

Neste trabalho, estamos interessados em dois modelos de FLNs que podem ser aplicados na solução de problemas de classificação: a *FLNN* (“fuzzy lattice neural network”) [63] e o modelo *FLR* (“fuzzy lattice reasoning”)[40]. Assim como outros modelos de FLNs, seus desenvolvimentos foram inspirados pela teoria de ressonância adaptativa [11]. Porém, veremos, até o final desta seção, que estas redes representam MNNs no sentido intuitivo.

Durante a fase de treinamento dos modelos FLNN e FLR, clusters em termos de hipercaixas são produzidos num produto de reticulados $\mathbb{L}_1 \times \dots \mathbb{L}_n$, onde cada \mathbb{L}_i é um reticulado totalmente ordenado. Ao final, regras de classificação baseadas nestas hipercaixas são produzidas, atribuindo rótulos de classes apropriados. Especificamente, um padrão de entrada \mathbf{x} , que pode ser identificado como uma hipercaixa $[\mathbf{x}, \mathbf{x}]$ ou, equivalentemente, como um intervalo generalizado (\mathbf{x}, \mathbf{x}) , é classificado como sendo da classe associada à hipercaixa (ou intervalo generalizado) B , cujo grau de inclusão de \mathbf{x} em B é maior. Lembre que a noção de medida de inclusão nos modelos FLNs coincide com a de ordem parcial fuzzy. Geralmente, quando utilizamos uma FLNN ou uma FLR, definimos uma medida de inclusão σ_k dada pela Equação 4.1 a partir de uma certa escolha de uma função de avaliação positiva v e um automorfismo dual θ .

Note que, cada hipercaixa B produzida na fase de treinamento corresponde a uma unidade de processamento que computa o grau de inclusão de um padrão de entrada \mathbf{x} em B a partir de uma certa ordem parcial fuzzy. No contexto da MM, o neurônio resultante executa uma erosão no sentido intuitivo da entrada \mathbf{x} por um elemento estruturante B . Portanto, os classificadores FLNNs e FLRs representam MNNs no sentido intuitivo.

Ocasionalmente, os algoritmos de treinamento e suas modificações tanto da FLNN quanto da FLR podem produzir sobreposições de hipercaixas pertencentes a classes distintas, embora, de acordo com Kaburlasos et al., esses eventos ocorram raramente [40].

Por fim, gostaríamos de ressaltar que a diferença essencial entre a FLR e a FLNN está no algoritmo de treinamento. Especificamente, eles se diferenciam na estratégia adotada para confeccionar agrupamentos no conjunto de treinamento em termos de hipercaixas. O algoritmo de treinamento da FLNN possui complexidade $\mathbf{O}(n^3)$, onde n é a quantidade de padrões de treinamento, e sua versão original não requer escolhas de parâmetros adicionais [63]. Em contraste, o algoritmo de treinamento da FLR, descrito em [40], inclui testes de assimilações de hipercaixas em termos de um parâmetro de vigilância $\rho_{\text{crit}} \in [0.5, 1]$, e exibe complexidade $\mathbf{O}(n^3)$ apenas no pior caso.

4.2 Perceptron morfológico

O perceptron morfológico (MP) foi uma das primeiras redes neurais morfológicas a aparecer na literatura. Inicialmente, foi desenvolvida como um modelo simples de uma camada oculta para solucionar problemas de classificação binários [69, 72]. A idéia básica para o desenvolvimento dos MPs foi substituir o produto matricial da álgebra linear usado no perceptron convencional, por um produto matricial da álgebra minimax. Uma vez que as funções descritas pelo MP não são diferenciáveis, o algoritmo backpropagation tradicional não pode ser aplicado para treinar esta rede. Ao invés de recorrer ao algoritmo modificado do backpropagation sugerido por Pessoa e Maragos [62], que é capaz de lidar com esses tipos de funções, um algoritmo completamente novo foi proposto no começo [83]. Este algoritmo, ao contrário dos mencionados anteriormente, não apenas determina os pesos dos neurônios da rede, mas também adiciona novos neurônios na camada oculta, resultando uma arquitetura feedforward que reflete a decomposição sugerida por Banon e Barrera [5]. Portanto, o perceptron morfológico pode ser visto como uma rede neural construtiva [88].

A função de agregação empregada nos neurônios individuais de MPs são da forma ε_A , δ_A , $\bar{\varepsilon}_A$ e $\bar{\delta}_A$, onde as componentes \mathbf{a}_{ij} da matriz \mathbf{A} , denotam os pesos sinápticos e possuem valores no l-grupo \mathbb{R} . Lembre que pelo Teorema 2, estes operadores representam operações elementares da morfologia matemática. Especificamente, dado um vetor de entrada $\mathbf{x} \in \mathbb{R}_{\pm\infty}^n = (\mathbb{R}_{\pm\infty})^n$ (na prática, estamos restritos a vetores de entrada $\mathbf{x} \in \mathbb{R}^n$), um vetor de pesos sinápticos $\mathbf{w} \in \mathbb{R}_{\pm\infty}^n$ e uma função de ativação f , um neurônio oculto do perceptron morfológico calcula uma saída \mathbf{y} de acordo com uma das seguintes regras:

1. $\mathbf{y} = f(\varepsilon_{\mathbf{w}}(\mathbf{x}))$, onde

$$\varepsilon_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^t \boxtimes \mathbf{x} = \bigwedge_{i=1}^n (x_i +' w_i); \quad (4.5)$$

2. $\mathbf{y} = f(\delta_{\mathbf{w}}(\mathbf{x}))$, onde

$$\delta_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^t \boxtimes \mathbf{x} = \bigvee_{i=1}^n (x_i + w_i); \quad (4.6)$$

3. $\mathbf{y} = f(\bar{\varepsilon}_{\mathbf{w}}(\mathbf{x}))$, onde

$$\bar{\varepsilon}_{\mathbf{w}}(\mathbf{x}) = \delta_{\mathbf{w}}(\mathbf{x}) \circ \mathbf{v}_*(\mathbf{x}) = \bigvee_{i=1}^n (\mathbf{x}_i^* + \mathbf{w}_i); \quad (4.7)$$

4. $\mathbf{y} = f(\bar{\delta}_{\mathbf{w}}(\mathbf{x}))$, onde

$$\bar{\delta}_{\mathbf{w}}(\mathbf{x}) = \varepsilon_{\mathbf{w}}(\mathbf{x}) \circ \mathbf{v}_*(\mathbf{x}) = \bigwedge_{i=1}^n (\mathbf{x}_i^* + ' \mathbf{w}_i). \quad (4.8)$$

Os pesos do MP são determinados usando um algoritmo de treinamento supervisionado [83] que constrói hipercaixas n -dimensionais em torno de conjuntos de dados que compartilham o mesmo rótulo de classe. Note que uma hipercaixa em $\mathbb{R}_{\pm\infty}^n$ é completamente determinada por um vértice inferior \mathbf{a} e um vértice superior \mathbf{b} , onde um elemento $\mathbf{x} \in \mathbb{R}_{\pm\infty}^n$ está contido em $[\mathbf{a}, \mathbf{b}]$ se e somente se $\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$. O Teorema 3 implica que, equivalentemente, temos $\varepsilon_{\mathbf{v}}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}}(\mathbf{x}) \geq 0$ onde $\mathbf{v} = (\mathbf{a}^*)^t$ e $\mathbf{w} = \mathbf{b}$. Entretanto, para verificar se \mathbf{x} está contido em uma hipercaixa $[\mathbf{a}, \mathbf{b}]$ é suficiente o cálculo de uma única erosão, pois, basta considerar $\mathbf{u}, \mathbf{y} \in \mathbb{R}_{\pm\infty}^{2n}$ dados por $\mathbf{u} = [\mathbf{v}^t, \mathbf{w}^t]^t$ e $\mathbf{y} = [\mathbf{x}^t, \mathbf{x}^*]^t$ para obtermos $\varepsilon_{\mathbf{v}}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}}(\mathbf{x}) = \varepsilon_{\mathbf{u}}(\mathbf{y})$.

Durante o treinamento, o MP gera automaticamente hipercaixas da forma $[\mathbf{a}^j, \mathbf{b}^j]$ que correspondem a pares consistentes de erosões $\varepsilon_{\mathbf{v}_j}$ e anti-dilatações $\bar{\delta}_{\mathbf{w}_j}$. Na fase de teste, um padrão de entrada \mathbf{x} é associado com a primeira classe se, e somente se, \mathbf{x} estiver contido em uma das hipercaixas $[\mathbf{a}^j, \mathbf{b}^j]$. Em outras palavras, a saída do MP é determinada pela seguinte equação para algum $m \in \mathbb{N}$:

$$\mathbf{y} = f\left(\bigvee_{j=1}^m (\varepsilon_{\mathbf{v}_j}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}_j}(\mathbf{x}))\right), \quad (4.9)$$

$$\text{onde } f(\mathbf{x}) = \begin{cases} 1, & \text{se } \mathbf{x} \geq 0, \\ 0, & \text{se } \mathbf{x} < 0. \end{cases} \quad (4.10)$$

Portanto, um padrão de entrada \mathbf{x} é classificado com pertencente à classe 1 se, e somente se, $\mathbf{y} = 1$.

A Figura 4.2 retrata a arquitetura final construída pelo algoritmo de treinamento. Note que o neurônio de saída executa uma dilatação $\mathbb{R}_{\pm\infty}^m \rightarrow \mathbb{R}_{\pm\infty}$, dada

por $\mathbf{x} \rightarrow \bigvee_{i=1}^m x_i$, seguida pela aplicação da função de ativação f .

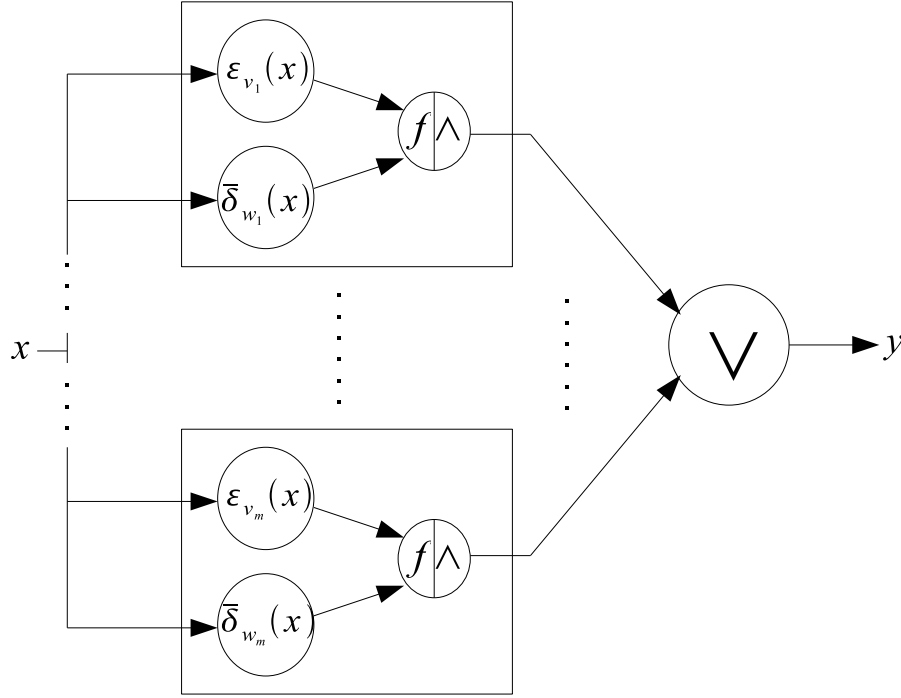


Figura 4.1: Arquitetura do perceptron morfológico.

O objetivo da aprendizagem do MP é estimar uma função desconhecida ψ que mapeia padrões de entrada em um reticulado completo $\mathbb{R}_{\pm\infty}^n$ para rótulos de classe no reticulado completo $\{0, 1\}$ produzindo perfeita classificação. Note que a função limiar f representa um homomorfismo $\mathbb{R}_{\pm\infty}^n \rightarrow \{0, 1\}$. Portanto, a Equação 4.12 pode ser reescrita na forma

$$y = \bigvee_{j=1}^m (f(\varepsilon_{v_j}(\mathbf{x})) \wedge f(\bar{\delta}_{w_j}(\mathbf{x}))). \quad (4.11)$$

Uma vez que a função f representa uma erosão $\mathbb{R}_{\pm\infty}^n \rightarrow \{0, 1\}$ (no sentido algébrico), podemos utilizar o Lema 2 para inferir que $f \circ \varepsilon_{v_j}$ é uma erosão $\mathbb{R}_{\pm\infty}^n \rightarrow \{0, 1\}$ e que $f \circ \bar{\delta}_{w_j}$ é uma anti-dilatação $\mathbb{R}_{\pm\infty}^n \rightarrow \{0, 1\}$ para todo $j = 1, \dots, m$. Dessa maneira, o MP estima a função de classificação perfeita $\psi : \mathbb{R}_{\pm\infty}^n \rightarrow \{0, 1\}$ em termos de um máximo de pares de mínimos de erosões e anti-dilatações. Recorde que $\psi : \mathbb{R}_{\pm\infty}^n \rightarrow \{0, 1\}$ pode ser decomposta exatamente pela Equação 2.10 usando um supremo de pares de mínimos de erosões e anti-dilatações [5].

Dado um conjunto de padrões de treinamento $\mathbf{x}^\xi \in \mathbb{R}^n$ divididos em duas

classes distintas: C_1 e C_0 . O algoritmo construtivo de treinamento do MP constrói m hipercaixas, tais que, a união entre elas contém todos os padrões de treinamento de C_1 , enquanto, todos os padrões de treinamentos restantes pertencentes à classe C_0 residem fora desta. Em outras palavras, se \mathbf{y}^ξ denota $\bigvee_{j=1}^m (f(\varepsilon_{v_j}(\mathbf{x}^\xi)) \wedge f(\bar{\delta}_{w_j}(\mathbf{x}^\xi)))$ então o algoritmo de treinamento produz $\mathbf{x}^\xi = 1$ se, e somente se, $\mathbf{y}^\xi \in C_1$.

Recentemente, foi publicada uma versão modificada do algoritmo de treinamento original do MP que possui garantia de convergência em um número finitos de passos, produzindo uma superfície de decisão que separa perfeitamente os dados de treinamento da classe C_1 dos dados de treinamento da classe C_0 [88]. Esta versão modificada também produziu melhores resultados de classificação do que a versão original em várias simulações. Entretanto, assim como a versão original, a versão melhorada apenas pode ser aplicada a problemas de classificação binário [74, 88].

4.3 Perceptron morfológico com aprendizagem competitiva (MP/CL)

Visto que a abordagem inicial do perceptron morfológico pode ser aplicada apenas a problemas de classificação binários, submetemos a novas modificações a fim de lidar com S classes, onde $S > 1$. Para este fim, podemos aplicar o algoritmo de treinamento original melhorado [88] por S vezes para diferenciar os dados de treinamento de cada rótulo de classe dos demais dados de treinamento. Este processo gera S módulos com arquitetura similar àquela ilustrada na Figura 4.2. Substituindo a função de ativação f pela função identidade e os neurônios de saída limiares por neurônios de saída competitiva, os S módulos podem ser combinados utilizando uma operação do tipo $\arg \max$ tal como a arquitetura da Figura 4.3. O modelo resultante é chamado *perceptron morfológico com neurônios competitivos* (MP/C).

Precisamente, seja $C'_s \subseteq \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$ o conjunto de padrões de treinamento que pertencem à s -ésima classe, onde $s = 1, \dots, S$. Para cada $s = 1, \dots, S$, simplesmente atribuímos $C_1 = C'_s$ e $C_0 = \bigcap_{t \neq s} C'_t$ e aplicamos o algoritmo de treinamento do MP, produzindo assim um conjunto de hipercaixas que diferencia os padrões de treinamento da s -ésima classe das demais. Este procedimento gera

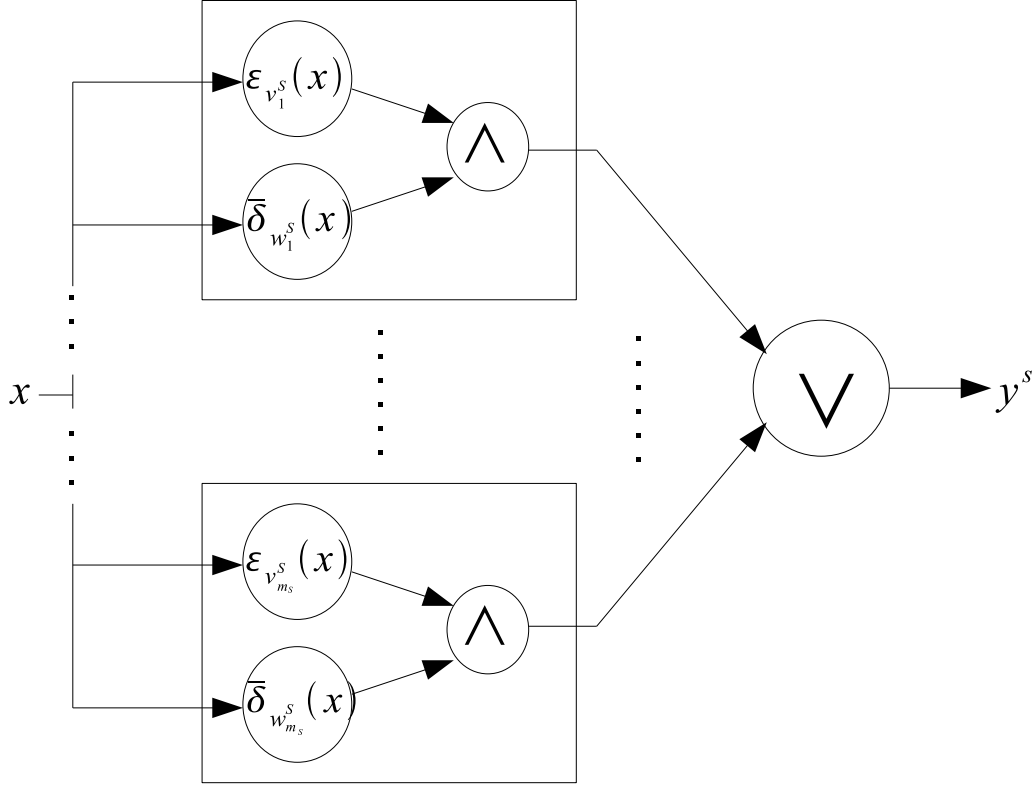


Figura 4.2: Módulo de um MP/C ou MP/CL correspondente à s -ésima classe.

módulos da forma ilustrada na Figura 4.2 para todo $s = 1, \dots, S$. Os pesos do s^o módulo são denotados por \mathbf{v}_j^s e \mathbf{w}_j^s , onde $j = 1, \dots, m_s$. Dado um padrão de entrada $\mathbf{x} \in \mathbb{R}^n$, o s^o módulo computa a seguinte saída y_s :

$$y_s = \bigvee_{j=1}^{m_s} (\epsilon_{v_j^s}(\mathbf{x}) \wedge \bar{\delta}_{w_j^s}(\mathbf{x})) \quad (4.12)$$

Subsequentemente, uma competição entre os S neurônios de saídas ocorre para determinar a classe de pertinência de um padrão de entrada \mathbf{x} produzida pelo MP/C. Em outras palavras, o MP/C atribui o rótulo de classe $\mathbf{y} = \arg \max_{s=1, \dots, S} y_s$ para um padrão de entrada \mathbf{x} tal como visualizado na Figura 4.3. Recorde que a camada de saída “winner-take-all” ($\arg \max$) pode ser implementada em software pela simples seleção do neurônio de saída que possui maior valor de ativação ou em termos de uma rede neural conhecida como MAXNET [53].

Apesar do MP/C atingir resultados satisfatórios e exigir baixo tempo de CPU (ou processamento) para treinamento em algumas simulações realizadas [88, 91], o algoritmo de treinamento do MP/C possui as seguintes desvantagens:

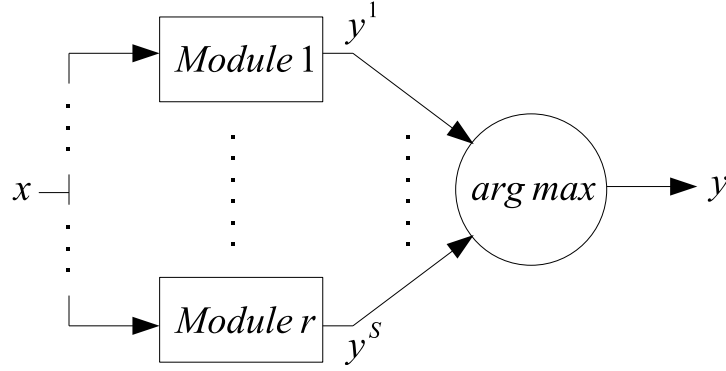


Figura 4.3: Arquitetura tanto do MP/C quanto do MP/CL.

- Pode produzir áreas de indecisão correspondentes a hipercaixas com rótulos de classes distintos;
- O MP/C resultante depende da ordem com que os padrões de treinamento são apresentados à rede durante o treinamento.

Para superar essas desvantagens, sugerimos um algoritmo de treinamento completamente diferente [89, 91]. Este novo algoritmo também se aplica a problemas de múltiplas classes e constrói o mesmo tipo de arquitetura que o algoritmo de treinamento do MP/C, porém, não possui as desvantagens mencionadas anteriormente. O modelo treinado usando este novo algoritmo é denominado *perceptron morfológico com aprendizagem competitiva* (MP/CL).

Inicialmente, descreveremos o algoritmo de treinamento para classificação binária em \mathbb{L}^n , onde \mathbb{L} é uma corrente. Em seguida, o estenderemos para lidar com múltiplas classes. Para começar, precisamos introduzir algumas notações pertinentes. Definimos os seguintes semi-espacos $H_i^-(\mathbf{c})$ e $H_i^+(\mathbf{c})$ assim como os seguintes hiperplanos $P_i(\mathbf{c})$ para todo $\mathbf{c} \in \mathbb{L}^n$ e todo $i = 1, \dots, n$.

$$\begin{aligned} H_i^-(\mathbf{c}) &= \{\mathbf{x} \in \mathbb{L}^n : x_i < c_i\}, \quad H_i^+(\mathbf{c}) = \{\mathbf{x} \in \mathbb{L}^n : x_i > c_i\}, \\ \text{e } P_i(\mathbf{c}) &= \{\mathbf{x} \in \mathbb{L}^n : x_i = c_i\}. \end{aligned} \quad (4.13)$$

O objetivo do algoritmo, é construir duas famílias disjuntas de hipercaixas $\mathcal{F}^1 = \{[\mathbf{a}_1^1, \mathbf{b}_1^1], \dots, [\mathbf{a}_{m_1}^1, \mathbf{b}_{m_1}^1]\}$ e $\mathcal{F}^2 = \{[\mathbf{a}_1^2, \mathbf{b}_1^2], \dots, [\mathbf{a}_{m_2}^2, \mathbf{b}_{m_2}^2]\}$ em \mathbb{L}^n , tais que \mathcal{F}^1 contenha todos os padrões de treinamento da primeira classe e \mathcal{F}^2 contenha todos os padrões de treinamento da segunda classe. Note que, se \mathbb{L} é um l-grupo

totalmente ordenado, então, a hipercaixa $[\mathbf{a}, \mathbf{b}]$ em \mathbb{L}^n determina uma erosão $\varepsilon_{\mathbf{v}} : \bar{\mathbb{L}}^n \rightarrow \bar{\mathbb{L}}$, onde $\mathbf{v} = -\mathbf{a}$, assim como, uma anti-dilatação $\bar{\delta}_{\mathbf{w}} : \bar{\mathbb{L}}^n \rightarrow \bar{\mathbb{L}}$ onde $\mathbf{w} = \mathbf{b}$. Adicionalmente, teríamos que um elemento arbitrário $\mathbf{x} \in \mathbb{L}^n$ está contido em $[\mathbf{a}, \mathbf{b}]$ se, e somente se, $\varepsilon_{\mathbf{v}}(\mathbf{x}) \wedge \bar{\delta}_{\mathbf{w}}(\mathbf{x}) \geq 0$, onde $0 \in \mathbb{L}$ representa o elemento neutro com respeito à operação de grupo $+$ sob \mathbb{L} . Desta maneira, cada família de hipercaixas \mathcal{F}^s gera \mathbf{m}_s pares consistentes de uma erosão $\varepsilon_{\mathbf{v}_j^s}$ e uma anti-dilatação $\bar{\delta}_{\mathbf{w}_j^s}$ para $j = 1, \dots, \mathbf{m}_s$ e $s = 1, 2$, que resultam em uma MNN com dois módulos similares ao ilustrado na Figura 4.2.

4.3.1 Algoritmo do MP/CL para problemas de classificação binários

Considere o conjunto treinamento $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \mathbb{L}^n$, onde \mathbb{L} é uma corrente. Sejam C^s o conjunto dos padrões de treinamento pertencentes à s -ésima classe, para $s = 1, 2$, e \mathcal{C} o símbolo utilizado para denotar a família de padrões de treinamento que ainda devem ser considerados pelo algoritmo. A seguir será apresentado o algoritmo para classificação de problemas binários.

Inicialize \mathcal{C} com $\mathcal{C} := \{X\}$, e as famílias de hipercaixas \mathcal{F}^s com $\mathcal{F}^s := \emptyset$ para $s = 1, 2$. Enquanto $\mathcal{C} \neq \emptyset$ execute os seguintes passos:

- (1): Selecione uma família de padrões de treinamento arbitrária $\bar{X} \in \mathcal{C}$ e defina $\bar{C}^s := \bar{X} \cap C^s$ para $s = 1, 2$. Compute, então, a menor hipercaixa $[\mathbf{a}^s, \mathbf{b}^s] = \text{box}(\bar{C}^s) = [\bigwedge \bar{C}^s, \bigvee \bar{C}^s]$ em $\bar{\mathbb{L}}^n$ contendo todos os elementos de \bar{C}^s onde $s = 1, 2$ (relembre que o reticulado limitado $\bar{\mathbb{L}}$ surge da junção de \mathbb{L} com as fronteiras universais $0_{\mathbb{L}}$ e $1_{\mathbb{L}}$).

Em seguida, determine a intersecção $I = [\mathbf{a}^1, \mathbf{b}^1] \cap [\mathbf{a}^2, \mathbf{b}^2]$. Note que $I = [\mathbf{a}, \mathbf{b}]$ onde $\mathbf{a} = \mathbf{a}^1 \vee \mathbf{a}^2$ and $\mathbf{b} = \mathbf{b}^1 \wedge \mathbf{b}^2$.

- (2): Se $I = \emptyset$, remova \bar{X} do conjunto \mathcal{C} e adicione as hipercaixas $[\mathbf{a}^1, \mathbf{b}^1]$ e $[\mathbf{a}^2, \mathbf{b}^2]$ nas famílias correntes de hipercaixas produzindo

$$\mathcal{F}^s := \mathcal{F}^s \cup \{[\mathbf{a}^s, \mathbf{b}^s]\}, \quad s = 1, 2. \quad (4.14)$$

- (3): Caso contrário, se $I \neq \emptyset$, um entre os casos (a) e (b) abaixo ocorre.

(a): Se $[\mathbf{a}^1, \mathbf{b}^1] \neq [\mathbf{a}^2, \mathbf{b}^2]$, aumentamos as famílias \mathcal{F}^s de hipercaixas pela execução dos seguintes comando para $i = 1, \dots, n$:

$$\mathcal{F}^s := \mathcal{F}^s \cup \{[\bigwedge_i (H_i^-(\mathbf{a}) \cap \bar{C}^s), \bigvee_i (H_i^-(\mathbf{a}) \cap \bar{C}^s)]\}, \quad s = 1, 2, \quad (4.15)$$

$$\mathcal{F}^s := \mathcal{F}^s \cup \{[\bigwedge_i (H_i^+(\mathbf{b}) \cap \bar{C}^s), \bigvee_i (H_i^+(\mathbf{b}) \cap \bar{C}^s)]\}, \quad s = 1, 2. \quad (4.16)$$

Este procedimento é visualizado na Figura 4.4. As Equações 4.15 e 4.16 equivalem a gerar a menor hipercaixa contendo todos os padrões de treinamento da s° classe que estão localizados, respectivamente, nos semi-espacos $H_i^-(\mathbf{a})$ e $H_i^+(\mathbf{b})$. Para $s = 1, 2$, as hipercaixas resultantes com o rótulo de classe s são, então, adicionadas ao conjunto de hipercaixas \mathcal{F}^s corrente.

Após atualizar as famílias de hipercaixas \mathcal{F}^1 e \mathcal{F}^2 , se $\bar{X} \cap I \neq \emptyset$, substitua \bar{X} por $\bar{X} \cap I$ em \mathcal{C} , caso contrário, remova \bar{X} de \mathcal{C} .

(b): Vamos analisar agora o caso em que $I = [\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2]$. A Figura 4.5 provê uma interpretação visual desta rara situação em \mathbb{R}^2 .

Sejam \mathbf{x} um vértice arbitrário da hipercaixa $I = [\mathbf{a}, \mathbf{b}]$ e \mathcal{P} o conjunto de hiperplanos da forma $P_i(\mathbf{x})$, $i \in \{1, \dots, n\}$, que contém o menor número de padrões de treinamento em \bar{X} . Sem perda de generalidade, escolhemos $\mathbf{x} = \mathbf{b}$. Deste modo, obtemos

$$\mathcal{P} = \{P_i(\mathbf{b}) : |P_i(\mathbf{b}) \cap \bar{X}| \leq |P_j(\mathbf{b}) \cap \bar{X}| \forall i, j \text{ onde } i, j = 1, \dots, n\} \quad (4.17)$$

Seja \mathcal{U} a união de todas as intersecções entre dois hiperplanos distintos de \mathcal{P} . A Figura 4.6 retrata \mathcal{P} e \mathcal{U} numa ocorrência de $[\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2]$ em \mathbb{R}^3 (note que mudamos a perspectiva para propósitos de visualização). Na Figura 4.6, temos $\mathcal{P} = \{P, P'\}$. Formalmente, temos

$$\mathcal{U} = \bigcup_{P, P' \in \mathcal{P}, P \neq P'} (P \cap P'). \quad (4.18)$$

Primeiro, aumentamos o conjunto \mathcal{F}^s adicionando todas hipercaixas da forma $[\mathbf{u}, \mathbf{u}] = \{\mathbf{u}\}$ para $\mathbf{u} \in \mathcal{U} \cap C^s$:

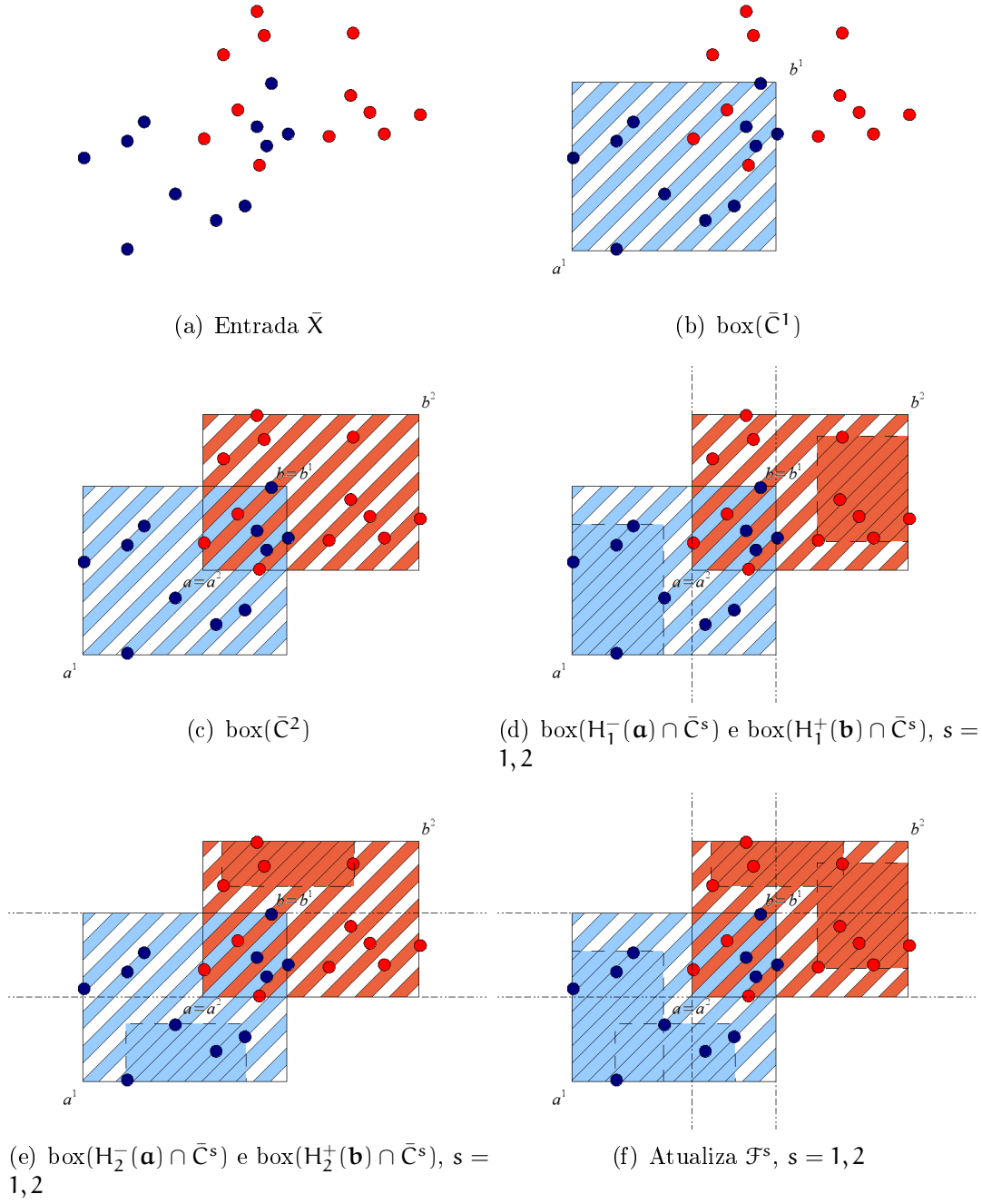


Figura 4.4: Uma execução do Passo **3(a)** do algoritmo MP/CL para classificação binária.

$$\mathcal{F}^s := \mathcal{F}^s \cup \{\{\mathbf{u}\} : \mathbf{u} \in \mathcal{U} \cap \mathcal{C}^s\}, \quad s = 1, 2. \quad (4.19)$$

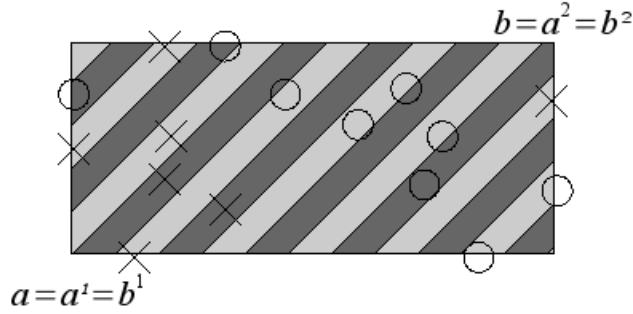


Figura 4.5: Uma rara situação que requer tratamento especial durante a execução do algoritmo do MP/CL.

Em segundo, substituímos em \mathcal{C} o conjunto \bar{X} por $\bar{X} \setminus \bigcup \mathcal{P}$. Lembre que \mathcal{C} representa a família de conjuntos de dados de treinamento que ainda temos que tratar. Em terceiro, definimos $\bar{\mathcal{P}}$ pela remoção das intersecções entre os hiperplanos em \mathcal{P} como se segue

$$\bar{\mathcal{P}} := \{\mathcal{P} \setminus \mathcal{U} : \mathcal{P} \in \mathcal{P}\}. \quad (4.20)$$

Finalmente, atribuímos $\mathcal{C} := \mathcal{C} \cup \{\bar{X} \cap \bar{\mathcal{P}} : \bar{\mathcal{P}} \in \bar{\mathcal{P}} \text{ e } \bar{X} \cap \bar{\mathcal{P}} \neq \emptyset\}$, o que implica que os conjuntos de padrões de treinamento $\bar{X} \cap \bar{\mathcal{P}} \neq \emptyset$, onde $\bar{\mathcal{P}} \in \bar{\mathcal{P}}$, serão tratados nas próximas iterações do algoritmo. Note que na Figura 4.6, $\bar{\mathcal{P}}$ compreende $\mathcal{P} \setminus \mathcal{U}$ e $\mathcal{P}' \setminus \mathcal{U}$.

Após convergir, o algoritmo de treinamento acima obtém duas famílias de hipercaixas: \mathcal{F}^1 e \mathcal{F}^2 . Se \mathbb{L} possui adicionalmente uma estrutura de grupo então cada família de hipercaixas pode ser usada para construir um módulo do MP/CL da forma ilustrada na Figura 4.2. A arquitetura final do MP/CL é dada em termos da combinação destes dois módulos usando $\arg \max$ como na Figura 4.3. Observe que o algoritmo do MP/CL pode ser aplicado sem nenhuma mudança para problemas de classificação binários, em um produto de reticulados $\mathbb{L}_1 \times \dots \times \mathbb{L}_n$ cujos constituintes são correntes. Entretanto, mesmo se cada \mathbb{L}_i para $i = 1, \dots, n$ é um grupo totalmente ordenado, não somos capazes de definir erosões e anti-dilatações da forma ε_v e $\bar{\delta}_w$, uma vez que não se pode tomar o supremo entre as componentes de um vetor pertencente a este produto de reticulados. Ao invés disso, neste caso, podemos recorrer às medidas de inclusão ou distâncias tais como as definidas por Kaburlasos et al. [40, 42]. A Figura 4.7

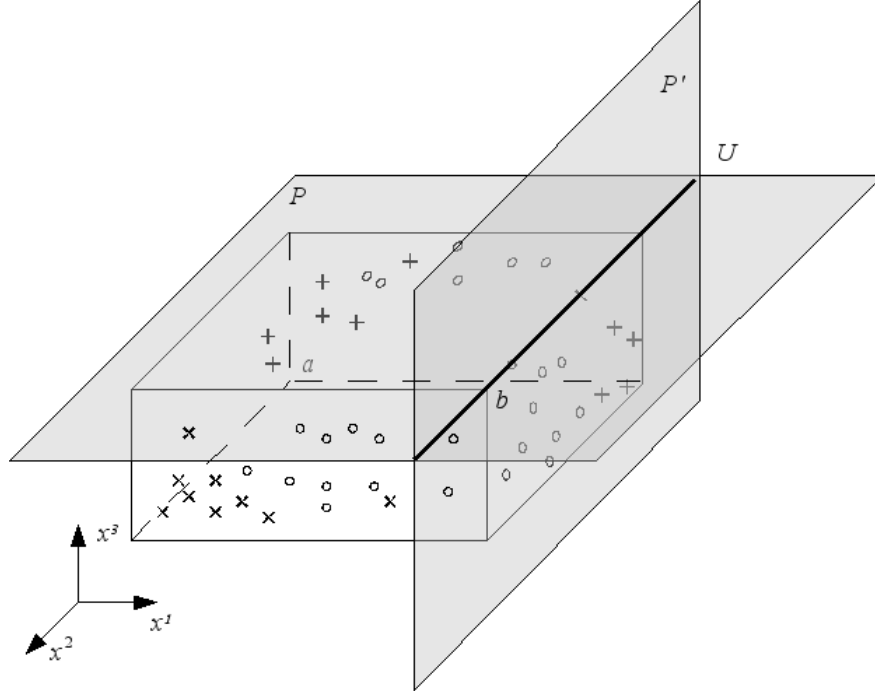


Figura 4.6: Um exemplo de $[\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2]$ em \mathbb{R}^3 incluindo as visualizações dos conjuntos \mathcal{P} e \mathcal{U} .

ilustra um exemplo em \mathbb{R} da superfície de decisão produzida por um MP/CL com dois módulos, onde o primeiro módulo é sintetizado utilizando a hipercaixa $[\mathbf{a}^1, \mathbf{b}^1]$ e o segundo a hipercaixa $[\mathbf{a}^2, \mathbf{b}^2]$.

4.3.2 Propriedades do algoritmo do MP/CL para problemas de classificação binários

Nesta subseção investigaremos as propriedades do algoritmo de treinamento do MP/CL para classificação binária descrito acima em termos de lemas, corolários e teoremas. Quase todos os resultados formulados são demonstrados no decorrer do texto, e para os que não são, indicamos a idéia central de como os mesmos podem ser verificados. Em particular, sondaremos o algoritmo com respeito à independência da ordem com que os padrões de treinamento são apresentados a rede, às regiões de indecisão produzida e à convergência.

Note que, em cada iteração do algoritmo os Passos 1 - 3 são executados para um elemento \bar{X} de \mathcal{C} , onde \bar{X} representa uma família de padrões de treinamento. Nesta subseção mostraremos que o resultado do algoritmo não depende da escolha

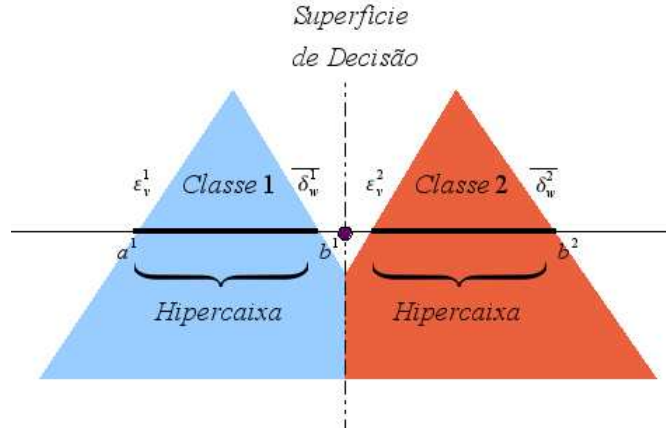


Figura 4.7: Exemplo da superfície de decisão produzida por um MP/CL para um classificação binário em \mathbb{R} .

de \bar{X} em \mathcal{C} a cada iteração.

Para este fim, introduzimos um contador de iteração t e algumas notações com respeito a t . Vamos assumir que o algoritmo inicia com $t = 0$. Seja $\mathcal{F}^s(t)$ a família de hipercaixas referente a s -ésima classe que foram geradas até a conclusão da iteração t , para $s = 1, 2$. O símbolo $\mathcal{C}(t)$ será usado para denotar as famílias de padrões de treinamento que ainda devem ser consideradas após a conclusão da iteração t . Note que $\mathcal{C}(t)$ é da forma $\mathcal{C}(t) = \{X_1(t), \dots, X_{l_t}(t)\}$ para todo $t = 0, 1, \dots$

Lema 5. *Suponha que $\mathcal{C}(t) = \{X_1(t), \dots, X_{l_t}(t)\}$ onde $t \geq 0$, temos que*

$$\text{box}(X_i(t)) \cap \text{box}(X_j(t)) = \emptyset \quad \forall i \neq j. \quad (4.21)$$

Demonstração. Provaremos este lema por indução. Em $t = 0$, temos que $\mathcal{C}(0) = \{X\}$, e portanto não há nada o que mostrar.

Agora suponha que a Equação 4.21 vale para $t = p$ e $\mathcal{C}(p) = \{X_1(p), \dots, X_{l_p}(p)\}$. A menos que $\mathcal{C}(p) = \emptyset$, a iteração $p + 1$ é executada. Sejam $\bar{X} = X_{i^*}$, para algum $i^* \in \{1, \dots, l_p\}$, um elemento arbitrário de $\mathcal{C}(p)$ e $[a^s, b^s]$, para $s = 1, 2$, a menor hipercaixa contendo os padrões em $\bar{X} \cap C^s$. Um dos seguintes casos ocorre:

- $I = [a^1, b^1] \cap [a^2, b^2] = \emptyset$: Neste caso, o passo (2) produz $\mathcal{C}(p + 1) = \mathcal{C}(p) \setminus \{X_{i^*}\} \subseteq \mathcal{C}(p)$. Pela hipótese de indução, a Equação 4.21 vale para $X_i(p + 1), X_j(p + 1) \in \mathcal{C}(p + 1) \subseteq \mathcal{C}(p)$ tal que $i \neq j$.
- $I = [a^1, b^1] \cap [a^2, b^2] \neq \emptyset$:

- (a), $[\mathbf{a}^1, \mathbf{b}^1] \neq [\mathbf{a}^2, \mathbf{b}^2]$: O Passo (3)(a) produz $\mathcal{C}(\mathbf{p} + 1) = [\mathcal{C}(\mathbf{p}) \setminus \{\bar{X}\}] \cup \{\bar{X} \cap I\}$. Seja \tilde{X}, X' dois elementos distintos de $\mathcal{C}(\mathbf{p} + 1)$. Se ambos \tilde{X} e X' estão contidos em $\mathcal{C}(\mathbf{p}) \setminus \{\bar{X}\}$ então $\text{box}(\tilde{X}) \cap \text{box}(X') = \emptyset$ pela hipótese de indução. Agora vamos assumir sem perda de generalidade que $\tilde{X} = \bar{X} \cap I \subseteq \bar{X}$. Imediatamente obtemos $\text{box}(\tilde{X}) \cap \text{box}(X') = \emptyset$ pois $\text{box}(\tilde{X}) \cap \text{box}(X') \subseteq \text{box}(\bar{X}) \cap \text{box}(X') = \emptyset$ pela hipótese de indução.
- (b), $[\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2]$: O Passo (3)(b) produz $\mathcal{C}(\mathbf{p} + 1) = [\mathcal{C}(\mathbf{p}) \setminus \{\bar{X}\}] \cup \{\bar{X} \setminus \bigcup \mathcal{P}\} \cup [\bigcup_{\bar{P} \in \bar{\mathcal{P}}, \bar{X} \cap \bar{P} \neq \emptyset} \{\bar{X} \cap \bar{P}\}]$. Considere um par de conjuntos arbitrários $\tilde{X}, X' \in \mathcal{C}(\mathbf{p} + 1)$ tal que $\tilde{X} \neq X'$.

Se ambos \tilde{X} e X' estão contidos em $\bigcup_{\bar{P} \in \bar{\mathcal{P}}, \bar{X} \cap \bar{P} \neq \emptyset} \{\bar{X} \cap \bar{P}\}$ então $\tilde{X} = \bar{X} \cap \bar{P} \neq \emptyset$ e $X' = \bar{X} \cap \bar{P}' \neq \emptyset$ para alguns $\bar{P}, \bar{P}' \in \bar{\mathcal{P}}$ tal que $\bar{P} \neq \bar{P}'$. Relembre que $\bar{P} \cap \bar{P}' = \emptyset$, o que implica em $\text{box}(\tilde{X}) \cap \text{box}(X') = \text{box}(\bar{X} \cap \bar{P}) \cap \text{box}(\bar{X} \cap \bar{P}') = \emptyset$ devido a $\text{box}(\bar{X} \cap \bar{P}) \subseteq \bar{P}$ e $\text{box}(\bar{X} \cap \bar{P}') \subseteq \bar{P}'$.

Agora vamos considerar a situação onde um dos conjuntos \tilde{X}, X' está contido em $\bigcup_{\bar{P} \in \bar{\mathcal{P}}, \bar{X} \cap \bar{P} \neq \emptyset} \{\bar{X} \cap \bar{P}\}$ e um outro conjunto é igual a $\bar{X} \setminus \bigcup \mathcal{P}$. Sem perda de generalidade, vamos assumir que $\tilde{X} = \bar{X} \cap \bar{P} \neq \emptyset$ para algum $\bar{P} \in \bar{\mathcal{P}}$ e que $X' = \bar{X} \setminus \bigcup \mathcal{P}$. Temos que $\bar{P} \subseteq P_i(\mathbf{b})$ para algum $i \in \{1, \dots, n\}$. Como $X' = \bar{X} \setminus \bigcup \mathcal{P} \subseteq \bar{X} \setminus P_i(\mathbf{b})$ segue que $x'_i < b_i$ para todo $\mathbf{x}' \in X'$. A última afirmação não apenas implica que $X' \subseteq \mathbb{L}^n \setminus \bar{P}$ mas também que $\text{box}(X') \subseteq \mathbb{L}^n \setminus \bar{P}$. Portanto, concluímos que $\text{box}(\tilde{X}) \cap \text{box}(X') = \emptyset$.

Suponha que $X' \in \mathcal{C}(\mathbf{p}) \setminus \{X\}$ e que $\tilde{X} = \bar{X} \cap \bar{P} \neq \emptyset$ para algum $\bar{P} \in \bar{\mathcal{P}}$. Como $\tilde{X} \subseteq \bar{X}$, temos $\text{box}(\tilde{X}) \subseteq \text{box}(\bar{X})$. A igualdade $\text{box}(\bar{X}) \cap \text{box}(X') = \emptyset$ segue da hipótese de indução para $\bar{X}, X' \in \mathcal{C}(\mathbf{p})$. Logo $\text{box}(\tilde{X}) \cap \text{box}(X') = \emptyset$.

Similarmente, se $X' \in \mathcal{C}(\mathbf{p}) \setminus \{X\}$ e $\tilde{X} = \bar{X} \setminus \bigcup \mathcal{P}$, então, também temos $\tilde{X} \subseteq \bar{X}$, o que implica em $\text{box}(\tilde{X}) \subseteq \text{box}(\bar{X})$. Novamente, uma aplicação da hipótese de indução provê o fato de que $\text{box}(\tilde{X})$ e $\text{box}(X')$ não se interceptam.

Finalmente, no caso onde ambos \tilde{X} e X' estão contidos em $\mathcal{C}(\mathbf{p}) \setminus \{X\}$, o resultado desejado decorre imediatamente da hipótese de indução.

□

Como consequência do Lema 5, obtemos o seguinte colorário:

Corolário 2. *A intersecção de dois elementos distintos $X_i(t)$ e $X_j(t)$ de $\mathcal{C}(t)$ para $i \neq j$ é vazio.*

O Corolário 2 é utilizado para provar que o algoritmo de aprendizagem do MP/CL não depende de escolhas particulares de \bar{X} no Passo (1), porque todos os conjuntos que são utilizados para construir as hipercaixas resultantes do Passo (2), (3)(a) ou (3)(b) são disjuntos, dois a dois. Este fato é expresso no seguinte corolário [91].

Corolário 3. *As famílias de hipercaixas \mathcal{F}^s , onde $s = 1, 2$, produzidas pelo algoritmo de treinamento do MP/CL após convergir e o número de iterações até convergir, não dependem de escolhas particulares de \bar{X} no Passo (1). Em particular, o resultado e o tempo de convergência do algoritmo, independem da ordem com que os padrões de treinamento são apresentados à rede.*

A figura a seguir visualiza duas árvores que correspondem a duas aplicações diferentes do algoritmo do MP/CL no mesmo problema de classificação. O t^o nível das árvores contém o elemento $\mathcal{C}(t)$ para as duas instâncias distintas do algoritmo. Por exemplo, a raiz de cada árvore corresponde ao conjunto inicializado $\mathcal{C} \equiv \mathcal{C}(0)$, que contém apenas o conjunto inteiro de padrões de treinamento X . O círculo em torno de um certo conjunto $X_i(t)$, significa que o mesmo é selecionado pelo Passo (1) durante a iteração $t + 1$, i.e., $\bar{X} = X_i(t)$. Os rótulos entre os níveis de uma certa árvore denota qual passo é executado pelo algoritmo subsequente ao conjunto \bar{X} correspondente ao nó selecionado.

Repare que na Figura 4.8 em ambas as instâncias do algoritmo, o Passo (3)(b) é executado na segunda iteração, indicando que a condição $[\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2] \neq \emptyset$ é satisfeita. Tal situação é visualizada na Figura 4.6 onde os padrões em $\bar{X} \cap \mathcal{U}$ são removidos de \bar{X} e o restante $(\bar{X} \setminus \mathcal{U})$ é particionado nos conjuntos $\bar{X} \setminus \bigcup \mathcal{P}$, $\bar{X} \cap \bar{\mathcal{P}}$ e $\bar{X} \cap \bar{\mathcal{P}}'$ para formar $\mathcal{C}(2)$. Note que as duas instâncias do algoritmo diferem da escolha de \bar{X} na terceira iteração, em particular, a instância da esquerda considera $\bar{X} = X_2(2)$ enquanto a instância da direita considera $\bar{X} = X_3(2)$. As arestas tracejadas entre os níveis de uma árvore são usadas para representar a cópia dos elementos não escolhidos de $\mathcal{C}(t)$ na iteração $t + 1$ para o conjunto $\mathcal{C}(t + 1)$. Lembre que no Passo (2), executado quando a condição $[\mathbf{a}^1, \mathbf{b}^1] \cap [\mathbf{a}^2, \mathbf{b}^2] = \emptyset$ ocorre, o conjunto $\mathcal{C}(t + 1)$ é formado removendo \bar{X} de $\mathcal{C}(t)$, para simbolizar este procedimento utilizamos um quadrado após a execução do Passo (2). Por fim,

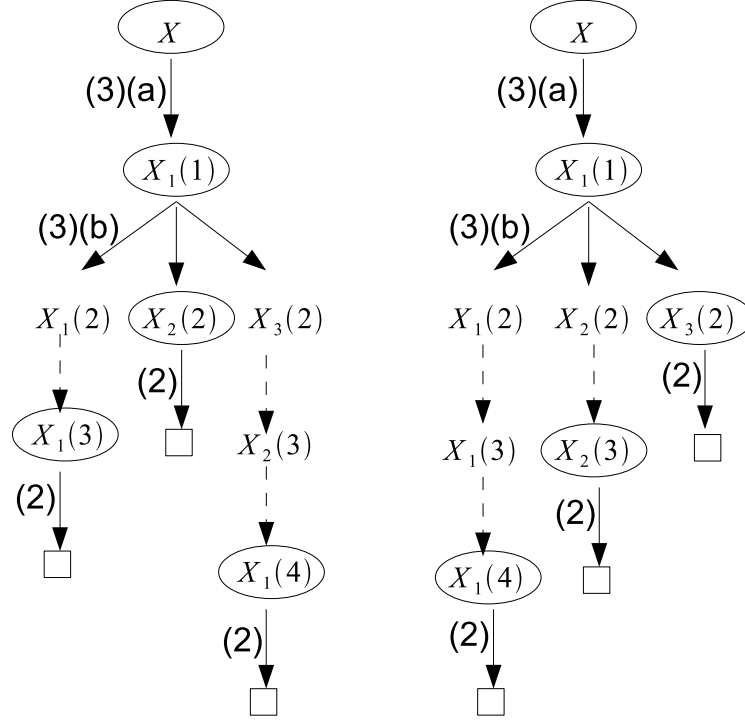


Figura 4.8: Duas diferentes instâncias do algoritmo do MP/CL.

repare que ambas as instâncias dos algoritmos convergem em 5 iterações, como sugerido pelo Corolário 3.

Note que em uma iteração $t + 1$, cada família de padrões de treinamento que não foi selecionada de $\mathcal{C}(t)$ no Passo (1), i.e., cada $X' \in \mathcal{C}(t)$ tal que $X' \neq \bar{X}$ permanece inalterada e é transferida para $\mathcal{C}(t + 1)$, e assim, sucessivamente até que X' seja escolhido no Passo (1) em alguma iteração $t^* > t$. Dessa maneira, as atualizações nas famílias de hipercaixas \mathcal{F}^s para $s = 1, 2$ provenientes da atribuição $\bar{X} = X'$ independe de qual iteração isto ocorre. Esta última observação é utilizada como idéia central para demonstrar o Corolário 3.

Outra excelente propriedade exibida pelo algoritmo é que não existe sobreposições entre hipercaixas com rótulos de classes distintos em $\mathcal{F}^1 \cup \mathcal{F}^2$. Para provar esta afirmação é necessário apresentar outros dois lemas.

Lema 6. *Seja $t \geq 0$. As hipercaixas $\text{box}(X')$ e B são disjuntas para todo $X' \in \mathcal{C}(t)$ e para todo $B \in \bigcup_{s=1}^2 \mathcal{F}^s(t)$.*

Demonstração. Primeiramente, para $t = 0$ temos que $\mathcal{C}(0) = \{X\}$ e $\mathcal{F}^1(0) = \mathcal{F}^2(0) = \emptyset$. Logo, não há nada o que mostrar, já que não existe nenhum $B \in \bigcup_{s=1}^2 \mathcal{F}^s(t)$.

Suponha que a afirmação do lema é verdade para $t = p$. Se $\mathcal{C}(p) = \emptyset$ então nada resta a ser mostrado. Caso contrário, a iteração $p + 1$ é executada. No Passo (1) um $\bar{X} \in \mathcal{C}(p)$ arbitrário é selecionado.

Se o Passo (2) é invocado, então, $\mathcal{F}^s(p + 1)$ e $\mathcal{C}(p + 1)$ são respectivamente gerados, adicionando as hipercaixas $\text{box}(\bar{C}^s) = [\mathbf{a}^s, \mathbf{b}^s]$ à $\mathcal{F}^s(p)$ e removendo \bar{X} de $\mathcal{C}(p)$. Formalmente, temos que:

$$\mathcal{F}^s(p+1) = \mathcal{F}^s(p) \cup \{\text{box}(\bar{C}^s)\} = \mathcal{F}^s(p) \cup \{[\mathbf{a}^s, \mathbf{b}^s]\}, s = 1, 2, \quad \text{e} \quad \mathcal{C}(p+1) = \mathcal{C}(p) \setminus \{\bar{X}\}. \quad (4.22)$$

Para todo $X' \in \mathcal{C}(p + 1) \subseteq \mathcal{C}(p)$ e todo $B \in \mathcal{F}^s(p)$, onde $s = 1, 2$, temos que $\text{box}(X') \cap B = \emptyset$ pela hipótese de indução. Ainda falta mostrar que $\text{box}(\bar{C}^s) \cap \text{box}(X') = \emptyset$ para $s = 1, 2$. Como $\bar{C}^s = \bar{X} \cap C^s \subseteq \bar{X}$ deduzimos que $\text{box}(\bar{C}^s) \subseteq \text{box}(\bar{X})$ e, portanto, $\text{box}(\bar{C}^s) \cap \text{box}(X') \subseteq \text{box}(\bar{X}) \cap \text{box}(X') = \emptyset$ pelo Lema 5, dado que $\bar{X}, X' \in \mathcal{C}(p)$.

Suponha que $[\mathbf{a}^1, \mathbf{b}^1] \cap [\mathbf{a}^2, \mathbf{b}^2] \neq \emptyset$. Se $[\mathbf{a}^1, \mathbf{b}^1]$ difere de $[\mathbf{a}^2, \mathbf{b}^2]$, então, o Passo (3)(a) é executado produzindo

$$\begin{aligned} \mathcal{F}^s(p + 1) &= \mathcal{F}^s(p) \cup \bigcup_{i=1}^n \{\text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)\} \cup \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}, s = 1, 2, \\ \text{e} \quad \mathcal{C}(p + 1) &= (\mathcal{C}(p) \setminus \{\bar{X}\}) \cup \{\bar{X} \cap I\}. \end{aligned} \quad (4.23)$$

Seja $s \in \{1, 2\}$. Suponha que $X' \in \mathcal{C}(p) \setminus \{\bar{X}\}$. Para $B \in \mathcal{F}^s(p)$, já sabemos que $\text{box}(X') \cap B = \emptyset$ pela hipótese de indução. Se $B = \text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)$ para algum índice i , então, o fato que $H_i^-(\mathbf{a}) \cap \bar{C}^s \subseteq \bar{X}$ implica que $B \subseteq \text{box}(\bar{X})$, donde obtemos $\text{box}(X') \cap B \subseteq \text{box}(X') \cap \text{box}(\bar{X}) = \emptyset$ pelo Lema 5. Similarmente, verificamos que $\text{box}(X') \cap B = \emptyset$ para todo $X' \in \mathcal{C}(p) \setminus \{\bar{X}\}$ e para todo $B \in \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}$.

Agora, no caso onde $X' = \bar{X} \cap I$ temos que a relação $\text{box}(X') \subseteq \text{box}(\bar{X})$ vale. Portanto, como $\bar{X} \in \mathcal{C}(p)$, temos pela hipótese de indução que $\text{box}(X') \cap B \subseteq \text{box}(\bar{X}) \cap B = \emptyset$ para todo $B \in \mathcal{F}^s(p)$. Também sabemos que $X' = \bar{X} \cap I$ implica que $\text{box}(X') \subseteq I$. Dado $B = \text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)$ para algum $i \in \{1, \dots, n\}$, temos que $B \subseteq H_i^-(\mathbf{a})$. A última afirmação produz $\text{box}(X') \cap B = \emptyset$. Similarmente, podemos mostrar que $\text{box}(X') \cap B = \emptyset$ para $X' = \bar{X} \cap I$ e para todo $B \in \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}$.

Finalmente, vamos lidar com a situação $[\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2]$ que exige a exe-

cução do Passo (3)(b), resultando em

$$\begin{aligned} \mathcal{F}^s(\mathbf{p} + 1) &= \mathcal{F}^s(\mathbf{p}) \cup \{\{\mathbf{u}\} : \mathbf{u} \in \mathcal{U} \cap \bar{\mathcal{C}}^s\}, s = 1, 2, \\ \text{e } \mathcal{C}(\mathbf{p} + 1) &= [\mathcal{C}(\mathbf{p}) \setminus \{\bar{X}\}] \cup \{\bar{X} \setminus \bigcup \mathcal{P}\} \cup \left[\bigcup_{\bar{P} \in \bar{\mathcal{P}}, \bar{X} \cap \bar{P} \neq \emptyset} \{\bar{X} \cap \bar{P}\} \right]. \end{aligned} \quad (4.24)$$

Novamente, seja $s \in \{1, 2\}$ arbitrário. Suponha que $X' \in \mathcal{C}(\mathbf{p}) \setminus \{\bar{X}\}$. Para $B \in \mathcal{F}^s(\mathbf{p})$, temos que $\text{box}(X') \cap B = \emptyset$ pela hipótese de indução. O próximo caso a ser considerado é $B \in \{\{\mathbf{u}\} : \mathbf{u} \in \mathcal{U} \cap \bar{\mathcal{C}}^s\}$, i.e., $B = \{\mathbf{u}\} = [\mathbf{u}, \mathbf{u}]$ para algum $\mathbf{u} \in \mathcal{U}$. Note que $B = \{\mathbf{u}\} \subseteq \mathcal{U} \cap \bar{\mathcal{C}}^s \subseteq \bar{X}$. Portanto, a igualdade $\text{box}(X') \cap B = \emptyset$ decorre do Lema 5.

Agora, vamos assumir que $X' = \bar{X} \setminus \bigcup \mathcal{P} \subseteq \bar{X}$. Se $B \in \mathcal{F}^s(\mathbf{p})$ então $\text{box}(X') \cap B \subseteq \text{box}(\bar{X}) \cap B = \emptyset$, o que revela que $\text{box}(X') \cap B = \emptyset$ como desejado. Note que existe pelo menos um índice i tal que $P_i(\mathbf{b}) \in \mathcal{P}$ e que todo elemento \mathbf{x} de X' satisfaz $x_i < b_i$ para todo $P_i(\mathbf{b}) \in \mathcal{P}$. Logo, como X' é finito temos que $\text{box}(X') \cap P_i(\mathbf{b}) = \emptyset$ para todo $P_i(\mathbf{b}) \in \mathcal{P}$. Em particular, temos que $\text{box}(X') \cap B = \emptyset$ se $B = \{\mathbf{u}\} \subseteq \bigcup \mathcal{P}$ para algum $\mathbf{u} \in \mathcal{U} \cap \bar{\mathcal{C}}^s$.

Por fim, vamos considerar o caso onde $X' = \bar{X} \cap \bar{P} \neq \emptyset$ para algum $\bar{P} \in \bar{\mathcal{P}}$. Como anteriormente, se $B \in \mathcal{F}^s(\mathbf{p})$, então, $\text{box}(X') \cap B \subseteq \text{box}(\bar{X}) \cap B = \emptyset$. Se $B = \{\mathbf{u}\}$ para algum $\mathbf{u} \in \mathcal{U} \cap \bar{\mathcal{C}}^s$, então, $\mathbf{u} \in P \cap P'$ para alguns $P = P_i(\mathbf{b}), P' = P_j(\mathbf{b}) \in \mathcal{P}$ tal que $i \neq j$, i.e., $u_i = b_i$ e $u_j = b_j$. Por hipótese, $X' \subseteq \bar{P}$ para algum $\bar{P} \in \bar{\mathcal{P}} = \{P \setminus \mathcal{U} : P \in \mathcal{P}\}$. Como $X' \subseteq P \setminus \mathcal{U}$ para algum $P \in \mathcal{P}$, existe um índice i^* tal que $P = P_{i^*}(\mathbf{b}) \in \mathcal{P}$ e $x_{i^*} = b_{i^*}$ para todo $\mathbf{x} \in X'$. Temos que $i \neq i^*$ e/ou $j \neq i^*$, suponha, sem perda de generalidade, que $j \neq i^*$. Como $P_{i^*}(\mathbf{b}) \cap P_j(\mathbf{b}) \subseteq \mathcal{U}$ segue que $x_j < b_j$ para todo $\mathbf{x} \in X'$. O fato de X' ser finito, garante que $(\bigvee X')_j < b_j = u_j$ e consequentemente $\text{box}(X') \cap B = \emptyset$.

□

Lema 7. Se $B^1 \in \mathcal{F}^1(\mathbf{t})$ e $B^2 \in \mathcal{F}^2(\mathbf{t})$ então $B^1 \cap B^2 = \emptyset$.

Demonstração. A prova deste lema é dada por indução. Para $\mathbf{t} = \mathbf{0}$ temos $\mathcal{F}^1(\mathbf{0}) = \mathcal{F}^2(\mathbf{0}) = \emptyset$. Logo não há nada o que mostrar.

Suponha que a afirmação do lema é verdade para $\mathbf{t} = \mathbf{p}$. Se $\mathcal{C}(\mathbf{p}) = \emptyset$, então, de novo não há nada o que mostrar. Caso contrário, a iteração $\mathbf{p} + 1$ é executada. No Passo (1) um $\bar{X} \in \mathcal{C}(\mathbf{p})$ arbitrário é selecionado. Em seguida, uma das seguintes situações ocorre:

- Se $[\mathbf{a}^1, \mathbf{b}^1] \cap [\mathbf{a}^2, \mathbf{b}^2] = \emptyset$ então o Passo (2) será executado produzindo

$$\mathcal{F}^s(\mathbf{p} + 1) = \mathcal{F}^s(\mathbf{p}) \cup [\mathbf{a}^s, \mathbf{b}^s], s = 1, 2. \quad (4.25)$$

Seja $B^1 \in \mathcal{F}^1(\mathbf{p} + 1)$ e $B^2 \in \mathcal{F}^2(\mathbf{p} + 1)$. Se $B^s \in \mathcal{F}^s(\mathbf{p})$ para $s = 1, 2$ então $B^1 \cap B^2 = \emptyset$ pela hipótese de indução. Se $B^s \in \mathcal{F}^s(\mathbf{p})$ e $B^r = [\mathbf{a}^r, \mathbf{b}^r] \subseteq \text{box}(\bar{X})$ para $r \neq s$ então $B^s \cap B^r = \emptyset$ pelo Lema 6. Finalmente, se $B^s = [\mathbf{a}^s, \mathbf{b}^s]$ para $s = 1, 2$, então, a intersecção $B^s \cap B^r$ é nula por hipótese.

- Se $[\mathbf{a}^1, \mathbf{b}^1] \cap [\mathbf{a}^2, \mathbf{b}^2] = I \neq \emptyset$ e $[\mathbf{a}^1, \mathbf{b}^1] \neq [\mathbf{a}^2, \mathbf{b}^2]$, então, o Passo (3)(a) produz

$$\mathcal{F}^s(\mathbf{p} + 1) = \mathcal{F}^s(\mathbf{p}) \cup \bigcup_{i=1}^n \{\text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)\} \cup \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}, s = 1, 2. \quad (4.26)$$

Primeiramente, seja $B^s \in \mathcal{F}^s(\mathbf{p})$ para algum $s = 1, 2$. Sem perda de generalidade assumimos que $B^1 \in \mathcal{F}^1(\mathbf{p})$. Novamente, o caso $B^2 \in \mathcal{F}^2(\mathbf{p})$ segue imediatamente da hipótese de indução. Agora, seja $B^2 = \text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^2)$ para algum índice arbitrário i . Como $H_i^-(\mathbf{a}) \cap \bar{C}^2 \subseteq \bar{X} \in \mathcal{C}(\mathbf{p})$, as hipercaixas B^1 e B^2 não se interseptam de acordo com o Lema 6. Um argumento similar revela que $B^1 \cap B^2 = \emptyset$ se $B_2 \in \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}$.

Agora, seja $B^s \in \bigcup_{i=1}^n \{\text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)\} \cup \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}$ para $s = 1, 2$. Temos que $B^1 \cap B^2 \subseteq I$ e $I \cap [\bigcup_{i=1}^n H_i^-(\mathbf{a}) \cup \bigcup_{i=1}^n H_i^+(\mathbf{b})] = \emptyset$. Como $B^s \subseteq \text{box}(\bar{C}^s) \cap [\bigcup_{i=1}^n H_i^-(\mathbf{a}) \cup \bigcup_{i=1}^n H_i^+(\mathbf{b})]$, para $s = 1, 2$, segue que $B^1 \cap B^2 = \emptyset$.

- Se $[\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2] \neq \emptyset$ então o Passo (3)(b) resulta em

$$\mathcal{F}^s(\mathbf{p} + 1) = \mathcal{F}^s(\mathbf{p}) \cup \{\{\mathbf{u}\} : \mathbf{u} \in \mathcal{U} \cap \bar{C}^s\}, s = 1, 2. \quad (4.27)$$

Considere $B^s \in \mathcal{F}^s(\mathbf{p} + 1)$ arbitrários, para $s = 1, 2$. Sabemos pela hipótese de indução que B^1 e B^2 tais que $B^s \in \mathcal{F}^s(\mathbf{p})$ para $s = 1, 2$ não se interseptam. Agora, vamos assumir sem perda de generalidade que $B^1 \in \mathcal{F}^1(\mathbf{p})$ e que $B^2 = \{\mathbf{u}\}$ para algum $\mathbf{u} \in \mathcal{U} \cap \bar{C}^s$. Observe que $B^2 = \{\mathbf{u}\} \subseteq \bar{C}^s \subseteq \bar{X} \in \mathcal{C}(\mathbf{p})$. Uma breve olhada no Lema 6 revela que $B^1 \cap B^2 = \emptyset$. Finalmente, a

intersecção $B^1 \cap B^2$ é obviamente vazia se $B^s \in \{\{u\} : u \in U \cap \bar{C}^s\}$ para $s = 1, 2$ pois $\bar{C}^1 \cap \bar{C}^2 = \emptyset$.

□

O Lema 4 estabelece que não existem sobreposições de hipercaixas em $\mathcal{F}^1(t)$ com hipercaixas em $\mathcal{F}^2(t)$ no estágio t do algoritmo. Em particular, temos

Corolário 4. *Após convergir, o algoritmo de treinamento do MP/CL para classificação binária produz famílias de hipercaixas \mathcal{F}^1 e \mathcal{F}^2 cujos elementos não se interseptom, i.e.,*

$$B^1 \cap B^2 = \emptyset \quad \forall B^1 \in \mathcal{F}^1, B^2 \in \mathcal{F}^2. \quad (4.28)$$

Vamos agora investigar a convergência do algoritmo, em especial, provaremos que o algoritmo converge em um número finito de passos. Para isto, vamos usar o seguinte fato que pode ser facilmente verificado por indução: todo \bar{X} em $\mathcal{C}(t)$ é não vazio.

Lema 8. *Todo elemento de $\mathcal{C}(t)$ é não vazio.*

Demonstração. Primeiramente, se considerarmos $t = 0$ teremos $\mathcal{C}(0) = \{X\}$ o que está de acordo com o lema. Suponha que a afirmação do lema seja verdade para $t = p$. Se $\mathcal{C}(p) = \emptyset$ então não resta nada o que provar. Caso contrário, a iteração $p + 1$ é executada e um $\bar{X} \in \mathcal{C}(p)$ arbitrário é selecionado no Passo (1).

Se $I = \emptyset$ então o Passo (2) é invocado e $\mathcal{C}(p + 1)$ é gerado removendo \bar{X} de $\mathcal{C}(p)$. Neste caso, todo $X' \in \mathcal{C}(p + 1)$ está contido em $\mathcal{C}(p)$ e $X' \neq \emptyset$ pela hipótese de indução.

Caso contrário, se $I \neq \emptyset$, existem dois casos que temos de considerar: $[a^1, b^1] \neq [a^2, b^2]$ e $[a^1, b^1] = [a^2, b^2]$.

No primeiro caso, o Passo (3)(a) é chamado. Se $\bar{X} \cap I = \emptyset$ então $\mathcal{C}(p + 1) = \mathcal{C}(p) \setminus \{\bar{X}\}$ e os mesmos argumentos acima podem ser empregados para mostrar que $X' \neq \emptyset$ para todo $X' \in \mathcal{C}(p + 1)$. Caso contrário, $\mathcal{C}(p + 1)$ é dado por $[\mathcal{C}(p) \setminus \{\bar{X}\}] \cup \{\bar{X} \cap I\}$. Já sabemos pela hipótese de indução que todo $X' \in \mathcal{C}(p) \setminus \{\bar{X}\}$ é diferente do conjunto vazio e, além disso, $\bar{X} \cap I \neq \emptyset$ por suposição. Logo, temos o desejado.

No segundo caso, o Passo (3)(b) produz

$$\mathcal{C}(p+1) = [\mathcal{C}(p) \setminus \{\bar{X}\}] \cup \left\{ \bar{X} \setminus \bigcup \mathcal{P} \right\} \cup \left[\bigcup_{\bar{P} \in \bar{\mathcal{P}}, \bar{X} \cap \bar{P} \neq \emptyset} \{\bar{X} \cap \bar{P}\} \right]. \quad (4.29)$$

Seja $X' \in \mathcal{C}(p+1)$. Dadas as observações feitas acima, apenas o caso $X' = \bar{X} \setminus \bigcup \mathcal{P}$ precisa ser estudado. Note que, a desigualdade $\mathbf{a} < \mathbf{b}$ é satisfeita pois senão teríamos que \mathbf{a} pertence tanto a $\bar{\mathcal{C}}^1$ quanto a $\bar{\mathcal{C}}^2$, contradizendo a suposição que $\mathcal{C}^1 \cap \mathcal{C}^2 = \emptyset$. Consequentemente, existe um padrão $\mathbf{x} \in \bar{X}$ tal que $x_i = a_i < b_i$ para algum índice i , o que implica que $\mathbf{x} \notin \bigcup \mathcal{P}$. Em outras palavras, $X' \neq \emptyset$.

□

Lema 9. *Seja $A(t)$ o conjunto de padrões de treinamento $\mathbf{x} \in X$ que estão contidos em alguma hipercaixa $B_{\mathbf{x}} \in \mathcal{F}^1(t) \cup \mathcal{F}^2(t)$. Se $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$, então, $X = [\bigcup \mathcal{C}(t)] \cup A(t)$ e $[\bigcup \mathcal{C}(t)] \cap A(t) = \emptyset$ para todo $t \geq 0$.*

Demonstração. Para todo $t \geq 0$ ambos $A(t)$ e $\bigcup \mathcal{C}(t)$ são subconjuntos de X . Portanto, temos $[\bigcup \mathcal{C}(t)] \cup A(t) \subseteq X$. Vamos mostrar que $X \subseteq [\bigcup \mathcal{C}(t)] \cup A(t)$ por indução.

Para $t = 0$, temos $\mathcal{C}(0) = \{X\}$ o que implica que $X \subseteq [\bigcup \mathcal{C}(0)] \cup A(0)$. Vamos assumir que $X \subseteq [\bigcup \mathcal{C}(p)] \cup A(p)$ para $t = p$.

Note que, é suficiente mostrar que $[\bigcup \mathcal{C}(p)] \cup A(p) \subseteq [\bigcup \mathcal{C}(p+1)] \cup A(p+1)$ para concluir que $X \subseteq [\bigcup \mathcal{C}(p+1)] \cup A(p+1)$. De fato, basta verificar que $[\bigcup \mathcal{C}(p)] \subseteq [\bigcup \mathcal{C}(p+1)] \cup A(p+1)$, pois, $A(p) \subseteq A(p+1)$. Para isto, vamos tomar um elemento arbitrário \mathbf{x} de $\bigcup \mathcal{C}(p)$ e verificar que $\mathbf{x} \in [\bigcup \mathcal{C}(p+1)] \cup A(p+1)$.

Se $\mathcal{C}(p) = \emptyset$ então nada resta para ser mostrado. Caso contrário, a iteração $p+1$ é executada o que implica que um $\bar{X} \in \mathcal{C}(p)$ arbitrário é selecionado no Passo (1). O fato que $\mathbf{x} \in \bigcup \mathcal{C}(p)$ implica que $\mathbf{x} \in X'$ para algum $X' \subseteq \mathcal{C}(p)$. Este conjunto X' é único pois os subconjuntos de X que residem em $\mathcal{C}(p)$ são disjuntos pelo Corolário 2. Se $X' \neq \bar{X}$ então $X' \in \mathcal{C}(p+1)$ e, portanto, \mathbf{x} está localizado em $\bigcup \mathcal{C}(p+1)$. O caso onde $\mathbf{x} \in X' = \bar{X}$ requer algumas examinações mais detalhadas do algoritmo:

- Suponha que o Passo (2) é executado. Dados os fatos que $\bar{\mathcal{C}}^1 \subseteq [\mathbf{a}^1, \mathbf{b}^1] \in \mathcal{F}^1(p+1)$ e $\bar{\mathcal{C}}^2 \subseteq [\mathbf{a}^2, \mathbf{b}^2] \in \mathcal{F}^2(p+1)$, concluímos que $\bar{X} = \bar{\mathcal{C}}^1 \cup \bar{\mathcal{C}}^2 \subseteq A(p+1)$.

- Vamos considerar o Passo **(3)(a)**. Recorde que $\bar{X} \cap I \in \mathcal{C}(p+1)$ se $\bar{X} \cap I \neq \emptyset$. Portanto, temos $\mathbf{x} \in \bigcup \mathcal{C}(p+1)$ se $\mathbf{x} \in \bar{X} \cap I$. Agora, se $\mathbf{x} \notin \bar{X} \cap I$ então existe um índice i tal que $\mathbf{x} \in H_i^-(\mathbf{a})$ ou $\mathbf{x} \in H_i^+(\mathbf{b})$, o que implica que existe um rótulo de classe s tal que $\mathbf{x} \in (H_i^-(\mathbf{a}) \cap \bar{C}^s) \cup (H_i^+(\mathbf{b}) \cap \bar{C}^s) \subseteq A(p+1)$.
- Finalmente, vamos considerar o Passo **(3)(b)**. Como anteriormente, se $\mathbf{x} \in U \cap \bar{X}$ temos que $\mathbf{x} \in A(p+1)$. Por fim, note que, se $\mathbf{x} \in \bar{P} \cap \bar{X}$ para algum $\bar{P} \in \bar{\mathcal{P}}$ ou se $\mathbf{x} \in \bar{X} \setminus \bigcup \mathcal{P}$, então, obtemos que $\mathbf{x} \in \bigcup \mathcal{C}(p+1)$.

Para sumarizar, obtemos que $X = [\bigcup \mathcal{C}(t)] \cup A(t)$ para todo $t = 0, 1, \dots$

A segunda parte do lema é verificada a partir das implicações a seguir. Para um $t \geq 0$ arbitrário, suponha $\mathbf{x} \in [\bigcup \mathcal{C}(t)] \cap A(t)$. Por um lado, temos que $\mathbf{x} \in B$ para algum $B \in \bigcup_{s=1}^2 \mathcal{F}^s(t)$ pois $\mathbf{x} \in A(t)$. Por outro lado, \mathbf{x} está localizado em $X' \subseteq \text{box}(X')$ para algum $X' \in \mathcal{C}(t)$. Portanto, a aplicação do Lema 6 leva à contradição $\mathbf{x} \in B \cap \text{box}(X') = \emptyset$.

□

Agora, estamos prontos para provar que o algoritmo converge em um número finito de iterações. Pelo Lema 9, o critério de parada $\mathcal{C}(t) = \emptyset$ é atingido se, e somente se, $A(t) = X$. Logo, é suficiente mostrar que existe uma iteração p tal que $A(p)$ (o conjunto dos padrões de treinamento que já foram classificados até a conclusão da iteração p) é igual a X .

Com a exceção de situações extremamente raras tal como ilustrada na Figura 4.5, apenas os Passos **(1)**, **(2)** e **(3)(a)** são executados no algoritmo. Portanto, inicialmente, focaremos nos casos usuais onde a situação $I = [\mathbf{a}^1, \mathbf{b}^1] = [\mathbf{a}^2, \mathbf{b}^2]$, que leva à execução do Passo **(3)(b)**, nunca acontece.

Teorema 9. *Considere um problema de classificação binário com um conjunto de dados de treinamento $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \mathbb{L}^n$, onde \mathbb{L} é uma corrente. As seguintes afirmações são válidas:*

1. *Se o Passo **(3)(b)** não for executado em uma aplicação do algoritmo de treinamento do MP/CL, então temos $|A(t+1)| \geq |A(t)| + 1$ para todo $t = 0, 1, \dots$ e o algoritmo converge no máximo em k iterações.*
2. *Se nenhuma suposição adicional que impeça a execução do Passo **(3)(b)** for considerada, então o algoritmo converge em menos de $2k$ iterações.*

Demonstração. Primeiramente, suponha que o Passo **(3)(b)** não é invocado durante a execução do algoritmo. Suponha que $t \in \mathbb{N} \cup \{0\}$ iterações do algoritmo já tenham sido completadas e que $\mathcal{C}(t) \neq \emptyset$ ou, equivalentemente, que $A(t) \neq X$ (conforme o Lema 9). Após a escolha de um elemento \bar{X} em $\mathcal{C}(t)$, as menores hipercaixas $[\mathbf{a}^s, \mathbf{b}^s]$ contendo todos os elementos de $\bar{C}^s = \bar{X} \cap C^s$ são construídas em \mathbb{L}^n . Subsequentemente, o algoritmo invoca ou o Passo **(2)** ou o Passo **(3)(a)**. No primeiro caso, $|A(t+1)| = |A(t)| + |\bar{X}| \geq |A(t)| + 1$, pois, $\bar{X} \cap A(t) = \emptyset$ pelo Lema 9 e $\bar{X} \neq \emptyset$ pelo Lema 8. No segundo caso, temos $[\mathbf{a}^1, \mathbf{b}^1] \neq [\mathbf{a}^2, \mathbf{b}^2]$ e $A(t+1) = A(t) \cup [\bar{X} \setminus (\bar{X} \cap I)]$, onde $I = [\mathbf{a}^1, \mathbf{b}^1] \cap [\mathbf{a}^2, \mathbf{b}^2]$. Sem perda de generalidade, podemos assumir que $\mathbf{a}^1 \neq \mathbf{a}^2$. Logo, \mathbf{a}^1 e \mathbf{a}^2 diferem em pelo menos em uma componente. Por isso, podemos assumir adicionalmente sem perda de generalidade que $\mathbf{a}_i^1 < \mathbf{a}_i^2$ para algum índice i , o que implica que existe \mathbf{x} pertencente à $\bar{C}^1 \setminus [\bar{X} \cap I]$ tal que $x_i = \mathbf{a}_i^1$. Logo $\mathbf{x} \in \text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^1)$ e, conseqüentemente, $|A(t+1)| \geq |A(t)| + 1$. Dessa maneira, concluímos que existe um índice $p \leq k$ tal que $A(p) = X$. Em outras palavras, o critério de parada $\mathcal{C} = \emptyset$ é atingido no máximo em k iterações.

Para provar a segunda afirmação do teorema, i.e., o caso geral onde pode incluir execuções do Passo **(3)(b)**, é suficiente demonstrar que existe um índice $p < 2k$ tal que $|A(p)| = k = |X|$. Para isto, mostraremos que $|A(t)| \geq K(t)$ para todo $t \geq 0$, onde

$$K(t) = \begin{cases} 0, & \text{para } t < k, \\ t - (k - 1), & \text{para } k \leq t < 2k - 1, \\ k, & \text{se } t \geq 2k - 1. \end{cases} \quad (4.30)$$

Seja $Q(t)$ o número de iterações que o Passo **(2)** ou o Passo **3 (a)** foi executado até a conclusão da iteração t , onde $0 \leq t \leq 2k-1$, e $0 \leq p_1 < \dots < p_{Q(t)} \leq 2k-1$ as respectivas iterações em que estes passos foram executados. Pelos mesmos argumentos do primeiro caso, temos que para cada $i = 1, \dots, Q(t)$ ao menos um elemento é adicionado à $A(p_i)$. Como $|A(0)| = 0$, segue que $|A(t)| \geq Q(t)$. Seja t tal que $0 \leq t \leq 2k-1$, vamos provar a relação a seguir:

$$|A(t)| \geq Q(t) \geq K(t). \quad (4.31)$$

Para $0 \leq t < k$, a desigualdade acima vale, pois $K(t) = 0$. Resta mostrar

que $Q(t) \geq t - (k - 1)$ ou, equivalentemente, que $t - Q(t) \leq k - 1$, para $k \leq t \leq 2k - 1$. Observe que até a iteração t o algoritmo invocou $t - Q(t)$ vezes o Passo (3)(b), e em cada vez o conjunto \bar{X} em questão foi particionado em no mínimo dois subconjuntos. Uma vez que o conjunto de padrões de treinamento X possui apenas k elementos, no máximo $k - 1$ partições como as realizadas no Passo (3)(b) podem ocorrer. Logo, $t - Q(t) \leq k - 1$. \square

O Teorema 9 revela que para problemas de classificação binários o número de iterações cresce linearmente com o número de padrões de treinamento. Além disso, o esforço computacional para o cálculo de ínfimos e supremos, escala linearmente com a dimensão dos dados. Portanto, num problema de classificação binário, mesmo uma grande quantidade de dados de alta dimensão não representa um sério desafio computacional ao algoritmo.

O próximo resultado a ser mostrado diz respeito ao erro produzido no conjunto de treinamento de um problema de classificação binário por um MP/CL treinado pelo algoritmo descrito na subseção anterior. Para isto, vamos enunciar dois corolários úteis.

Corolário 5. *Seja $\bar{X} \in \mathcal{C}(t)$, para $t \geq 0$. Não existe $x \in X \setminus \bar{X}$ tal que $x \in \text{box}(\bar{X})$, i.e., $[X \setminus \bar{X}] \cap \text{box}(\bar{X}) = \emptyset$.*

Demonstração. Pelo Lema 9 temos que $X = [\bigcup \mathcal{C}(t)] \cup A(t)$ e $[\bigcup \mathcal{C}(t)] \cap A(t) = \emptyset$. Por um lado, o Lema 6 nos fornece que $[\bigcup_{X' \in \mathcal{C}(t)} \text{box}(X')] \cap [\bigcup [\mathcal{F}^1(t) \cup \mathcal{F}^2(t)]] = \emptyset$ e, como $A(t) \subseteq \bigcup [\mathcal{F}^1(t) \cup \mathcal{F}^2(t)]$ e $\bar{X} \in \mathcal{C}(t)$, concluímos que $A(t) \cap \text{box}(\bar{X}) = \emptyset$. Por outro lado, o Lema 5 garante que $[\bigcup \mathcal{C}(t) \setminus \bar{X}] \cap \text{box}(\bar{X}) = \emptyset$, pois, $[\bigcup_{X' \in \mathcal{C}(t) \setminus \bar{X}} \text{box}(X')] \cap \text{box}(\bar{X}) = \emptyset$. \square

Corolário 6. *Sejam $s, r \in \{1, 2\}$ tal que $s \neq r$. Para todo $t \geq 0$, se $B \in \mathcal{F}^s(t)$ então $C^r \cap B = \emptyset$.*

Demonstração. A prova deste corolário é dada por indução. Para $t = 0$ temos $\mathcal{F}^1(0) = \mathcal{F}^2(0) = \emptyset$. Logo não há nada o que mostrar.

Suponha que a afirmação do corolário vale para $t = p$. Se $\mathcal{C}(p) = \emptyset$ então, novamente, nada resta o que mostrar. Caso contrário, a iteração $p+1$ é executada e um $\bar{X} \in \mathcal{C}(t)$ arbitrário é selecionado. Seja $\bar{\mathcal{F}}^s(p+1) = \mathcal{F}^s(p+1) \setminus \mathcal{F}^s(p)$, para $s = 1, 2$, temos que $\mathcal{F}^s(p+1) = \mathcal{F}^s(p) \cup \bar{\mathcal{F}}^s(p+1)$. Considere $B \in \mathcal{F}^s(p+1)$. Se $B \in \mathcal{F}^s(p)$ então, pela hipótese de indução, temos que $C^r \cap B = \emptyset$ para $r \neq s$.

No caso onde $B \in \bar{\mathcal{F}}^s(\mathbf{p} + 1)$, pelo Corolário 5 segue que $[C^r \setminus \bar{X}] \cap B = \emptyset$ pois $B \subseteq \text{box}(\bar{X})$. Para completar a demonstração basta verificar $B \cap \bar{C}^r = \emptyset$ (lembre que $\bar{C}^r = C^r \cap \bar{X}$). Para isto, uma verificação mais detalhada do algoritmo torna-se necessária.

Se o Passo (2) é executado, então, a condição $\text{box}(\bar{C}^s) \cap \text{box}(\bar{C}^r) = \emptyset$ é satisfeita e $\bar{\mathcal{F}}^s(\mathbf{p} + 1) = \text{box}(\bar{C}^s) = [\mathbf{a}^s, \mathbf{b}^s]$, o que implica que $\text{box}(\bar{C}^s) \cap \bar{C}^r = \emptyset$, pois, $\bar{C}^r \subseteq \text{box}(\bar{C}^r)$.

Se o Passo (3)(a) é executado, então, $\bar{\mathcal{F}}^s(\mathbf{p} + 1) = [\bigcup_{i=1}^n \{\text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)\}] \cup [\bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}]$ e $[\mathbf{a}^s, \mathbf{b}^s] \neq [\mathbf{a}^r, \mathbf{b}^r]$. Suponha, sem perda de generalidade, que $B = \text{box}(H_i^-(\mathbf{a}) \cap \bar{C}^s)$ para algum índice i tal que $B \neq \emptyset$. Logo existe $\mathbf{x} \in \bar{C}^s$ tal que $\mathbf{x} \in H_i^-(\mathbf{a})$ e $\mathbf{a}_i^s \leq x_i < \mathbf{a}_i$, o que implica que $\mathbf{a}_i = \mathbf{a}_i^s \vee \mathbf{a}_i^r = \mathbf{a}_i^r$. Portanto, $\bar{C}^r \cap H_i^-(\mathbf{a}) = \emptyset$, pois, para todo $\mathbf{y} \in \bar{C}^r$ temos que $y_i \geq \mathbf{a}_i$. Dessa maneira concluímos que $\bar{C}^r \cap B = \emptyset$. A demonstração para $B \in \bigcup_{i=1}^n \{\text{box}(H_i^+(\mathbf{b}) \cap \bar{C}^s)\}$ tal que $B \neq \emptyset$ se dá de maneira análoga.

Se o Passo (3)(b) é invocado, então, $\bar{\mathcal{F}}^s(\mathbf{p} + 1) = \{\{\mathbf{u}\} : \mathbf{u} \cap \bar{C}^s\}$. Como $C^s \cap C^r = \emptyset$ por suposição, segue $B \cap C^r = \emptyset$ para todo $B \in \bar{\mathcal{F}}^s(\mathbf{p} + 1)$.

□

O próximo teorema mostra que o algoritmo de treinamento do MP/CL para problemas de classificação binário após convergir, produz perfeita classificação em todo conjunto de treinamento, i.e., não há erro de treinamento. Recorde que para uma extensão de grupo totalmente ordenada \mathbb{L} , construímos dois módulos como na Figura 4.2 e uma arquitetura compreendendo estes dois módulos como na Figura 4.3. Portanto, para todo padrão de treinamento da s -ésima classe $\mathbf{x} \in X \cap C^s$ a arquitetura resultante produz $\arg \max_{s=1,2} y^s = s$, para $s = 1, 2$.

Teorema 10. *Seja $s \in \{1, 2\}$. Todo padrão de treinamento $\mathbf{x} \in X \cap C^s$ está contido em alguma hipercaixa $B \in \mathcal{F}^s$. Além disso, \mathbf{x} não está contido em nenhuma hipercaixa $B' \in \mathcal{F}^r$ tal que $s \neq r \in \{1, 2\}$.*

Demonstração. Uma vez que o algoritmo converge em um número finito de iterações pelo Teorema 9, existe um índice \mathbf{p} tal que $X = A(\mathbf{p}) = \{\mathbf{x} \in X : \mathbf{x} \in \bigcup [\mathcal{F}^1(\mathbf{p}) \cup \mathcal{F}^2(\mathbf{p})]\}$ pelo Lema 9. Para $s = 1, 2$, o Corolário 6 fornece que $C^s \cap [\bigcup \mathcal{F}^r(\mathbf{p})] = \emptyset$ para $s \neq r$, o que implica que $C^s \subseteq [\bigcup \mathcal{F}^s(\mathbf{p})]$, i.e., para todo $\mathbf{x} \in X \cap C^s$ existe $B \in \mathcal{F}^s$ tal que $\mathbf{x} \in B$. □

Como resultado do Teorema 10, em um problema de classificação binário o MP/CL resultante produz $\arg \max_{s=1,2} y^s = r$ para toda entrada $\mathbf{x} \in C^r$, para $r = 1, 2$, de acordo com a Equação 2.28.

4.3.3 Algoritmo do MP/CL para problemas de classificação com múltiplas classes

Suponha que nos deparemos com um problema de classificação com S classes, onde $S > 2$, e que nosso objetivo seja construir um MP/CL para solucioná-lo. Seja $C^s \subseteq X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$ o conjunto de padrões de treinamento em \mathbb{L}^n pertencentes à s^{a} classe para todo $s = 1, \dots, S$, tais que, $C^s \cap C^r = \emptyset$ para $s \neq r$ e $\bigcup_{s=1}^S C^s = X$. A estratégia adotada consiste em aplicar o algoritmo do MP/CL para cada par (C^s, C^r) com $s \neq r$. Dessa maneira, obtemos $S - 1$ famílias de hipercaixas para cada C^s , onde $s = 1, \dots, S$. Tomando todas as intersecções das hipercaixas de famílias distintas associadas com a mesma C^s , produzimos uma única família de hipercaixas \mathcal{F}^s para cada uma das S classes, que podem ser usadas para construir um único MP/CL com arquitetura similar à ilustrada na Figura 4.3 com S módulos, onde cada módulo possui a forma exibida na Figura 4.2. Formalmente, seja \mathcal{F}_r^s o símbolo que denota a família de hipercaixas associadas à s -ésima classe gerada pelo algoritmo do MP/CL considerando apenas os padrões de treinamento das classes C^s e C^r . Uma única família de hipercaixas \mathcal{F}^s é produzida pela seguinte equação

$$\mathcal{F}^s = \left\{ \bigcap_{r \neq s} [\mathbf{a}^r, \mathbf{b}^r] : [\mathbf{a}^r, \mathbf{b}^r] \in \mathcal{F}_r^s \right\} \text{ para } s = 1, \dots, S. \quad (4.32)$$

Note que o algoritmo dado acima para MP/CLs, para múltiplas classes consiste de uma extensão natural do algoritmo binário. Além disso, o algoritmo estendido herda as propriedades provadas para o caso binário, com exceção da complexidade computacional para o pior caso que cresce exponencialmente com o número de classes. Esta última afirmação decorre diretamente da observação de que dado $s \in \{1, \dots, S\}$, para todo $r \neq s$ o conjunto \mathcal{F}_r^s pode conter no máximo k hipercaixas, onde k é o número de padrões. Logo, a Equação 4.32 pode exigir no máximo k^{S-1} cálculos.

Apesar do processo de aplicar sucessivas intersecções para formar uma única família de hipercaixas ser extremamente custoso, a família resultante exibe as

mesmas propriedades demonstradas na subseção anterior para o caso binário. Isto pode ser confirmado através de verificações simples, cuja idéia central vem de que um elemento $B = \bigcap_{r \neq s} [\mathbf{a}^r, \mathbf{b}^r] \in \mathcal{F}^s$, preserva simultaneamente propriedades oriundas de cada $[\mathbf{a}^r, \mathbf{b}^r] \in \mathcal{F}_r^s$ (tudo graças às interseções).

Para sumarizar, o algoritmo de treinamento do MP/CL para duas classes ou mais exibem as seguintes propriedades [91]:

1. Algoritmo construtivo que gera automaticamente nós ocultos correspondentes a hipercaixas em \mathbb{L}^n ;
2. Garantia de parada do algoritmo de treinamento;
3. Complexidade $\mathcal{O}(\mathbf{n}^{S-1})$ no pior caso, onde S é o número de classes;
4. Perfeita classificação nos dados de treinamento;
5. Nenhuma sobreposição de hipercaixas com rótulo distintos;
6. Independência das escolhas dos conjuntos \bar{X} durante o algoritmo, em particular independência da sequência dos dados de treinamento.

4.4 Experimentos com redes neurais morfológicas construtivas

Nesta seção avaliamos o desempenho de classificação e o esforço computacional do modelo MP/CL em uma série de experimentos em quatro base de dados bem conhecidas: o problema sintético de Ripley [67, 68], a base de dados de segmentação de imagens, a base de dados de íris e a base de dados de câncer de mama de Wisconsin [3]. Além disso, comparamos os resultados produzidos pelo MP/CL com os produzidos pelo perceptron de múltiplas camadas (MLP), FLNN [63], classificador FLR [40], k vizinhos mais próximos (kNN) [24] e árvore de decisão [10, 26].

Em todos os experimentos, consideramos MLPs com funções de ativação sigmoid e com uma única camada escondida. As MLPs foram implementadas utilizando o toolbox de redes neurais do MATLAB [59], e treinadas utilizando retropropagação baseado no método do gradiente com momento e passo adaptativo com taxa de aprendizagem $\eta = 10^{-4}$ e constante de momento $\alpha = 0.9$. Em cada

problema individual, aplicamos validação cruzada com 10-fold para determinar o número de neurônios escondidos $H_a \in \{10, 20, 30\}$, o parâmetro de crescimento $\rho_{\text{inc}} \in \{1.0, 1.1, 1.2\}$ e o parâmetro de decrescimento $\rho_{\text{dec}} \in \{0.5, 0.6, 0.7\}$. A camada de saída de cada MLP aplicada a um dos problemas de classificação subsequentes compreende S neurônios de saídas, onde S representa o número de classes. Na fase de teste, associamos o padrão de entrada \mathbf{x} com a classe correspondente ao neurônio de saída mais ativo. Uma vez que os resultados da MLP depende da inicialização dos pesos, escolhemos os pesos que produziram o menor erro quadrático médio no treinamento em dez inicializações aleatorias. Finalmente, o algoritmo de treinamento foi interrompido quando o número de iterações atingiu 25000, ou o erro médio quadrático (MSE) no conjunto de treinamento atingiu um valor menor do que 0.005 ou variou em menos de 0.0002 da última iteração. Durante a aplicação do método de validação cruzada, também foi utilizado como critério de parada a quantidade de vezes consecutivas que o MSE no conjunto de validação aumentou, enquanto o MSE no conjunto de treinamento mantinha-se decrescendo.

Tanto a FLNN quanto o classificador FLR produzem resultados ligeiramente diferentes dependendo da ordem com a qual os padrões são apresentados a rede durante o treinamento. Nos experimentos conduzidos nesta seção optamos por manter a mesma ordem com que os padrões de treinamento aparecem nas bases de dados na internet.

Ao contrário dos outros modelos, o MP/CL é equipado com algoritmo de treinamento que exhibe independência da sequência dos padrões de treinamento e não requer inicializações de pesos. Além disso, assim como a FLNN, o MP/CL converge num número finitos de iterações atingindo, no final, erro de treinamento zero.

Em nossas simulações com os classificadores FLNs, construímos FLNNs e classificadores FLRs utilizando a medida de inclusão baseada na Equação 4.3 sob o produto de reticulados $\mathbb{L}_1 \times \dots \times \mathbb{L}_n$ com constituintes $\mathbb{L}_i = [0, 1]$. A medida de inclusão induzida pela função de avaliação positiva e automorfismo dual em \mathbb{R} da Equação 4.4, também foi empregada em outras FLRs testadas. Para simplificar, utilizaremos o símbolo FLR_l para denotar o classificador FLR equipado com a medida de inclusão baseada na função de avaliação positiva ν_l e no automorfismo dual θ_l . Similarmente, o símbolo FLR_e^ζ denota o classificador FLR equipado com a medida de inclusão baseada em ν_e^ζ e θ_e .

O projeto de uma FLR requer a escolha de alguns parâmetros, por exemplo, o parâmetro de vigilância ρ_{crit} utilizado no algoritmo de treinamento. Em nossos experimentos, aplicamos técnicas de validação cruzada com 10-folds para determinar tais parâmetros. Especificamente, para o modelo θ_l selecionamos o valor de $\rho_{\text{crit}} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ que produziu menor taxa de erro médio nos 10-folds. Com respeito ao modelo FLR_e^ζ , escolhemos os valores de $\zeta \in \{1, 5, 10, 15\}$ e $\rho_{\text{crit}} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ que produziram menor taxa de erro médio nos 10-folds.

Obviamente, o modelo kNN depende da escolha do parâmetro k . De maneira similar ao caso das FLRs, empregamos validação cruzada com 10-folds para determinar o melhor valor de $k \in \{1, 2, \dots, 15\}$ em termos da menor taxa de erro médio no conjunto de validação. Finalmente, incluímos em nossas comparações os resultados obtidos por árvores de decisão, especificamente, utilizamos a implementação padrão das árvores de classificação do MATLAB cujo critério da escolha de uma divisão é dada pelo índice de diversidade Gini [59].

As tabelas de 1 à 4 exibem a porcentagem de padrões mal classificados no treinamento e teste. Uma vez que o tipo de operações executadas por modelos individuais nas épocas de treinamento variam bastante de um modelo para o outro, optamos por adicionar em nossas comparações o tempo de CPU requerido para treinamento, validação e teste (sobre um AMD Athlon 64 X2 Dual Core Processor 4200+ com velocidade de processamento de 2.221 GHz) para cada modelo individual. Note que, o tempo utilizado para procedimentos de validação cruzada também foi considerado.

Todos os modelos e algoritmos foram implementados usando MATLAB. Nossas comparações de esforço computacional incluem também o número de neurônios. Nos modelos morfológicos construtivos, consideramos uma hipercaixa gerada na fase de treinamento como uma unidade de processamento, já que a expressão $\varepsilon_v(\mathbf{x}) \wedge \bar{\delta}_w(\mathbf{x})$ pode ser computada como uma única expressão, conforme explicado na Seção 4.2.

A Tabela 4.1 sumariza as características dos quatros problemas de classificação considerado neste trabalho.

Tabela 4.1: Características das quatro bases de dados utilizadas.

Base de Dados Sintético de Ripley			
<i>Atributos</i>	<i>Classes</i>	<i>Padrões Trein.</i>	<i>Padrões Teste</i>
19	7	30 por classe	300 por classe
Problema de Segmentação de Imagem			
<i>Atributos</i>	<i>Classes</i>	<i>Padrões Trein.</i>	<i>Padrões Teste</i>
19	7	30 por classe	300 por classe
Problema de Classificação de Iris			
<i>Atributos</i>	<i>Classes</i>	<i>Padrões Trein.</i>	<i>Padrões Teste</i>
4	3	35 por classe	15 por classe
Problema de Diagnóstico de Câncer de Mama			
<i>Atributos</i>	<i>Classes</i>	<i>Padrões Trein.</i>	<i>Padrões Teste</i>
30	2	249 benignos e 212 malignos	108 benignos e 64 malignos

4.4.1 Problema sintético de Ripley

O conjunto de dados sintético de Ripley consiste em exemplos de dados de duas classes [67, 68], onde cada padrão possui dois atributos, ou seja, consiste de um problema de classificação binário no \mathbb{R}^2 . Os dados são divididos em um conjunto de treinamento e um de teste consistidos, respectivamente, de 250 e 1000 padrões com o mesmo número de exemplares pertencentes a cada uma das duas classes.

A Tabela 4.2 revela que o kNN com $k = 13$, escolhido usando validação cruzada com 10-fold, exibiu o melhor resultado de classificação, porém, lembre que todos os 250 padrões de treinamento precisaram ser armazenados. Um desempenho de classificação similar foi atingido pelo MP/CL que gerou automaticamente 55 neurônios escondidos simples. O algoritmo do MP/CL convergiu muito rapidamente para o conjunto de pesos que separa perfeitamente os dados de treinamento. Assim como o MP/CL, a FLNN também não produziu erros no treinamento.

Em contraste com os demais modelos, o MP/CL não requer a escolha de nenhum parâmetro, o que justifica grandes diferenças entre os valores do MP/CL

com os de outros modelos na coluna $T_{\text{cpu}}^{\text{lt}}$ da Tabela 4.2. Por exemplo, a MLP que obteve um tempo de processamento total $T_{\text{cpu}}^{\text{lt}}$ por volta de 6 horas, onde muito deste custo excessivo de CPU decorre de aplicar a validação cruzada com 10-fold para determinar os seguintes parâmetros adequados: número de neurônios escondidos $H_a = 20$, parâmetro de crescimento $\rho_{\text{inc}} = 1.2$ e parâmetro de decrescimento $\rho_{\text{dec}} = 0.6$. Com respeito aos modelos FLR_l e FLR_e^ζ , a validação cruzada resultou em $\rho_{\text{crit}} = 0.9$ em ambos os casos e $\zeta = 10$.

Os resultados produzidos pela árvore de decisão, MLP e pela FLNN são comparáveis mas o tempo de CPU para treinamento da FLNN e especialmente da MLP foram relativamente longos comparados ao da árvore de decisão. Ambas a FLNN e os classificadores FLRs produziram áreas de indecisão correspondentes a sobreposições de hipercaixas, porém, isso poderia ter sido evitado, adicionando computações adicionais muito demoradas. Em nossos experimentos, nenhuma decisão com respeito às áreas de indecisão foi considerada. Uma visualização das áreas de indecisão geradas pelo modelo FLNN no problema sintético de Ripley pode ser encontrada em [88, 89]. A superfície de decisão gerada pelo modelo MP/CL é visualizada na Figura 4.9.

Tabela 4.2: Percentual de padrões mal classificados no treinamento (E_{tr}) e teste (E_{te}) nos experimentos, tempo de CPU em segundos para treinamento (T_{cpu}^l), total de tempo de CPU em segundos para treinamento, validação e teste ($T_{\text{cpu}}^{\text{lt}}$), e o número de neurônios ou hipercaixas (H_n).

<i>Modelo</i>	Base de Dados Sintético de Ripley				
	$E_{\text{tr}}(\%)$	$E_{\text{te}}(\%)$	T_{cpu}^l	$T_{\text{cpu}}^{\text{lt}}$	H_n
MP/CL	0.0	10.20	0.05	0.16	55
FLR_l ($\rho_{\text{crit}} = 0.9$)	8.80	15.30	0.41	33.05	77
FLR_e^{10} ($\rho_{\text{crit}} = 0.9$)	10.80	16.20	0.58	113.45	51
FLNN	0.0	14.20	84.43	85.65	46
MLP	4.0	14.60	57.10	21120.77	22
13NN	11.60	9.60	—	0.16	250
Árvore de Decisão	5.20	13.00	0.37	0.43	41

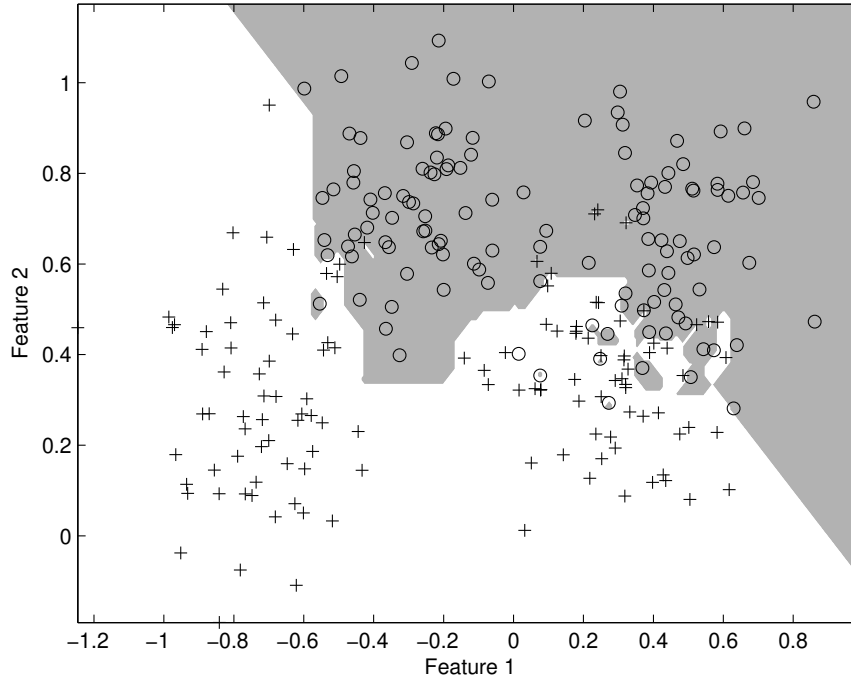


Figura 4.9: A diferença de tons da imagem corresponde à superfície de decisão produzida pelo modelo MP/CL no problema sintético de Ripley.

4.4.2 Problema de segmentação de imagem

O conjunto de dados de segmentação de imagem foi doada pelo “*Vision Group*” da Universidade de Massachussets e está incluído no “*Machine Learning Repository*” da Universidade da Califórnia, Irvine [3]. Este conjunto de dados consiste em 210 padrões de treinamento e 2100 padrões de teste, divididos entre 7 diferentes classes, onde cada padrão é descrito por 19 atributos contínuos.

Por um lado, a Tabela 4.3 mostra que a MLP superou todos os outros modelos em termos de precisão de classificação. Por outro lado, o tempo de treinamento T_{cpu}^l é relativamente longo e a busca pelos parâmetros adequados ($H_a = 30$, $\rho_{\text{inc}} = 1.2$, $\rho_{\text{dec}} = 0.7$) para a MLP usando validação cruzada com 10-fold produziu um tempo total de processamento $T_{\text{cpu}}^{\text{lv}}$ excessivo de quase sete horas. Além disso, a Tabela 4.3 revela que os classificadores FLRs, a árvore de decisão e o MP/CL, também produziram resultados satisfatórios, porém, o modelo FLR_l gerou um número excessivo de 208 hipercaixas a partir de 210 padrões de treinamento. A FLR_e^ζ , FLNN, árvore de decisão e MP/CL exibiram taxas de erros similares em

torno de 10%. Devido à simplicidade computacional dos neurônios do MP/CL, o número relativamente alto de 63 neurônios escondidos, construídos pelo algoritmo de treinamento do MP/CL, não representa uma desvantagem quando comparado aos 17 neurônios da FLNN e aos 37 neurônios da MLP, pois, envolvem cálculos computacionalmente caros. Em contraste, a árvore de decisão exigiu muito pouco tempo de CPU para treinamento e teste. Finalmente, o pior desempenho de classificação é dado pelo modelo kNN.

Tabela 4.3: Percentual de padrões mal classificados no treinamento (E_{tr}) e teste (E_{te}) nos experimentos, tempo de CPU em segundos para treinamento (T_{cpu}^l), total de tempo de CPU em segundos para treinamento, validação e teste (T_{cpu}^{lvt}), e o número de neurônios ou hipercaixas (H_n).

<i>Modelo</i>	Base de Dados de Segmentação de Imagem				
	$E_{tr}(\%)$	$E_{te}(\%)$	T_{cpu}^l	T_{cpu}^{lvt}	H_n
MP/CL	0.0	10.14	9.08	10.69	63
FLR _l ($\rho_{crit} = 0.9$)	0.0	8.00	0.27	151.07	208
FLR _e ⁵ ($\rho_{crit} = 0.9$)	0.48	9.62	0.15	101.39	48
FLNN	0.0	10.00	25.98	27.02	17
MLP	0.95	6.38	100.73	24627.53	37
1NN	0.0	19.52	—	11.10	210
Árvore de Decisão	2.38	9.71	0.05	0.08	27

4.4.3 Problema de classificação de flores íris

A base de dados de flores íris é um conjunto de dados multivalorados, introduzido por R.A. Fisher [27]. A base consiste em 50 exemplares de três tipos de espécies de flores íris: setosa, virginica e versicolor. Quatro propriedades foram medidas a partir de cada exemplo, que são: altura, largura, sépala e pétala.

Nesta base de dados, os conjuntos de treinamento e teste não são dados explicitamente como no problema sintético de Ripley e no problema de segmentação de imagem. Além disso, o número de exemplares presentes é relativamente pequeno se comparado aos demais problemas.

Conduzimos nossos experimentos de duas maneiras. Na primeira, dividimos o conjunto original [3] num conjunto de treinamento, consistindo dos primeiros 35 padrões de cada espécie, e num conjunto de teste, consistindo dos padrões

restantes. Em seguida, os utilizamos para treinar e testar cada um dos modelos. Na Tabela 4.4 apresentamos os resultados obtidos neste experimento. Note que os modelos MP/CL e FLNN classificaram perfeitamente todos os padrões tanto de treinamento quanto de teste. A árvore de decisão, MLP ($H_a = 10$, $\rho_{\text{inc}} = 1.2$, $\rho_{\text{dec}} = 0.6$) e kNN também atingiram perfeita classificação no conjunto de teste apesar dos respectivos erros de treinamento serem ligeiramente positivos.

Neste experimento, decididamente, o MP/CL e a árvore de decisão tiveram o melhor desempenho por causa que seus algoritmos de treinamento não apenas convergiram muito rapidamente mas, também construíram uma arquitetura final com um número pequeno de neurônios.

Tabela 4.4: Percentual de padrões mal classificados no treinamento (E_{tr}) e teste (E_{te}) nos experimentos, tempo de CPU em segundos para treinamento ($T_{\text{cpu}}^{\text{l}}$), total de tempo de CPU em segundos para treinamento, validação e teste ($T_{\text{cpu}}^{\text{lv}}$), e o número de neurônios ou hipercaixas (H_n).

<i>Modelo</i>	Base de Dados de Iris				
	$E_{\text{tr}}(\%)$	$E_{\text{te}}(\%)$	$T_{\text{cpu}}^{\text{l}}$	$T_{\text{cpu}}^{\text{lv}}$	H_n
MP/CL	0.0	0.0	0.08	0.13	15
FLR _l ($\rho_{\text{crit}} = 0.7$)	2.86	2.22	0.04	13.21	29
FLR _e ¹⁰ ($\rho_{\text{crit}} = 0.8$)	0.95	4.44	0.06	31.75	15
FLNN	0.0	0.0	5.63	5.70	10
MLP	0.95	0.0	46.76	16274.97	13
5NN	3.81	0.0	—	1.14	105
Árvore de Decisão	2.86	0.0	0.01	0.01	9

A segunda maneira consiste em dividir aleatoriamente 35 exemplares de cada classe de flores para treinamento e os demais para teste, por 100 vezes. Em cada divisão produzida, computamos o desempenho de cada modelo, treinado com os parâmetros iguais aos dos modelos da Tabela 4.4, nos dados de treinamento e teste. A Tabela 4.5 exibe os valores médios do percentual de acerto no conjunto de treinamento e teste, dos tempos requeridos de CPU e da quantidade de neurônios utilizados para cada modelo testado.

Note na Tabela 4.5, que o modelo kNN obteve o menor percentual médio de erro de classificação no conjunto de teste, porém isso não acontece no conjunto de treinamento e todos os 105 padrões de treinamento necessitam ser armazenados. O segundo melhor desempenho no conjunto de teste, ligeiramente maior que o

Tabela 4.5: Percentual médio de padrões mal classificados no treinamento (E_{tr}) e teste (E_{te}), tempo médio de CPU em segundos para treinamento (T_{cpu}^l), total de tempo médio de CPU em segundos para treinamento, validação e teste (T_{cpu}^{lvt}), e o número médio de neurônios ou hipercaixas (H_n).

<i>Modelo</i>	Base de Dados de Iris				
	$E_{tr}(\%)$	$E_{te}(\%)$	T_{cpu}^l	T_{cpu}^{lvt}	H_n
MP/CL	0.0	4.56	0.07	0.13	10.14
FLR _l ($\rho_{crit} = 0.7$)	1.27	5.04	0.04	0.27	26.92
FLR _e ¹⁰ ($\rho_{crit} = 0.8$)	1.51	5.47	0.06	0.33	15.89
FLNN	0.0	6.24	5.71	5.72	8.06
MLP	1.29	4.56	13.66	13.67	13
5NN	2.47	3.38	—	0.11	105
Árvore de Decisão	1.99	5.67	0.01	0.01	6.92

primeiro, ficam por conta do MP/CL e da MLP, mas diferente da MLP, o MP/CL não produz erro no conjunto de treinamento e não requer a escolha de nenhum parâmetro, além de exigir um tempo de CPU bem inferior ao da MLP.

4.4.4 Problema de câncer de mama de Wisconsin

O conjunto de dados de câncer de mama de Wisconsin foi criado por Wolberg, Street e Mangasarian da Universidade de Wisconsin [100]. Trinta propriedades com valores reais foram computadas a partir da digitalização de biópsias de câncer de mama, que descrevem características presentes no núcleo da célula presente na imagem. O conjunto de dados consiste em 569 instâncias, onde 357 são benignos e 212 malignos. Alocamos os primeiros 249 padrões benignos e os primeiros 148 padrões malignos para o conjunto de treinamento, e o restante aplicamos como conjunto de teste.

A Tabela 4.6 indica que neste experimento, os modelos FLR_e^ζ e MP/CL tiveram excelentes desempenhos. Apesar do modelo FLR_e^ζ produzir uma precisão de classificação ligeiramente melhor, o algoritmo do MP/CL requereu menos tempo para convergir e gerou menos neurônios do que o algoritmo construtivo do modelo FLR_e^ζ.

Finalmente, repare que o tempo total T_{cpu}^{lvt} requerido pela MLP foi muito alto (em torno de 11 horas). O método de validação cruzada produziu $H_a = 10$,

$\rho_{\text{inc}} = 1.2$, e $\rho_{\text{dec}} = 0.7$.

Tabela 4.6: Percentual de padrões mal classificados no treinamento (E_{tr}) e teste (E_{te}) nos experimentos, tempo de CPU em segundos para treinamento ($T_{\text{cpu}}^{\text{l}}$), total de tempo de CPU em segundos para treinamento, validação e teste ($T_{\text{cpu}}^{\text{lt}}$), e o número de neurônios ou hipercaixas (H_n).

<i>Modelo</i>	Problema de Câncer de Mama de Wisconsin				
	$E_{\text{tr}}(\%)$	$E_{\text{te}}(\%)$	$T_{\text{cpu}}^{\text{l}}$	$T_{\text{cpu}}^{\text{lt}}$	H_n
MP/CL	0.0	4.07	0.08	0.15	64
FLR_{l} ($\rho_{\text{crit}} = 0.6$)	0.0	9.30	3.53	555.06	376
$\text{FLR}_{\text{e}}^{10}$ ($\rho_{\text{crit}} = 0.9$)	0.0	3.49	2.30	404.41	133
FLNN	0.0	6.98	355.64	355.83	37
MLP	3.27	5.81	109.49	68981.24	12
10NN	5.79	5.81	—	14.73	397
Árvore de Decisão	0.5	8.14	0.08	0.13	23

Conclusões e considerações finais

Nos primeiros capítulos deste trabalho, discutimos a arquitetura teórica de reticulados das redes neurais morfológicas. Em particular, distinguimos entre definições de operações elementares da morfologia matemática no sentido algébrico do sentido intuitivo. Isso permitiu definir redes neurais morfológicas também no sentido intuitivo.

A memória associativa morfológica baseada na medida de subsethood de Kosko (KS-FAM) mostrou-se ser uma alternativa promissora na área de reconhecimento de padrões. Em experimentos de reconstrução de imagens envolvendo imagens em tons de cinza, a KS-FAM apresentou tolerância com respeito a vários tipos de ruídos: pimenta, sal e pimenta, e gaussiano. Além disso, o modelo KS-FAM obteve também excelente desempenho em experimentos envolvendo padrões incompletos, variações de luminosidade e variações de orientação dos padrões originais.

Neste trabalho, foi demonstrado que para um conjunto de memórias fundamentais que satisfaz uma única hipótese fraca ($\mathbf{x}^\xi \not\preceq \mathbf{x}^\gamma$ para todo $\xi \neq \gamma$), então a KS-FAM exibe capacidade de armazenamento absoluta. Também foram caracterizadas as saídas produzidas por uma KS-FAM em termos de máximos entre memórias fundamentais. Todavia, a fase de recordação desse modelo foi pouco explorada, e nenhuma caracterização adequada sobre as condições necessárias e suficientes para se obter perfeito reconhecimento foi elaborada ainda. Por fim, classificamos a KS-FAM como um modelo de rede neural morfológica no sentido intuitivo, uma vez que os neurônios da primeira camada executam erosões apenas

no sentido intuitivo.

Com respeito às redes neurais construtivas, revimos os modelos FLNs e o perceptron morfológico. Em especial, abordamos com detalhe o modelo MP/CL demonstrando várias de suas propriedades. Além disso, também apontamos a conexão entre o algoritmo de treinamento do MP/CL e a decomposição de mapeamentos entre reticulados completos sugerida por Banon e Barrera [5].

O algoritmo do MP/CL gera automaticamente uma arquitetura feedforward com neurônios escondidos correspondendo a hipercaixas em um reticulado completo $\bar{\mathbb{L}}^n$, onde \mathbb{L} corresponde a um reticulado totalmente ordenado que possui uma estrutura adicional de grupo. Vimos também que as funções de agregações empregadas nos neurônios do MP/CL, assim como seus predecessores MP e MP/C, representam operações morfológicas elementares entre reticulados completos.

As principais características do algoritmo construtivo do MP/CL são listadas a seguir [91]:

1. Garantia de parada do algoritmo de treinamento;
2. Complexidade $\mathcal{O}(n^{S-1})$ no pior caso, onde S é o número de classes;
3. Perfeita classificação dos dados de treinamento;
4. Nenhuma área de indecisão é produzida, isto é, não há sobreposição de hipercaixas com rótulo distintos;
5. Independência da ordem com que os padrões de treinamento são apresentados à rede.

Neste trabalho, investigamos o desempenho e o esforço computacional do modelo MP/CL em problemas de classificação bem conhecidos [3, 67]. O MP/CL exibiu resultados de classificação satisfatórios, além de convergir rapidamente na fase de treinamento, sem nenhum ajuste de parâmetro ou inicialização de pesos. Entre os modelos testados, um desempenho similar foi encontrado pelo classificador FLR, que necessita que alguns parâmetros sejam determinados de antemão. Apesar do MP/CL produzir um número de neurônios escondidos relativamente alto comparado ao da MLP, ressaltamos que as operações de produto min executadas nos nós escondidos do MP/CL envolve um esforço computacional muito pequeno, especialmente se implementado em hardware [65].

Os testes estatísticos sugeridos por Demšar [22] para comparar vários classificadores sobre várias bases de dados, em particular, o teste de Friedman com comparações post-hoc, não pôde ser aplicado devido ao número relativamente baixo de base de dados testadas. Em outras palavras, as informações de desempenho obtidas dos classificadores nas quatro bases de dados não são suficientes para se obter uma análise conclusiva utilizando tais testes.

Referências Bibliográficas

- [1] ARAÚJO, R., MADEIRO, F., AND PESSOA, L. Modular morphological neural network training via adaptive genetic algorithm for design translation invariant operators. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (Toulouse, France, 2006).
- [2] ARAÚJO, R., MADEIRO, F., SOUSA, R., PESSOA, L., AND FERREIRA, T. An evolutionary morphological approach for financial time series forecasting. In *Proceedings of the IEEE Congress on Evolutionary Computation* (Vancouver, Canada, 2006), pp. 2467–2474.
- [3] ASUNCION, A., AND NEWMAN, D. UCI repository of machine learning databases, 2007.
- [4] BACKHOUSE, R., AND CARRÉ, B. Regular algebra applied to path-finding problems. *J. Inst. Math. Appl.* 15 (1975), 161–186.
- [5] BANON, G., AND BARRERA, J. Decomposition of mappings between complete lattices by mathematical morphology, part 1. general lattices. *Signal Processing* 30, 3 (Feb. 1993), 299–327.
- [6] BELOHLÁVEK, R. Fuzzy logical bidirectional associative memory. *Information Sciences* 128, 1-2 (2000), 91–103.
- [7] BIRKHOFF, G. *Lattice Theory*, 3 ed. American Mathematical Society, Providence, 1993.
- [8] BLOCH, I. Mathematical morphology on bipolar fuzzy sets. In *Proceedings of the International Symposium on Mathematical Morphology* (2007), vol. 2, pp. 3–4.
- [9] BRAGA-NETO, U., AND GOUTSIAS, J. Supremal multiscale signal analysis. *SIAM Journal of Mathematical Analysis* 36, 1 (2004), 94–120.

- [10] BREIMAN, L., FRIEDMAN, J., OLSHEN, R. A., AND STONE, C. J. *Classification and Regression Trees*. Wadsworth International, Belmont, CA, U.S.A, 1984.
- [11] CARPENTER, G., AND GROSSBERG, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing* 37, 1 (Jan. 1987).
- [12] CARPENTER, G., AND GROSSBERG, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. In *Neural Networks: Theoretical Foundations*, C. Lau, Ed. IEEE Press, Piscataway, NJ, 1992.
- [13] CARPENTER, G., GROSSBERG, S., AND ROSEN, D. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks* 4, 6 (1991), 759–771.
- [14] CARRÉ, B. An algebra for network routing problems. *J. Inst. Math. Appl.* 7 (1971), 273–294.
- [15] CHUNG, F., AND LEE, T. On fuzzy associative memory with multiple-rule storage capacity. *IEEE Transactions on Fuzzy Systems* 4, 3 (August 1996), 375–384.
- [16] CORNELIS, C., DER DONCK, C. V., AND KERRE, E. Sinha-dougherty approach to the fuzzification of set inclusion revisited. *Fuzzy Sets and Systems* 134, 2 (March 2003), 283–295.
- [17] COSTANTINI, G., CASALI, D., AND PERFETTI, R. Neural associative memory storing gray-coded gray-scale images. *Neural Networks, IEEE Transactions on* 14, 3 (May 2003), 703–707.
- [18] CUNINGHAME-GREEN, R. *Minimax Algebra: Lecture Notes in Economics and Mathematical Systems* 166. Springer-Verlag, New York, 1979.
- [19] CUNINGHAME-GREEN, R. Minimax algebra and applications. In *Advances in Imaging and Electron Physics*, P. Hawkes, Ed., vol. 90. Academic Press, New York, NY, 1995, pp. 1–121.

- [20] DAVIDSON, J. Foundation and applications of lattice transforms in image processing. In *Advances in Electronics and Electron Physics*, P. Hawkes, Ed., vol. 84. Academic Press, New York, NY, 1992, pp. 61–130.
- [21] DAVIDSON, J., AND HUMMER, F. Morphology neural networks: An introduction with applications. *Circuits, Systems, and Signal Processing* 12, 2 (1993), 177–210.
- [22] DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.
- [23] DENG, T., AND HEIJMANS, H. Grey-scale morphology based on fuzzy logic. *Journal of Mathematical Imaging and Vision* 16, 2 (Mar. 2002), 155–171.
- [24] DEVROYE, L., GYÖRFI, L., AND LUGOSI, G. *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. Springer Verlag, Berlin-Heidelberg, Feb. 1997.
- [25] ESMI, E., AND SUSSNER, P. A fuzzy associative memory based on Kosko’s subethood measure. In *Proceedings of World Congress on Computational Intelligence* (2010). accepted for publication.
- [26] ESPOSITO, F., MALERBA, D., AND SEMERARO, G. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), 476–491.
- [27] FISHER, R. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7 (1936), 179–188.
- [28] GADER, P., WON, Y., AND KHABOU, M. Image algebra network for pattern recognition. In *Image Algebra and Morphological Image Processing V* (July 1994), vol. 2300 of *Proceedings of SPIE*, pp. 157–168.
- [29] GADER, P. D., WON, Y., AND KHABOU, M. Image algebra networks for pattern classification. In *Image Algebra and Morphological Image Processing V* (July 1994), vol. 2300 of *Proceedings of SPIE*, pp. 157–168.

- [30] GRAÑA, M., GALLEGO, J., TORREALDEA, F. J., AND D'ANJOU, A. On the application of associative morphological memories to hyperspectral image analysis. *Lecture Notes in Computer Science 2687* (2003), 567–574.
- [31] GRÄTZER, G. *Lattice Theory: First Concepts and Distributive Lattices*. W. H. Freeman, San Francisco, CA, 1971.
- [32] HARALICK, R., AND SHAPIRO, L. *Computer and Robot Vision*, vol. I. Addison-Wesley, New York, 1992.
- [33] HARALICK, R., STERNBERG, S., AND ZHUANG, X. Image analysis using mathematical morphology: Part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9, 4 (July 1987), 532–550.
- [34] HASSOUN, M. H. Dynamic associative neural memories. In *Associative Neural Memories: Theory and Implementation*, M. H. Hassoun, Ed. Oxford University Press, Oxford, U.K., 1993.
- [35] HASSOUN, M. H. *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge, MA, 1995.
- [36] HEIJMANS, H. *Morphological Image Operators*. Academic Press, New York, NY, 1994.
- [37] HOCAOGLU, A., AND GADER, P. Domain learning using Choquet integral-based morphological shared weight neural networks. *Image and Vision Computing* 21, 7 (2003), 663–673.
- [38] HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79 (Apr. 1982), 2554–2558.
- [39] JUNBO, F., FAN, J., AND YAN, S. A learning rule for fuzzy associative memories. In *Proceedings of the IEEE International Joint Conference on Neural Networks* (June 1994), vol. 7, pp. 4273 – 4277.
- [40] KABURLASOS, V., ATHANASIADIS, I., AND MITKAS, P. Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation. *International Journal of Approximate Reasoning* 45, 1 (May 2007), 152–188.

- [41] KABURLASOS, V., AND PAPADAKIS, S. Granular self-organizing map (gr-som) for structure identification. *Neural Networks* 19, 5 (June 2006), 623–643.
- [42] KABURLASOS, V., AND PAPADAKIS, S. A granular extension of the fuzzy-artmap (fam) neural classifier based on fuzzy lattice reasoning (flr). *Neurocomputing* 72, 10–12 (2009), 2067–2078.
- [43] KABURLASOS, V., AND PETRIDIS, V. Fuzzy lattice neurocomputing (FLN) models. *Neural Networks* 13, 10 (2000), 1145–1170.
- [44] KABURLASOS, V. G. FINs: Lattice theoretic tools for improving prediction of sugar production from populations of measurements. *IEEE Transactions on Systems Man and Cybernetics, Part B - Cybernetics* 34, 2 (2004), 1017–1030.
- [45] KHABOU, M., AND GADER, P. Automatic target detection using entropy optimized shared-weight neural networks. *IEEE Transactions on Neural Networks* 11, 1 (Jan. 2000), 186–193.
- [46] KHABOU, M., GADER, P. D., AND KELLER, J. M. LADAR target detection using morphological shared-weight neural networks. *Machine Vision and Applications* 11, 6 (May 2000), 300–305.
- [47] KIM, C. Segmenting a low-depth-of-field image using morphological filters and region merging. *IEEE Transactions on Image Processing* 14, 10 (2005), 1503–1511.
- [48] KLIR, G. J., AND YUAN, B. *Fuzzy Sets and Fuzzy Logic; Theory and Applications*. Prentice Hall, Upper Saddle River, N. Y., 1995.
- [49] KOHONEN, T. *Self-Organization and Associative Memory*. Springer Verlag, 1984.
- [50] KOSKO, B. *Fuzzy Engineering*. Prentice Hall, Upper Saddle River, N.J., 1992.
- [51] KOSKO, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, Englewood Cliffs, N.J., 1992.

- [52] LEE, D.-L. Improvements of complex-valued hopfield associative memory by using generalized projection rules. *IEEE Transactions on Neural Networks* 17, 5 (Sept. 2006), 1341–1347.
- [53] LIPPMANN, R. An introduction to computing with neural nets. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-4* (1987), 4–22.
- [54] LIU, P. The fuzzy associative memory of max-min fuzzy neural networks with threshold. *Fuzzy Sets and Systems* 107 (1999), 147–157.
- [55] MARAGOS, P. Lattice image processing: A unification of morphological and fuzzy algebraic systems. *Journal of Mathematical Imaging and Vision* 22, 2-3 (May 2005), 333–353.
- [56] MATHERON, G. *Elements Pour une Theorie des Milieux Poreux*. Masson, Paris, 1967.
- [57] MATHERON, G. *Théorie des Ensembles Aléatoires*. Ecole des Mines, Paris, 1967.
- [58] MATHERON, G. *Random Sets and Integral Geometry*. Wiley, New York, 1975.
- [59] THE MATHWORKS, INC. *MATLAB Documentation*, 2007. <http://www.mathworks.com>.
- [60] NACHTEGAEL, M., AND KERRE, E. Connections between binary, gray-scale and fuzzy mathematical morphologies. *Fuzzy Sets and Systems* 124, 1 (Nov. 2001), 73–85.
- [61] PEDRYCZ, W., AND GOMIDE, F. *Fuzzy Systems Engineering Toward Human-Centric Computing*. Wiley-IEEE Press, New York, 2007.
- [62] PESSOA, L., AND MARAGOS, P. Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition. *Pattern Recognition* 33 (2000), 945–960.
- [63] PETRIDIS, V., AND KABURLASOS, V. Fuzzy lattice neural network (FLNN): a hybrid model for learning. *IEEE Transactions on Neural Networks* 9, 5 (Sept. 1998), 877–890.

- [64] PITAS, I., AND VENETSANOPOULOS, A. Morphological shape decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 1 (1990), 38–45.
- [65] PORTER, R., HARVEY, N., PERKINS, S., THEILER, J., BRUMBY, S., BLOCH, J., GOKHALE, M., AND SZYMANSKI, J. Optimizing digital hardware perceptrons for multi-spectral image classification. *Journal of Mathematical Imaging and Vision* 19 (2003), 133–150.
- [66] RADUCANU, B., GRAÑA, M., AND ALBIZURI, X. F. Morphological scale spaces and associative morphological memories: Results on robustness and practical applications. *Journal of Mathematical Imaging and Vision* 19, 2 (2003), 113–131.
- [67] RIPLEY, B. D. Datasets for pattern recognition and neural networks, 1996. <http://www.stats.ox.ac.uk/pub/PRNN/>.
- [68] RIPLEY, B. D. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.
- [69] RITTER, G. X., AND SUSSNER, P. An introduction to morphological neural networks. In *Proceedings of the 13th International Conference on Pattern Recognition* (Vienna, Austria, 1996), pp. 709–717.
- [70] RITTER, G. X., AND SUSSNER, P. Morphological neural networks. In *Intelligent Systems: A Semiotic Perspective; Proceedings of the 1996 International Multidisciplinary Conference* (Gaithersburg, Maryland, 1996), pp. 221–226.
- [71] RITTER, G. X., AND SUSSNER, P. Associative memories based on lattice algebra. In *Computational Cybernetics and Simulation* (Orlando, Florida, 1997), 1997 IEEE International Conference on Systems, Man, and Cybernetics.
- [72] RITTER, G. X., AND SUSSNER, P. Morphological perceptrons. In *ISAS'97, Intelligent Systems and Semiotics* (Gaithersburg, Maryland, 1997).
- [73] RITTER, G. X., SUSSNER, P., AND DE LEON, J. L. D. Morphological associative memories. *IEEE Transactions on Neural Networks* 9, 2 (1998), 281–293.

- [74] RITTER, G. X., AND URCID, G. Lattice algebra approach to single-neuron computation. *IEEE Transactions on Neural Networks* 14, 2 (March 2003), 282–295.
- [75] RONSE, C. Why mathematical morphology needs complete lattices. *Signal Processing* 21, 2 (Oct. 1990), 129–154.
- [76] SERRA, J. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [77] SERRA, J. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, New York, 1988.
- [78] SOBANIA, A., AND EVANS, J. P. O. Morphological corner detector using paired triangular structuring elements. *Pattern Recognition* 38, 7 (2005), 1087–1098.
- [79] SOILLE, P. *Morphological Image Analysis*. Springer Verlag, Berlin, 1999.
- [80] STERNBERG, S. Parallel architecture for image processing. In *Proceedings of the Third International IEEE Compsac* (Chicago, 1979).
- [81] STERNBERG, S. Grayscale morphology. *Computer Vision, Graphics and Image Processing* 35 (1986), 333–355.
- [82] STOKBRO, K., UMBERGER, D., AND HERTZ, J. Exploiting neurons with localized receptive fields to learn chaos. *Complex Systems* 4 (1990), 603–622.
- [83] SUSSNER, P. Morphological perceptron learning. In *Proceedings of the IEEE International Symposium on Intelligent Control* (Gaithersburg, MD, 1998), pp. 477–482.
- [84] SUSSNER, P. Fixed points of autoassociative morphological memories. In *Proceedings of the International Joint Conference on Neural Networks* (Como, Italy, July 2000), pp. 611–616.
- [85] SUSSNER, P. Associative morphological memories based on variations of the kernel and dual kernel methods. *Neural Networks* 16, 5 (July 2003), 625–632.

- [86] SUSSNER, P. Generalizing operations of binary morphological autoassociative memories using fuzzy set theory. *Journal of Mathematical Imaging and Vision* 9, 2 (Sept. 2003), 81–93. Special Issue on Morphological Neural Networks.
- [87] SUSSNER, P., AND ESMI, E. A brief introduction to a two-layer morphological associative memory based on fuzzy operations. In *Proceedings of the 8th International Symposium on Mathematical Morphology* (Rio de Janeiro, Brazil, 2007), vol. 2, pp. 179–190.
- [88] SUSSNER, P., AND ESMI, E. Constructive morphological neural networks: some theoretical aspects and experimental results in classification. In *Studies in Computational Intelligence*, J. Kacprzyk, Ed., Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Germany, 2009.
- [89] SUSSNER, P., AND ESMI, E. Introduction to morphological perceptrons with competitive learning. In *Proceedings of the International Joint Conference on Neural Networks* (Atlanta, GA, 2009), pp. 3024–3031.
- [90] SUSSNER, P., AND ESMI, E. An introduction to the Kosko subethood fam. In *Proceedings of Conference on Hybrid Artificial Intelligence Systems* (2010). accepted for publication.
- [91] SUSSNER, P., AND ESMI, E. Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm. *Information Science* (2010). in press.
- [92] SUSSNER, P., AND VALLE, M. E. Grayscale morphological associative memories. *IEEE Transactions on Neural Networks* 17, 3 (May 2006), 559–570.
- [93] SUSSNER, P., AND VALLE, M. E. Implicative fuzzy associative memories. *IEEE Transactions on Fuzzy Systems* 14, 6 (2006), 793–807.
- [94] SUSSNER, P., AND VALLE, M. E. Recall of patterns using morphological and certain fuzzy morphological associative memories. In *Proceedings of the IEEE World Conference on Computational Intelligence 2006* (Vancouver, Canada, 2006).

- [95] SUSSNER, P., AND VALLE, M. E. Morphological and certain fuzzy morphological associative memories for classification and prediction. In *Computational Intelligence Based on Lattice Theory*, V. Kambhampati and G. X. Ritter, Eds., vol. 67. Springer Verlag, Heidelberg, Germany, 2007, pp. 149 – 173.
- [96] SUSSNER, P., AND VALLE, M. E. Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality. *Journal of Mathematical Imaging and Vision* 32, 2 (2008), 139–159.
- [97] SUSSNER, P., AND VALLE, M. E. Fuzzy associative memories and their relationship to mathematical morphology. In *Handbook of Granular Computing*, W. Pedrycz, A. Skowron, and V. Kreinovich, Eds. John Wiley and Sons, Inc., New York, 2008, ch. 33.
- [98] TANAKA, G., AND AIHARA, K. Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction. In *IEEE International Joint Conference on Neural Networks* (June 2008), pp. 3086–3092.
- [99] VALLE, M. E., AND SUSSNER, P. A general framework for fuzzy morphological associative memories. *Fuzzy Sets and Systems* 159, 7 (2008), 747–768.
- [100] WOLBERG, W., AND MANGASARIAN, O. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences* (U.S.A, 1990), vol. 87, pp. 9193–9196.
- [101] ZADEH, L. A. Fuzzy sets. *Information and Control* 8, 3 (1965), 338–353.
- [102] ZADEH, L. A. Fuzzy sets and information granularity. In *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. North Holland, Amsterdam, 1979, pp. 3–18.
- [103] ZHANG, B.-L., ZHANG, H., AND GE, S. S. Face recognition by applying wavelet subband representation and kernel associative memory. *IEEE Transactions on Neural Networks* 15, 1 (Jan. 2004), 166–177.