

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



**OTIMIZAÇÃO DO DESEMPENHO DE REDES
NEURAIS DO TIPO FUNÇÕES DE BASE
RADIAL UTILIZANDO VETORES BIPOLARES
ORTOGONAIS**

CAMILA DA CRUZ SANTOS

Uberlândia - MG
2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

OTIMIZAÇÃO DO DESEMPENHO DE REDES NEURAIS DO
TIPO FUNÇÕES DE BASE RADIAL UTILIZANDO
VETORES BIPOLARES ORTOGONAIS

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia perante os membros da banca examinadora, como requisito parcial para obtenção do título de Mestre em Ciências.

Orientador: Keiji Yamanaka, Phd

Uberlândia - MG
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

- S237o
2018 Santos, Camila da Cruz, 1995-
Otimização do desempenho de redes neurais do tipo funções de base radial utilizando vetores bipolares ortogonais / Camila da Cruz Santos. - 2018.
88 f. : il.
- Orientador: Keiji Yamanaka.
Dissertação (mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Disponível em: <http://dx.doi.org/10.14393/ufu.di.2018.1131>
Inclui bibliografia.
1. Engenharia elétrica - Teses. 2. Reconhecimento de padrões - Teses. 3. Redes neurais (Computação) - Teses. I. Yamanaka, Keiji. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.3

OTIMIZAÇÃO DO DESEMPENHO DE REDES NEURAIS DO TIPO FUNÇÕES DE BASE RADIAL UTILIZANDO VETORES BIPOLARES ORTOGONAIS

CAMILA DA CRUZ SANTOS

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia perante os membros da banca examinadora, como requisito parcial para obtenção do título de Mestre em Ciências.

Prof. Keiji Yamanaka, Dr.
FEELT/UFU (Orientador)

Prof. Igor Santos Peretta, Dr.
FEELT/UFU

Prof. José Ricardo Gonçalves Manzan, Dr.
IFTM - Campus Avançado Uberaba Parque Tecnológico

Prof. Hugo Xavier Rocha, Dr.
IFG - Campus Itumbiara

Uberlândia - MG
2018

Dedico este trabalho a Deus por nortear minha vida. Aos meus pais e irmã pelo exemplo, incentivo, amor e carinho. Ao meu companheiro e amigos pelo apoio e atenção nos momentos tristes e alegres.

Agradecimentos

Ao meu orientador Professor Keiji Yamanaka pela amizade, incentivo e confiança no meu trabalho. Obrigada por aceitar minhas dificuldades e limitações, por me permitir continuar nesse projeto de pesquisa.

Ao verdadeiro amigo, Professor José Ricardo Gonçalves Manzan, por toda colaboração, paciência, sugestão e incentivo. Suas visões, percepções e sugestões colaboraram de forma fundamental para o desenvolvimento desse trabalho.

Aos professores Igor Santos Peretta e Hugo Xavier Rocha pela aceitação em participar da banca e pelas observações em relação ao meu trabalho, que muito contribuíram para a melhoria do mesmo.

Aos meus pais, Cláudia e Reinaldo, por todos os ensinamentos de vida, por terem me guiado nos meus primeiros passos e por estarem servindo de referência para todos os outros passos que ainda terei que tomar. Por toda a dedicação, amor e carinho.

À minha irmã Cinthia e meu companheiro Willer, pelo apoio, paciência, compreensão e carinho.

À Cinara e Patrícia, por desenvolverem seu trabalho com seriedade e competência, sem deixar de lado valores como amizade, caridade e simpatia.

Aos demais mestres e amigos que foram, em diferentes níveis, fundamentais para minha formação e que me prestaram auxílio em muitas ocasiões.

Resumo

SANTOS, Camila C. *Otimização do desempenho de Redes Neurais do Tipo Funções de Base Radial utilizando Vetores Bipolares Ortogonais*, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2018.

Este trabalho propõe o uso de vetores bipolares ortogonais (VBO) como novos alvos para Redes Neurais Artificiais (RNA), do tipo Funções de Base Radial (RBF). Tais vetores propiciam a ampliação da distância entre os pontos do espaço de saída, também compreendidos como alvos da RNA. A ampliação dessa distância reduz as chances de classificação incorreta de padrões. A rede foi treinada e testada com três conjuntos de dados biométricos (íris humana, dígitos manuscritos e sinais australianos). O objetivo do trabalho é verificar a performance da rede com o uso dos vetores ortogonais e comparar os resultados obtidos com os apresentados para as redes do tipo Multilayer Perceptron. Além disso, deseja-se comparar duas técnicas de treinamento para redes do tipo RBF. Os conjuntos de dados utilizados nos experimentos foram obtidos do CASIA Iris Image Database desenvolvido pela Chinese Academy of Sciences - Institute of Automation, Semeion Handwritten Digit of Machine Learning Repository e UCI - Machine Learning Repository. As redes foram modeladas utilizando como alvo os VBOs e os vetores bipolares convencionais, para o propósito da comparação dos resultados. A classificação dos padrões na camada de saída foi baseada na distância euclidiana. A partir dos experimentos realizados, foi observado que o uso dos VBOs no processo de treinamento da rede melhorou a taxa de acerto e reduziu a quantidade de ciclos necessários para a convergência.

Palavras-chave: Reconhecimento de padrões, redes neurais artificiais, multilayer perceptron, funções de base radial, vetores bipolares ortogonais.

Abstract

SANTOS, Camila C. *Optimization of the performance of Neural Networks of the Radial Basis Functions type using Orthogonal Bipolar Vectors*, Uberlândia, Faculty of Electric Engineering - UFU, 2018.

This work proposes the use of orthogonal bipolar vectors (VBO) as new targets for Artificial Neural Networks (ANN) of the Radial Base Functions (RBF) type. Such vectors provide the expansion of the distance between the points of the output space, also understood as targets of RNA. The expansion of this distance reduces the chances of incorrect classification of patterns. The network was trained and tested with three sets of biometric data (human iris, handwritten digits and signs of the Australian sign language). The objective was to verify the network performance with the use of OBVs and compare the results obtained with those presented for the Multilayer Perceptron (MLP) networks. In addition, it is desired to compare two training techniques for RBF-type networks. Datasets used in the experiments were obtained from the CASIA Iris Image Database developed by the Chinese Academy of Sciences - Institute of Automation, Semeion Handwritten Digit of Machine Learning Repository and UCI - Machine Learning Repository. The networks were modeled using OBVs and conventional bipolar vectors for the purpose of comparing the results and the classification of the patterns in the output layer was based on the Euclidean distance. The results show that the use of OBVs in the network training process improved the hit rate and reduced the amount of cycles required for convergence.

Keywords: Pattern recognition, artificial neural networks, multilayer perceptron, radial basis function, orthogonal bipolar vectors.

Sumário

Lista de Figuras	ix
Lista de Tabelas	x
Lista de Siglas	xi
1 Introdução	13
1.1 Considerações iniciais	13
1.2 Motivação	14
1.3 Objetivos da dissertação	15
1.4 Estrutura do trabalho	16
2 Fundamentos teóricos	17
2.1 Reconhecimento de padrões	17
2.1.1 Conceitos de um sistema de reconhecimento de padrões	17
2.1.2 Etapas de um sistema de reconhecimento de padrões	19
2.1.3 Técnicas para classificação de padrões	21
2.2 Redes Neurais Artificiais	22
2.2.1 Neurônio biológico	24
2.2.2 Neurônio artificial	24
2.2.3 Tipos de Redes Neurais	26
2.2.4 Funções de ativação	27
2.2.5 Processos de Aprendizagem	27
2.2.5.1 Aprendizagem Supervisionada	29
2.2.5.2 Aprendizagem Não-supervisionada	29
2.3 Conceitos matemáticos envolvidos	30
2.3.1 Produto interno e distância euclidiana de vetores no espaço R^n	30
2.3.2 Ângulo e ortogonalidade entre vetores no espaço R^n	31
2.3.3 Teste Estatístico de Kolmogorov-Smirnov	32
2.3.4 Teste Estatístico de Mann-Whitney	33
3 Redes Neurais Artificiais do tipo Função de Base Radial	36
3.1 Arquitetura e características	36
3.2 Teoria da Separabilidade dos Padrões	37
3.3 Estratégias de Aprendizagem	38
3.3.1 Centros Fixos Seleccionados ao Acaso	39

3.3.2	Seleção Auto-Organizada de Centros	40
3.3.3	Seleção Supervisionada de Centros	40
3.4	Processo de Treinamento	41
3.4.1	Primeiro Estágio de Treinamento	41
3.4.2	Segundo Estágio de Treinamento	44
3.4.2.1	Otimização de Pesos - adaptação do método de Galerkin	44
3.4.2.2	Regra Delta Generalizada	47
3.4.2.3	Modificação na camada de saída	50
4	Alvos ortogonais em Redes de Funções Bases Radiais	51
4.1	Definição de vetores-alvo	51
4.2	Algoritmo de geração de vetores bipolares ortogonais	52
4.3	Observações sobre os vetores	55
4.4	Efeito dos alvos ortogonais no processo de treinamento	56
4.4.1	Técnica de Otimização dos pesos	57
4.4.2	Regra Delta Generalizada	59
5	Procedimentos Experimentais	64
5.1	Dados experimentais	64
5.1.1	Íris humana	64
5.1.2	Dígitos manuscritos	65
5.1.3	Signos da linguagem de sinais australiana	65
5.2	Vetores-alvo	67
5.3	Planejamento experimental	67
5.4	Parâmetros analisados	68
5.5	Resultados Experimentais	69
5.6	Discussão	77
6	Conclusão	80
A	Teste de Mann-Whitney	83
	Referências Bibliográficas	84

Lista de Figuras

2.1	Exemplo de separador de classes desejado	19
2.2	Exemplo de separador de classes próximo do ideal	19
2.3	Protótipo de um sistema de reconhecimento de padrões	20
2.4	Exemplo de histogramas representando características de um padrão	21
2.5	Ilustração de um neurônio biológico	24
2.6	Ilustração de um neurônio artificial	25
2.7	Diagrama de blocos da aprendizagem supervisionada	29
2.8	Diagrama de blocos da aprendizagem não-supervisionada	30
2.9	Ângulo entre vetores	31
2.10	Ilustração de vetores ortogonais	32
2.11	Esboço da região de aceitação e de rejeição para o Teste de Mann-Whitney	35
3.1	Arquitetura típica de uma RBF	37
3.2	Exemplos de dicotomias separáveis em 2 dimensões	38
3.3	Arquitetura de uma rede do tipo RBF modificada	50
4.1	Distância Euclidiana de vetores-alvo	56
4.2	Função de base radial do tipo gaussiana	57
4.3	Fronteira de separabilidade das redes MLP e RBF	58
4.4	Clusterização de amostras com VNO	59
4.5	Clusterização de amostras com VBO	59
4.6	Ilustração de regiões de convergência	62
4.7	Ilustração de regiões de convergência com alvos distantes	63
5.1	Comparação de resultados após a utilização de diferentes funções de base radial	72
5.2	Comparação de desempenho RBF - MRD: Dígitos Manuscritos	76
5.3	Comparação de desempenho RBF - MRD: Íris Humana	76
5.4	Comparação de desempenho RBF - MRD: Sinais Australianos	77

Lista de Tabelas

2.1	Principais funções de ativação	28
5.1	Dimensão dos vetores-alvo utilizados	67
5.2	Teste de normalidade de Kolmogorov-Smirnov - Dígitos Manuscritos	69
5.3	Teste de normalidade de Kolmogorov-Smirnov - Íris Humana	69
5.4	Teste de normalidade de Kolmogorov-Smirnov - Sinais Australianos	70
5.5	Teste de Mann-Whitney - Dígitos Manuscritos	70
5.6	Teste de Mann-Whitney - Íris Humana	71
5.7	Teste de Mann-Whitney - Sinais Australianos	71
5.8	Desempenho obtido com a rede Modelo de Haykin (MH)	73
5.9	Desempenho obtido com a rede Modelo Regra Delta (MRD)	74
5.10	Resultados MLP - Dígitos Manuscritos	74
5.11	Resultados MLP - Íris Humana	75
5.12	Resultados MLP - Sinais Australianos	75
5.13	Resultados RBF - MRD - Dígitos Manuscritos	75
5.14	Resultados RBF - MRD - Íris Humana	76
5.15	Resultados RBF - MRD - Sinais Australianos	77

Lista de Siglas e Abreviaturas

RNA	Rede Neural Artificial
MLP	Multilayer Perceptron
RBF	Função de Base Radial
VBO	Vetor Bipolar Ortogonal
VCN	Vetor Binário
VBC	Vetor Bipolar Convencional
VNO	Vetor Não Ortogonal
RP	Reconhecimento de Padrões
VC	Vetor Convencional
IC	Inteligência Computacional
MH	Modelo de Haykin
MRD	Modelo Regra Delta

Lista de Algoritmos

1	PRIMEIRO ESTÁGIO DE TREINAMENTO	43
2	SEGUNDO ESTÁGIO DE TREINAMENTO - OTIMIZAÇÃO DE PESOS	47
3	SEGUNDO ESTÁGIO DE TREINAMENTO - REGRA DELTA GENERALIZADA	49
4	Geração de vetores bipolares ortogonais	53

Capítulo 1

Introdução

1.1 Considerações iniciais

A Inteligência Computacional (IC) é uma área da ciência que utilizando técnicas inspiradas na Natureza, busca o desenvolvimento de sistemas inteligentes que imitem aspectos do comportamento humano, abrange Redes Neurais Artificiais (RNA), sistemas Fuzzy e computação evolutiva. É definida como uma metodologia que fornece a um sistema a capacidade para aprender e/ou lidar com uma nova situação, de modo que o sistema é caracterizado por possuir uma ou mais atribuições de raciocínio, como generalização, descoberta, associação e abstração. Essas técnicas têm sido combinadas entre si e com abordagens mais tradicionais, como a análise estatística, com o objetivo de resolver problemas extremamente desafiadores (Eberhart & Shi, 2007).

Muitas pesquisas são feitas nessa área e resultados promissores em problemas de recuperação de imagem têm sido obtidos por meio da aplicação de Redes Neurais Convolucionais (Wan et al., 2014). Utiliza-se a aprendizagem computacional baseada na inferência bayesiana para a predição de informações relacionadas às proteínas e suas interações (Birlutiu, d'Alche Buc, & Heskes, 2014). Outra aplicação recente é a separação de ruído e fala em sons de microfones (Healy, Yoho, Wang, & Wang, 2013). Também há importantes resultados de aplicação da inteligência computacional na área forense (Muda, Choo, Abraham, & Srihari, 2014). Parte

desse conjunto de técnicas já foi usada com sucesso na detecção de pedestres aptos a atravessarem ruas, com o objetivo de se evitarem acidentes (Xu et al., 2012).

Uma Rede Neural Artificial (RNA) é um paradigma de análise cuja modelagem é inspirada na estrutura do cérebro. Pode ser caracterizada como um processador paralelamente distribuído. Suas unidades de processamento possuem a função de armazenar e disponibilizar conhecimento experimental. Assemelha-se com o cérebro humano pelo fato de adquirir conhecimento através de processos de aprendizagem, além disso, trabalha com pesos sinápticos para armazenar o conhecimento (Haykin, 2008).

São amplamente utilizadas pela sua habilidade de aprender e conseqüentemente, generalizar a informação demonstrando seu poder computacional, podem ser utilizadas em problemas de classificação de padrões, aproximação de funções, entre outros. Seu uso em tarefas de reconhecimento de padrões é bastante consolidado. Nesse contexto há várias aplicações de RNAs relacionadas ao reconhecimento de padrões (Wan et al., 2014; Birlutiu et al., 2014; Muda et al., 2014; Xu et al., 2012; Huang, Huang, Song, & You, 2015).

As redes neurais do tipo Função de Base Radial (RBF) são apenas um tipo dentre os existentes. Uma RBF consiste em um modelo neural multicamadas, capaz de aprender padrões complexos e resolver problemas não linearmente separáveis. Amplamente utilizadas em várias aplicações, como prognóstico de baterias de lítio (Sbarufatti, Corbetta, Giglio, & Candini, 2017), previsão de vendas (Kuo, Hu, & Chen, 2009), solução de sistemas de equações integrais não lineares (Golbabai, Mammadov, & Seifollahi, 2019), predição de tráfego em rede ethernet (Vieira, Lemos, & Ling, 2003), entre outras.

1.2 Motivação

As RNAs são muito utilizadas por serem um conjunto de ferramentas complexas para resolução de problemas. Nos últimos anos, vários estudos foram realizados para melhorar o desempenho delas. Esses estudos têm abordagens como a melhoria do algoritmo de treinamento, a determinação de topologias ideais para cada problema, entre outros. Porém poucos estudos

foram feitos para melhorar o desempenho visando o espaço de saída.

Em 2004, o uso dos VBOs como vetores-alvo no treinamento de RNAs do tipo Multilayer Perceptron (MLP) foi apresentado pelos pesquisadores Keiji Yamanaka e Shiguo Nomura para a análise experimental no reconhecimento de padrões de placas de trânsito e dígitos manuscritos. Os experimentos demonstraram que estes vetores melhoraram a habilidade de generalização da MLP e reduziram o número de ciclos necessários para a convergência da rede (Nomura, Yamanaka, Katai, Kawakami, & Shiose, 2004, 2005).

Posteriormente, eles foram objeto de estudo nas pesquisas de mestrado e doutorado de José Ricardo Gonçalves Manzan, sob orientação dos pesquisadores Keiji Yamanaka e Shiguo Nomura. A análise experimental em dados de imagens, sons e EGGs, em redes MLP, demonstrou a mesma eficácia dos VBOs no reconhecimento destes padrões (Manzan, Yamanaka, & Nomura, 2011; Nomura, Manzan, & Yamanaka, 2010; Manzan, Nomura, Yamanaka, Carneiro, & Veiga, 2012; Manzan, Nomura, & Filho, 2014). Além disso, a pesquisa apresentou a comprovação matemática da eficácia dos VBOs em MLPs (Manzan, Nomura, & Yamanaka, 2011; Nomura, Manzan, & Yamanaka, 2011; Manzan, Nomura, & Yamanaka, 2012; Manzan, 2016).

A propriedade de ortogonalidade dos VBOs, onde a distância euclidiana obtida entre os alvos e a resposta produzida pela rede é maior quando comparada aos vetores convencionais, é o que proporciona a melhoria no desempenho da rede. Uma RBF é considerada um aproximador universal e uma alternativa popular à MLP, já que ela possui estrutura mais simples e processo de treinamento mais rápido (Du & Swamy, 2013).

1.3 Objetivos da dissertação

Considerando o ganho de desempenho usando Vetores Bipolares Ortogonais (VBOs) para redes de tipo MLP e o fato de que uma das utilizações da RBF é a classificação de padrões, é desejado investigar se esses novos alvos melhoram o desempenho de uma RNA do tipo RBF e se o uso de diferentes dimensões para os vetores de alvos altera a taxa de sucesso.

1.4 Estrutura do trabalho

No Capítulo 2 são apresentados fundamentos teóricos de Reconhecimento de Padrões (RP), RNA e os conceitos matemáticos envolvidos na pesquisa. O Capítulo 3 apresenta os fundamentos teóricos de RNAs do tipo RBF e os processos e estratégias de aprendizagem utilizados nos experimentos. Os Alvos Ortogonais e seus efeitos em redes RBF estão descritos e analisados no Capítulo 4. O procedimento experimental é definido e discutido no Capítulo 5. O Capítulo 6 apresenta a conclusão deste trabalho.

Capítulo 2

Fundamentos teóricos

2.1 Reconhecimento de padrões

O Reconhecimento de Padrões (RP) é uma das habilidades mais naturais dos seres vivos, e ao mesmo tempo é uma das mais extraordinárias. Ela é caracterizada pela capacidade de classificar e/ou separar objetos e dados em categorias, tornando-os generalizados, facilitando a tomada de decisão. A assimilação de escritas e conseqüentemente, a diferenciação de palavras ou idiomas são exemplos de RP no dia a dia.

Dentro da ciência da computação, RP é uma área derivada da inteligência computacional que tem por objetivo fazer com que computadores reproduzam a habilidade humana de separar/classificar dados em categorias específicas. O interesse na área de RP tem crescido muito devido as aplicações que, além de serem desafiantes, são também computacionalmente cada vez mais exigentes.

2.1.1 Conceitos de um sistema de reconhecimento de padrões

O RP é uma área de pesquisa que tem por objetivo a classificação de objetos (padrões) em um conjunto de categorias ou classes (Theodoridis & Koutroumbas, 1999). Um padrão é a descrição quantitativa ou qualitativa de um objeto ou de outra entidade de interesse em uma imagem ou em um sinal (Gonzalez, 1992). Essa descrição pode ser feita por uma ou mais

medidas que são denominadas atributos ou características do padrão. Um conjunto de padrões com características semelhantes é denominado classe (Gonzalez, 1992).

Há duas maneiras de se reconhecer e/ou classificar um padrão (Connel & Jain, 2001):

(i) classificação supervisionada: o padrão é identificado como um membro de uma classe pré-definida, ou seja, a classe é definida pelo projetista do sistema, ou (ii) classificação não-supervisionada: o padrão é determinado por uma “fronteira” de classe desconhecida.

Um problema de RP consiste em uma tarefa de classificação ou categorização, onde as classes são definidas pelo projetista do sistema (classificação supervisionada) ou são “aprendidas” de acordo com a similaridade dos padrões (classificação não-supervisionada) (Bianchi, 2006).

As técnicas de RP têm por concepção a seleção de características dos conjuntos de objetos e a separação dos objetos em suas devidas classes. De acordo com Duda et al. (2001), essas técnicas permitem uma representação mais simples de uma coleção de dados por meio das características que apresentam maior relevância, resultando na partição em classes. Essa representação é geralmente dada pela reunião das características em um vetor.

Contudo as técnicas de reconhecimento de padrões nem sempre são simples. Em um dado problema, simples ou complexo, busca-se sempre chegar a um separador com menor taxa de erro. Em boa parte dos problemas de classificação, a determinação do separador desejado exige muito esforço, tanto na determinação dos parâmetros da técnica utilizada, quanto no custo computacional. Na prática, a determinação de um separador desejado é quase sempre inviável. A Figura 2.1 representa um reconhecedor desejado capaz de separar amostras de duas classes predefinidas com 100% de acerto. Alguns problemas toleram certo grau de erro. Dessa maneira, é possível encontrar um separador com um esforço viável que consiga separar os elementos das classes com um bom nível de acerto. A Figura 2.2 representa um reconhecedor próximo do desejado, capaz de separar duas classes com alguns erros de separação.

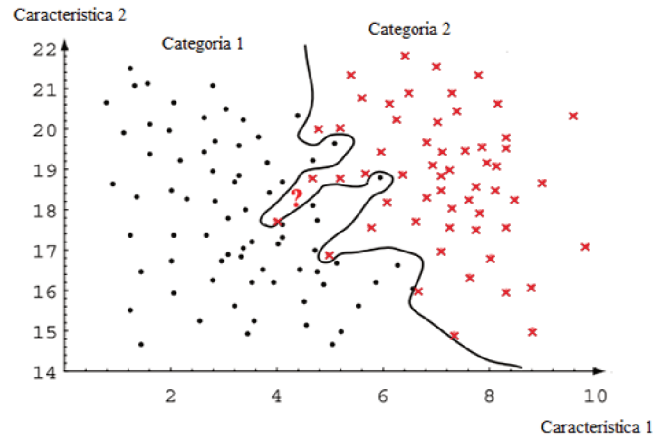


Figura 2.1: Exemplo de separador de classes desejado
 Fonte: (Duda, Hart, & Stork, 2001)

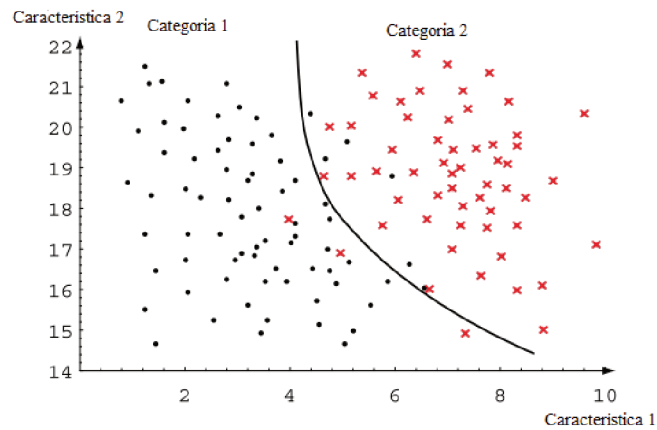


Figura 2.2: Exemplo de separador de classes próximo do desejado
 Fonte: (Duda et al., 2001)

2.1.2 Etapas de um sistema de reconhecimento de padrões

Em um sistema de reconhecimento de padrões, algumas etapas bem definidas são necessárias. Duda et al. (2001) sugere uma divisão e ordem para realização desta tarefa, a primeira parte é a modelagem do sistema, mostrada na Figura 2.3. Para isso, precisamos conhecer o conjunto de dados que será utilizado e garantir que existem diferenças entre a população. O objetivo e a abordagem abrangentes na classificação dos padrões são hipotetizar a classe desses modelos, processar os dados detectados para eliminar o ruído (não devido aos modelos) e, para qualquer padrão detectado, escolher o modelo que melhor corresponda.

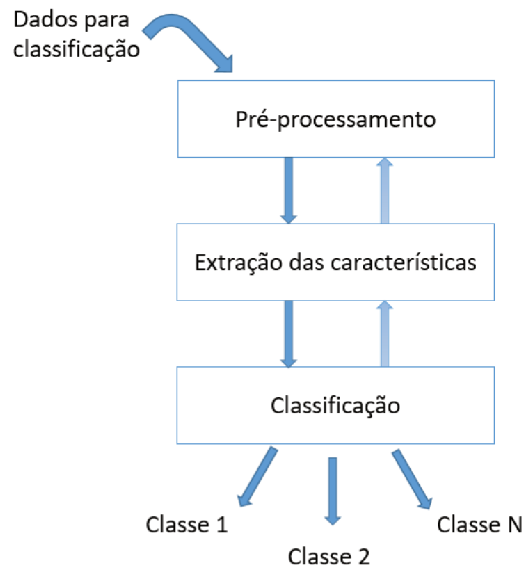


Figura 2.3: Protótipo de um sistema de reconhecimento de padrões
Fonte: (Duda et al., 2001)

Em seguida, faz-se o pré-processamento, onde os dados são capturados do ambiente externo e os sinais pré-processados para simplificar as operações subsequentes sem perder informações. A informação de um único dado é enviada para um extrator de recursos, cujo objetivo é reduzir os dados analisando algumas características ou propriedades. Os valores dos recursos obtidos são passados para um classificador que avalia as evidências apresentadas e toma uma decisão final quanto à classe do padrão inserido.

Ainda de acordo com Duda et al. (2001), as características que são escolhidas afetam diretamente no processo de classificação. Uma característica que não é muito bem definida, pode deixar muitos pontos de intersecção no gráfico, fazendo com que o classificador não saiba qual classe o padrão pertence, como pode ser observado na Figura 2.4 (a). Porém quando se escolhe uma característica que apresenta valores distintos para as categorias utilizadas, a chance do sistema classificar um padrão de maneira incorreta é muito menor, conforme apresentado no exemplo da Figura 2.4 (b).

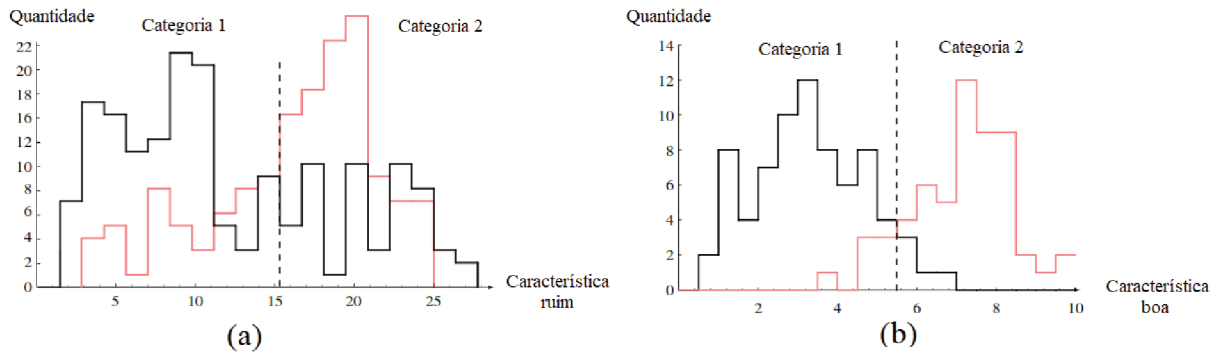


Figura 2.4: Exemplo de histograma utilizando (a) uma característica ruim para classificação de 2 categorias (b) uma característica boa para classificação de 2 categorias

Fonte: (Duda et al., 2001)

2.1.3 Técnicas para classificação de padrões

Dentre as técnicas de reconhecimento de padrões mais utilizadas, destacam-se a abordagem estatística (paramétrica e não paramétrica) e a abordagem conexionista, que utiliza RNAs. No caso da abordagem estatística, o conjunto de dados de treinamento serve para realizar a estimação de parâmetros estatísticos de cada classe. Dessa maneira, cada classe terá uma distribuição específica, formando assim, o classificador estatístico (Duda et al., 2001).

A abordagem estatística não paramétrica é subdividida em várias técnicas. Algumas delas se dão com a utilização de uma função de distância do objeto a ser mapeado em relação às classes disponíveis no espaço de características, direcionando padrões desconhecidos àquela classe que detém a menor distância com ele. As principais representantes das técnicas de distância são a distância euclidiana e a distância de Mahalanobis. A distância de Mahalanobis apresenta a vantagem de considerar a matriz de covariância de todas as classes além da média aritmética. Assim, problemas que contenham classes com diferentes variâncias serão melhor classificados com a distância de Mahalanobis. Dentro das técnicas não paramétricas, existem ainda as funções de Kernel, os K-vizinhos mais próximos, entre outros (Duda et al., 2001).

Na abordagem estatística paramétrica, destaca-se o classificador de Bayes, que considera a probabilidade de um objeto desconhecido pertencer a uma determinada classe. De acordo com Duda et al. (2001), este também é considerado um tipo ótimo de classificador,

pois minimiza a probabilidade média de erro na classificação. Existem também a função de discriminação e a regra Naive de Bayes.

A abordagem conexionista é composta por tipos de classificadores mais complexos, capazes de se adaptarem a qualquer tipo de distribuição de dados. São baseados no funcionamento das estruturas neurais inteligentes que aprendem por meio de exemplos, como as RNAs.

Trabalhos destinados ao estudo do reconhecimento de padrões, tanto no que se refere à descrição, quanto à classificação e desde 2001, têm ganhando grande destaque no campo da computação (Duda et al., 2001). Atualmente, as RNAs têm se tornado uma técnica amplamente utilizada em razão de resultados bastante satisfatórios e promissores.

2.2 Redes Neurais Artificiais

Redes Neurais Artificiais são modelos inspirados no sistema nervoso de seres vivos e possuem capacidade de aquisição e manutenção do conhecimento. Elas são formadas por um conjunto de neurônios artificiais, que consistem em unidades de processamento da rede, que são interligadas por conexões, chamadas de pesos sinápticos.

Podem-se entender as RNAs como mecanismos capazes de receber o sinal de determinado padrão na sua entrada, analisá-lo e então informar sobre a classe a qual ele pertence. Sua metodologia de treinamento é inspirada no funcionamento dos neurônios biológicos, em que aprendizagem ocorre por meio de exemplos, fazendo com que a tentativa e o erro desencadeiem o processo de apropriação da habilidade de diferenciar padrões.

O primeiro registro do surgimento das RNAs aconteceu com a publicação de um artigo de McCulloch e Pitts em 1943 (McCulloch & Pitts, 1943). De acordo com Silva et al. (2010), essa nova área da computação tinha por concepção o funcionamento dos neurônios biológicos. Em 1949, Donald Hebb apresentou o primeiro método de treinamento para RNAs (Hebb, 1968). Entre 1957 e 1958, Frank Rosenblat desenvolveu o primeiro neurocomputador e, no período de 1958 a 1962, criou uma grande classe de RNAs denominada como Perceptrons (Fausett & Hall, 1994).

A rede ADALINE (Adaptive Linear Element) foi desenvolvida em 1960 por Widrow e Hoff, posteriormente, recebeu aperfeiçoamentos correspondentes à associação de múltiplas redes Adaline, resultando no nome MADALINE (Fausett & Hall, 1994). Os resultados obtidos nessas pesquisas motivaram vários pesquisadores a estudarem as RNAs, até que, em 1969, Minsky e Papert demonstraram matematicamente as limitações das redes constituídas de uma única camada, como o Perceptron e o Adaline. No clássico livro *Perceptrons - An Introduction to Computational Geometry*, eles usam um simples problema de lógica denominado “ou exclusivo” para mostrarem que essas redes eram incapazes de resolvê-lo (Fausett & Hall, 1994).

O trabalho de Minsk e Papert causou grande impacto entre os pesquisadores da neurocomputação, fazendo com que o interesse pela área ficasse bastante reduzido. Isso, de certa forma, ocasionou a ausência de novos resultados para a área por um longo período (Silva, Spatti, & Flauzino, 2010). Nesse período de relativa turbulência das pesquisas envolvendo RNAs, foi implementada a rede ART (Adaptive Resonance Theory). Num trabalho de Grossberg, em 1980, foi realizada a formulação de mapas auto-organizáveis de Kohonen em 1982 e a proposta de redes recorrentes de Hopfield em 1982. A partir do trabalho de Hopfield, a neurocomputação voltou a receber a atenção dos pesquisadores.

Contudo a teoria das RNAs conseguiu realmente se estabelecer a partir da publicação do livro de Rumelhart, Hinton e Williams, *Parallel Distributed Processing*, em 1986. Nele, os autores mostraram um algoritmo capaz de treinar redes com múltiplas camadas e que, por sua vez, eram capazes de resolver as limitações apresentadas por Minsk e Papert. Esses acontecimentos se deram no momento em que estavam sendo desenvolvidos computadores com maior capacidade de processamento (Silva et al., 2010). Mais recentemente, novas contribuições e inúmeras aplicações práticas, em diferentes ramos do conhecimento têm permitido o desenvolvimento de estudos associados às RNAs.

2.2.1 Neurônio biológico

A Figura 2.5, presente em (Silva et al., 2010), ilustra um neurônio biológico, o qual é dividido em três partes principais: os dendritos, a soma ou corpo celular e o axônio.

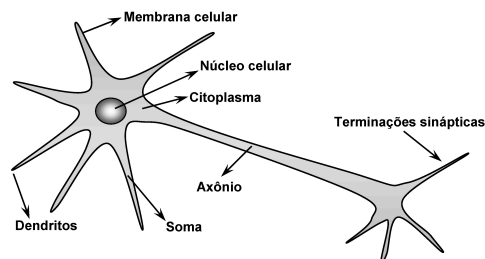


Figura 2.5: Ilustração de um neurônio biológico
Fonte: (Silva et al., 2010)

Os dendritos são especializados em receber os estímulos vindos de outros neurônios ou do meio externo que estão associados. O corpo celular processa a informação recebida, criando um potencial de ativação que, posteriormente, poderá ou não ser enviado ao axônio (Silva et al., 2010).

O axônio apresenta comprimento muito variável podendo medir de alguns milímetros até metros e é especializado em gerar e conduzir o potencial de ação, que ocorre por meio de substâncias neurotransmissoras. A porção terminal do axônio sofre várias ramificações para formar de centenas a milhares de terminações sinápticas, no interior das quais são armazenados os neurotransmissores químicos usados no envio de informação das sinapses para os dendritos de outros neurônios (Fausett & Hall, 1994).

O conjunto de bilhões dessas estruturas compõe a complexa estrutura denominada cérebro humano, que é capaz de realizar inúmeras tarefas com alto grau de dificuldade.

2.2.2 Neurônio artificial

Um neurônio artificial é uma unidade de processamento da informação que é fundamental para a operação de uma rede neural. Eles recebem a informação, a processam de acordo com seu papel dentro da rede e enviam uma nova informação para outros neurônios ou para a

saída do sistema. A Figura 2.6 mostra o modelo de um neurônio básico, proposto por MucCulloch e Pitts para o projeto de redes neurais, a partir dela verifica-se que o neurônio é constituído de 7 elementos básicos (Silva et al., 2010):

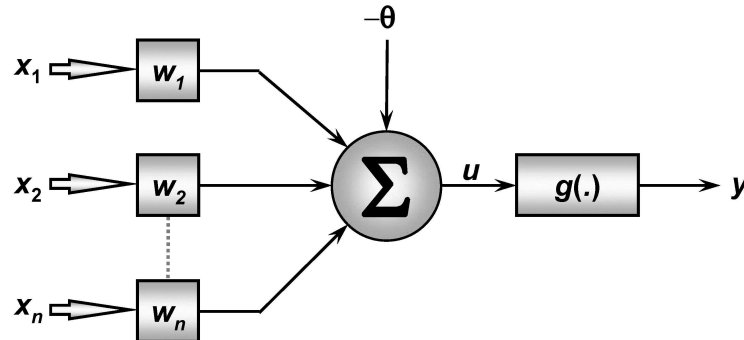


Figura 2.6: Ilustração de um neurônio artificial

Fonte: (Silva et al., 2010)

- Camada de entrada $\{x_1, x_2, \dots, x_n\}$: sinais vindos do meio externo;
- Pesos sinápticos $\{w_1, w_2, \dots, w_n\}$: valores para ponderar cada uma das variáveis de entrada da rede;
- Combinador linear $\{\Sigma\}$: agregar os sinais de entrada após a ponderação;
- Limiar de ativação $\{\theta\}$: é uma variável que especifica qual será o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção à saída do neurônio;
- Potencial de ativação $\{u\}$: representa a diferença do valor produzido entre o combinador linear e o limiar de ativação;
- Função de ativação $\{g\}$: cujo objetivo é restringir a amplitude da saída;
- Camada de saída $\{y\}$: valor final produzido pelo neurônio

O modelo proposto por MucCulloch e Pitts tem em sua concepção de funcionamento o processamento paralelo da informação com alta conectividade e é o modelo mais utilizado nos modelos de RNAs. Em termos matemáticos, podemos descrever o neurônio pelo seguinte

par de equações (Silva et al., 2010):

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (2.1)$$

$$y = g(u) \quad (2.2)$$

2.2.3 Tipos de Redes Neurais

Há vários tipos de RNAs, algumas possuem uma arquitetura mais simples, na qual há apenas duas camadas, sendo uma de entrada e outra de saída. Por outro lado, existem arquiteturas mais complexas, com a existência de três ou mais camadas. As camadas adicionais são conceituadas como intermediárias ou ocultas.

Em redes multicamadas, a camada de entrada tem a função do recebimento dos sinais advindos do ambiente externo. A associação das características, bem como a separação das classes são feitas pelos neurônios da(s) camada(s) intermediária(s). Fica a cargo da camada de saída a apresentação dos resultados finais da rede.

Perceptron e Adaline são tipos de RNAs mais simples, nelas a camada de separação das classes é a própria camada de saída. Esses tipos de redes são hábeis para problemas que têm classes linearmente separáveis. Problemas que não são linearmente separáveis requerem redes com mais de duas camadas, ou seja, pelo menos uma camada intermediária. Os exemplos mais conhecidos são as redes MLP, RBF e Hopfield.

A rede RBF será descrita no Capítulo 3, já a MLP é uma rede multicamadas e, de acordo com Haykin (2008), ela possui três características que a distinguem das outras:

1. Para cada neurônio da rede há uma função de ativação não linear, cuja curvatura é suave.
2. A rede contém uma ou mais camadas ocultas, que são diferentes da camada de entrada e saída
3. Existe um alto grau de conectividade entre os neurônios, isto é, um neurônio de qualquer camada está conectado a todos os outros da camada anterior.

2.2.4 Funções de ativação

As funções de ativação possuem o objetivo de processar e mapear toda somatória dos estímulos, provenientes dos neurônios que as alimenta com informações e ponderados pelos pesos. O resultado deste processamento faz com que o neurônio alcance um novo estado de ativação que será propagado através de sua saída (Araújo, 2005). As funções de ativação também são responsáveis por limitar o sinal da saída a um intervalo de interesse (Silva et al., 2010)

As funções mais utilizadas são expostas na Tabela 2.1.

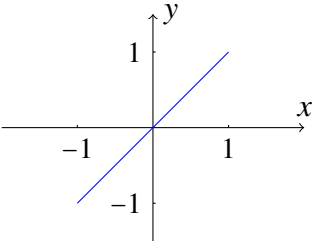
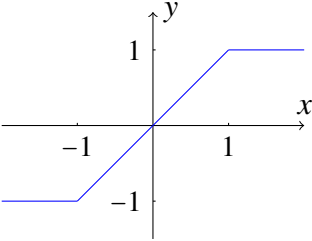
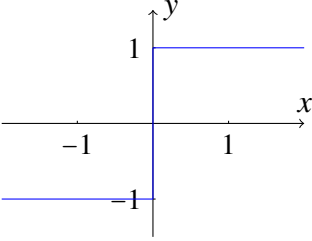
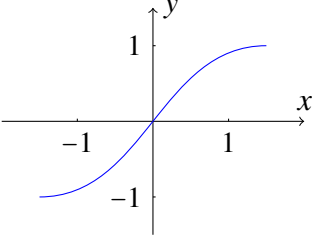
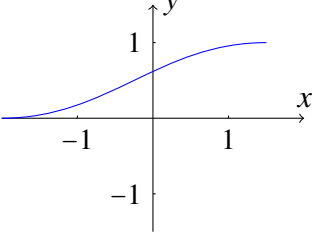
2.2.5 Processos de Aprendizagem

O objetivo fundamental de uma rede neural é utilizar o processo de aprendizagem para melhorar seu desempenho. O tempo e algumas medidas pré-estabelecidas é que vão determinar o aperfeiçoamento da taxa de acerto. Idealmente, após cada iteração do processo de aprendizagem a rede se torna mais preparada sobre seu ambiente (Haykin, 2008).

A aprendizagem no contexto de redes neurais pode ser definida como um processo que através de estímulos vindos do ambiente em que a rede está inserida adapta os parâmetros livres desta rede (Haykin, 2008). A forma que os parâmetros são modificados é que determina o tipo da aprendizagem.

Um algoritmo de aprendizagem é definido como um conjunto de regras bem-definidas para a solução de um problema de aprendizagem. Para as redes neurais, como esperado, não há um único algoritmo de aprendizagem (Haykin, 2008). Durante a execução desses algoritmos, a rede será capaz de extrair as características discriminantes do sistema a ser mapeado a partir de amostras que foram recebidas inicialmente vindas do ambiente ao qual ela está inserida (Silva et al., 2010).

Tabela 2.1: Principais funções de ativação adaptada da Tabela 3.1 de Araújo, 2005

Função	Equação	Representação Gráfica
Linear	$f(n) = n$	
Rampa	$f(n) = \begin{cases} 1 & \text{se } n \geq 1 \\ n & \text{se } n < 1 \\ -1 & \text{se } n \leq -1 \end{cases}$	
Degrau	$f(n) = \begin{cases} 1 & \text{se } n \geq 0 \\ 0 & \text{se } n < 0 \end{cases}$	
Tangente hiperbólica (logística)	$f(n) = \frac{2}{(1+e^{-2n})} - 1$	
Sigmoide (logística)	$f(n) = \frac{2}{1+e^{-n}} - 1$	

2.2.5.1 Aprendizagem Supervisionada

A aprendizagem supervisionada, também chamada de aprendizagem com um professor, é uma estratégia de treinamento que necessita ter disponível as saídas desejadas para cada amostra inserida. A Figura 2.7 representa um diagrama de blocos que ilustra esta forma de aprendizagem.

Podemos considerar que o professor, que representa as saídas desejadas, possui o conhecimento do ambiente, porém a rede não o conhece. O professor fornece para a rede a resposta desejada e os parâmetros são ajustados de acordo com o sinal do erro informado pelo professor, que é a diferença da resposta desejada com a apresentada pela rede. Este ajuste dos parâmetros é realizado várias vezes, até a rede conseguir emular o professor, dizemos então que o conhecimento do professor foi transferido para a rede. (Haykin, 2008)

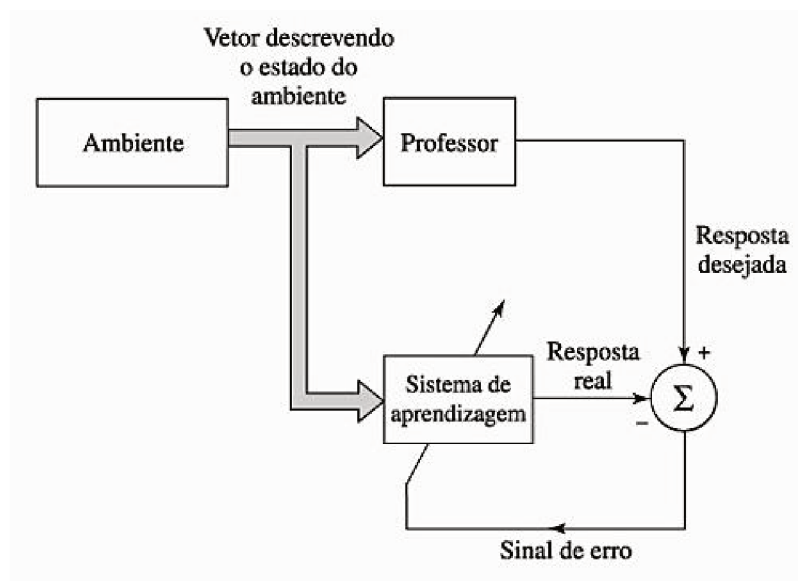


Figura 2.7: Diagrama de blocos da aprendizagem supervisionada
Fonte: (Haykin, 2008)

2.2.5.2 Aprendizagem Não-supervisionada

Neste algoritmo, não existem as saídas desejadas, diferentemente do processo de aprendizagem supervisionada. A própria rede precisa identificar os subconjuntos que contêm similaridade entre as amostras e se auto-organizar a partir dessas particularidades. Os

pesos sinápticos e os limiares dos neurônios da rede são então ajustados de forma a refletir as particularidades internamente dentro da própria rede (Silva et al., 2010).

A quantidade máxima desses possíveis subconjuntos pode ser determinada pelo projetista, levando em conta seu conhecimento a respeito do problema, ou pela própria rede (Silva et al., 2010). A Figura 2.8 mostra um diagrama de blocos com essa forma de aprendizagem.



Figura 2.8: Diagrama de blocos da aprendizagem não-supervisionada
Fonte: (Haykin, 2008)

2.3 Conceitos matemáticos envolvidos

Os conceitos matemáticos apresentados nas seções seguintes, serão utilizados na justificativa do uso do VBOs e na comparação entre os vetores VBO, VNO e VBC.

2.3.1 Produto interno e distância euclidiana de vetores no espaço R^n

Considere-se que $\vec{V}_i = (v_1, v_2, \dots, v_n)$ e $\vec{U}_i = (u_1, u_2, \dots, u_n)$ sejam dois vetores do espaço R^n . O produto interno desses vetores, representado pela Eq. 2.3 é uma das operações que podem ser realizadas entre vetores pertencentes a espaços de qualquer dimensão.

$$\vec{V}_i \bullet \vec{U}_i = v_1 \cdot u_1 + v_2 \cdot u_2 + \dots + v_n \cdot u_n \quad (2.3)$$

Outra operação que será realizada independente da dimensão do espaço R_n é a distância euclidiana. A Eq. 2.4 representa essa distância calculada entre \vec{V}_i e \vec{U}_i .

$$d_{V,U} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2} \quad (2.4)$$

2.3.2 Ângulo e ortogonalidade entre vetores no espaço R^n

Considere-se ainda que os vetores \vec{V}_i e \vec{U}_i pertencem ao espaço R^n . A razão do produto interno e o produto entre os módulos dos vetores são usados para calcular o cosseno do ângulo formado entre \vec{V}_i e \vec{U}_i , representado pela equação

$$\cos(\theta) = \frac{\vec{V}_i \cdot \vec{U}_i}{\|\vec{V}_i\| \|\vec{U}_i\|} \quad (2.5)$$

O ângulo θ formado entre estes vetores pode ser determinado por meio da função arco-cosseno, a inversa da função cosseno, representada pela Eq. 2.6. A Figura 2.5 ilustra o ângulo entre dois vetores do espaço R^3 .

$$\theta = \arccos\left(\frac{\vec{V}_i \cdot \vec{U}_i}{\|\vec{V}_i\| \|\vec{U}_i\|}\right) \quad (2.6)$$

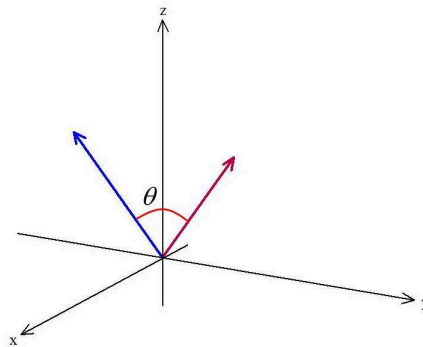


Figura 2.9: Ângulo entre vetores

Podem-se destacar, a partir do cálculo para determinação do ângulo entre vetores no espaço R^n :

- A função arco-cosseno tem sua imagem restrita ao intervalo $[0, \pi]$.
- O produto dos módulos dos vetores dado por $\|\vec{V}_i\| \|\vec{U}_i\|$ é sempre positivo.
- O cosseno é positivo no intervalo de $[0, \frac{\pi}{2}[$.
- O cosseno é nulo para o arco $\frac{\pi}{2}$.
- O cosseno é negativo no intervalo de $]\frac{\pi}{2}, \pi]$.

A partir das afirmações anteriores, nota-se que o ângulo entre os vetores \vec{V}_i e \vec{U}_i é agudo sempre que o produto interno for positivo, e reto sempre que o produto interno for igual a zero e é obtuso sempre que o produto interno for negativo.

Portanto dois vetores são ortogonais se o produto interno entre eles for igual a zero. A Figura 2.10 representa dois vetores com essa característica.

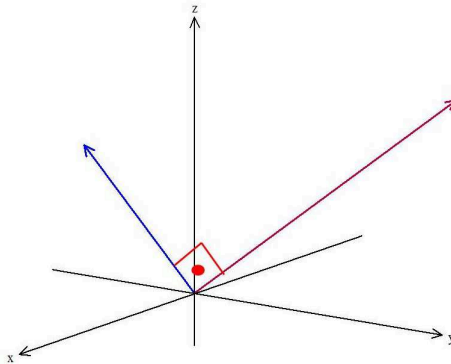


Figura 2.10: Ilustração de vetores ortogonais

2.3.3 Teste Estatístico de Kolmogorov-Smirnov

Os métodos estatísticos dividem-se basicamente em duas categorias: paramétricos e não paramétricos. Os testes paramétricos pressupõem que os dados se ajustam a uma distribuição normal, além disso, dependem de parâmetros como a média e a variância. Os não paramétricos não dependem destes parâmetros nem do ajuste à normalidade pela distribuição dos dados analisados.

Assim é importante saber se os dados a serem analisados se ajustam a distribuição normal antes de decidir qual teste aplicar. O teste de Kolmogorov-Smirnov permite verificar se uma amostra de dados se ajusta a alguma distribuição teórica, por exemplo, a distribuição normal (Conover, 1999). Trata-se de um teste bastante utilizado e, nas análises estatísticas deste trabalho, foi empregado para decisão sobre a utilização de um teste paramétrico ou não paramétrico.

Para a realização do teste, os seguintes passos devem ser executados (Conover, 1999):

1. Ordenam-se os dados x_i em ordem crescente.

2. Para cada x_i dado é atribuído o valor $F_n(x_i)$, denominado valor empírico, que é o resultado da razão de i por n , em que n é o total de dados.
3. Para cada i , atribui-se o valor normal padronizado correspondente $F(x_i)$, denominado valor teórico, que é obtido pela Tabela de distribuição normal padrão e pela fórmula representada pela equação

$$Z_i = \frac{x_i - \bar{x}}{\sigma} \quad (2.7)$$

em que σ é o desvio-padrão da amostra de dados e \bar{x} é a média dos dados.

4. Para cada i , calcula-se o módulo da diferença entre o valor teórico e o valor empírico $|F(x_i) - F_n(x_i)|$.
5. Para cada i , calcula-se o módulo da diferença entre o valor teórico e o valor empírico anterior $|F(x_i) - F_n(x_{i-1})|$.
6. Calcula-se a somatória das diferenças entre o valor teórico e o valor empírico $\Sigma |F(x_i) - F_n(x_i)|$, denominada D^+ .
7. Calcula-se a somatória das diferenças entre o valor teórico e o valor empírico anterior $\Sigma |F(x_i) - F_n(x_{i-1})|$, denominada D^- .
8. Obtém-se o máximo entre D^+ e D^- , denominado D_n .
9. Para um nível de significância α e a quantidade de amostras n , compara-se o valor crítico V da Tabela com o valor de D_n . Se $D_n \leq V$, os dados seguem uma distribuição normal. Se $D_n > V$, os dados não seguem uma distribuição normal.

2.3.4 Teste Estatístico de Mann-Whitney

Os testes não paramétricos são utilizados quando a amostra dos dados não se ajusta a uma distribuição normal. O Teste de Mann-Whitney (Martins & Fonseca, 2006) permite a comparação de igualdade de médias sem a exigência de igualdade de variâncias em problemas de comparação de médias em grupos independentes.

Para a realização do teste, deve-se seguir os passos:

1. Considera-se n_1 como o número de amostras do menor grupo e n_2 como o número de amostras do maior grupo.
2. Os dados dos dois grupos são reunidos e organizados em ordem crescente. O menor dado recebe o número 1 e todos os outros são ordenados até o último dado, que corresponde a $N = n_1 + n_2$.
3. Para as amostras iguais (empatadas), calcula-se a média entre seus postos, e cada uma recebe o posto médio.
4. Calcula-se R_1 = soma dos postos do grupo n_1 e R_2 = soma dos postos do grupo n_2 .
5. Escolhe-se a menor soma entre R_1 e R_2 .
6. Calculam-se as estatísticas:

$$\mu_1 = n_1 \cdot n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad (2.8)$$

$$\mu_2 = n_1 \cdot n_2 + \frac{n_2(n_2 + 1)}{2} - R_2 \quad (2.9)$$

7. Hipóteses do teste:
 H_0 : não há diferença entre os grupos.
 H_1 : há diferença entre os grupos.
8. Definição do nível α de significância.
9. Com o auxílio da Tabela de distribuição de probabilidades normal padronizada, definem-se as regiões de aceitação e de rejeição de H_0 conforme ilustração da Figura 2.11.
10. Cálculo do valor variável Z_{cal} :

$$Z_{cal} = \frac{\mu - \mu(u)}{\sigma(u)} \quad (2.10)$$

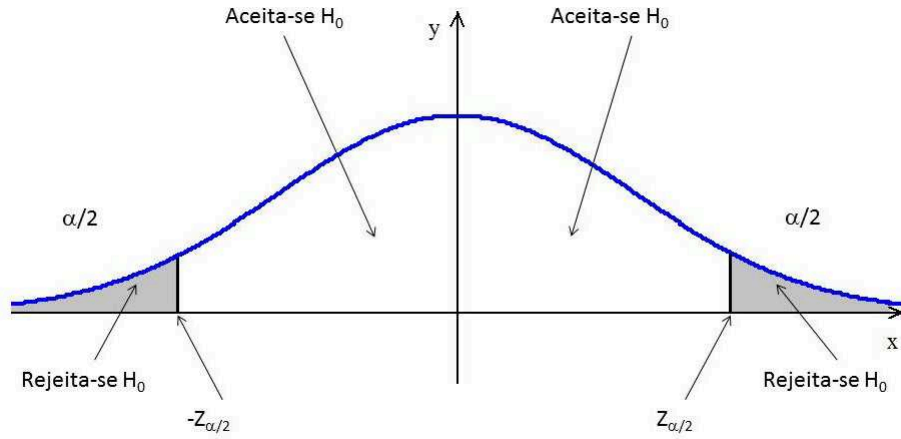


Figura 2.11: Esboço da região de aceitação e de rejeição para o Teste de Mann-Whitney

em que

$$\mu(u) = \frac{n_1 \cdot n_2}{2} \quad (2.11)$$

$$\sigma(u) = \sqrt{\frac{n_1 \cdot n_2 (n_1 + n_2 + 1)}{12}} \quad (2.12)$$

11. Conclusão:

Se $-Z_{\frac{\alpha}{2}} \leq Z_{cal} \leq Z_{\frac{\alpha}{2}}$, aceita-se H_0 .

Se $-Z_{cal} > Z_{\frac{\alpha}{2}}$ ou $Z_{cal} < -Z_{\frac{\alpha}{2}}$, rejeita-se H_0 , assumindo-se com risco α que os grupos possuem diferença em relação às suas médias.

Capítulo 3

Redes Neurais Artificiais do tipo Função de Base Radial

As redes do tipo RBF (*Radial Basis Function, em inglês*), são RNAs que utilizam a arquitetura *feedforward*, ou seja, os dados são propagados da camada de entrada, seguindo para a camada oculta, até chegarem na camada de saída, sem retroalimentação da rede (Fausett & Hall, 1994). Podem ser empregadas em quase todos os problemas tratados pela MLP incluindo problemas de aproximação funcional e reconhecimento de padrões (Silva et al., 2010).

3.1 Arquitetura e características

A construção tradicional de uma RBF apresenta 3 camadas totalmente distintas, que podem ser observadas na Figura 3.1. A camada de entrada é constituída por unidades sensoriais que interligam a rede com o ambiente externo. A camada oculta faz transformações não lineares no espaço de entrada, sendo caracterizada pela utilização de funções de base radial. E a camada de saída é responsável pela resposta final da rede modelada (Haykin, 2008). Os pesos sinápticos da camada intermediária, também chamados de centros, representam os centros dos subgrupos de dados categorizados e são utilizados como centros das funções de bases radiais no processo de treinamento.

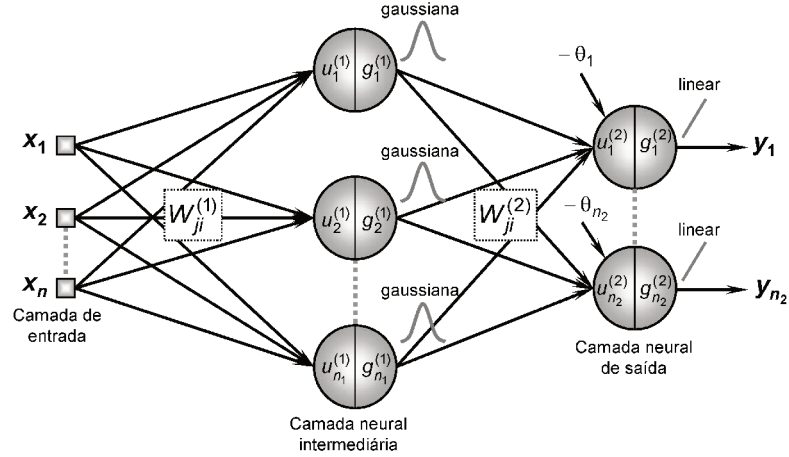


Figura 3.1: Arquitetura típica de uma RBF
Fonte: (Silva et al., 2010)

3.2 Teoria da Separabilidade dos Padrões

Quando uma tarefa complexa de classificação de padrões será realizada por uma rede RBF, o problema é resolvido pela sua transformação para um espaço de alta dimensionalidade, de uma forma não-linear. Esse processo foi justificado por Cover (1965) em seu teorema sobre a separabilidade dos padrões.

Considere que Ω represente um conjunto de N padrões divididos em duas classes. A partição binária (dicotomia) dos pontos pode ser feita se existir uma superfície que separe os pontos da classe Ω_1 da classe Ω_2 (Haykin, 2008). Para cada padrão $x \in \Omega$, é definido um vetor constituído de um conjunto de funções, como mostra a Equação (3.1)

$$\phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_{m_1}(x)]^T \quad (3.1)$$

Considerando que o padrão x representa um vetor de entrada em um espaço de dimensão m_0 , o vetor $\phi(x)$ mapeia os pontos desse padrão de entrada para um novo espaço de dimensão m_1 , onde $m_1 > m_0$ (Haykin, 2008). Uma dicotomia $\{\Omega_1, \Omega_2\}$ de Ω é dita ser separável

por ϕ , se existir um vetor w de dimensão m_1 para o qual podemos escrever (Cover, 1965):

$$\begin{aligned} w^T \phi(x) &> 0, & x \in \Omega_1 \\ w^T \phi(x) &\leq 0, & x \in \Omega_2 \end{aligned} \quad (3.2)$$

O hiperplano que descreve a superfície de separação no espaço ϕ é definido pela Equação (3.3). A Figura 3.2 mostra três exemplos de dicotomias separáveis por ϕ de diferentes conjuntos de cinco pontos em duas dimensões.

$$w^T \phi(x) = 0 \quad (3.3)$$

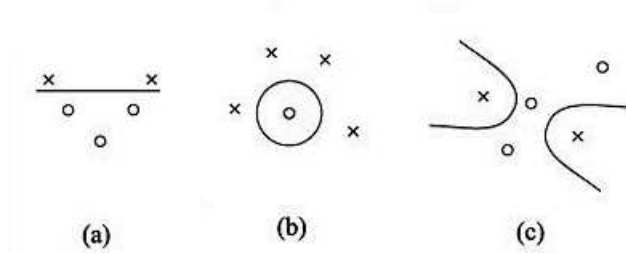


Figura 3.2: Exemplos de dicotomias separáveis em 2 dimensões (a) dicotomia linearmente separável; (b) dicotomia esfericamente separável; (c) dicotomia quadricamente separável

Fonte: (Haykin, 2008)

De acordo com Haykin (2008), o teorema do Cover sobre a separabilidade dos padrões engloba dois ingredientes básicos:

1. A Formulação não-linear da função oculta definida por $\phi_i(x)$, onde x é o vetor de entrada e $i = 1, 2, \dots, m_1$.
2. A alta dimensionalidade do espaço oculto comparado com o espaço de entrada; essa dimensionalidade é determinada pelo valor atribuído a m_1 .

3.3 Estratégias de Aprendizagem

No processo de treinamento de uma RBF, cada uma das entradas x_i , vindas do ambiente externo, serão propagadas pela camada intermediária em direção à camada de saída (Silva

et al., 2010). Além disso, os pesos lineares associados com os neurônios da camada de saída tendem a evoluir em uma “escala de tempo” diferente, comparada às funções de ativação da camada oculta.

A evolução da camada oculta é lenta, de acordo com a estratégia de otimização não-linear, já os pesos da camada de saída se ajustam mais rapidamente devido a sua otimização linear. Com base nessas informações, é razoável separar o processo de otimização das camadas utilizando técnicas diferentes (Haykin, 2008). Existem várias estratégias de aprendizagem que podem ser aplicadas a uma rede RBF, essas estratégias variam dependendo de como os centros são especificados.

3.3.1 Centros Fixos Selecionados ao Acaso

Esta é a abordagem mais simples, onde as funções de base radial são fixas e definem as funções de ativação dos neurônios da camada oculta. Normalmente a localização dos centros é escolhida aleatoriamente a partir do conjunto de dados de treinamento, mas isso só é possível se os dados estiverem bem distribuídos (Haykin, 2008).

Nesse tipo de estratégia, é utilizada como função de base radial a função gaussiana isotrópica, cujo desvio padrão é fixado de acordo com o espalhamento dos centros. Uma função gaussiana cujo centro é t_i pode ser definida como:

$$G(\|x - t_i\|^2) = \exp\left(-\frac{m_1}{d_{max}^2} \|x - t_i\|^2\right) \quad i = 1, 2, \dots, m_1 \quad (3.4)$$

onde m_1 é o número de centros e d_{max} é a distância máxima entre os centros escolhidos. O desvio padrão de todas as funções de base radial gaussianas é fixo em

$$\sigma = \frac{d_{max}}{\sqrt{2m_1}} \quad (3.5)$$

esta equação garante que as funções utilizadas não sejam pontiagudas demais ou planas demais, condições que devem ser evitadas. Nesta abordagem, os únicos parâmetros que devem ser

aprendidos são os pesos lineares na camada de saída (Haykin, 2008), essa fase do processo de aprendizagem será descrita na Seção 3.4.2.

3.3.2 Seleção Auto-Organizada de Centros

De acordo com Haykin (2008), o maior problema do método de centros fixos, descritos na sessão anterior, é a necessidade de um conjunto de treinamento grande para que se obtenha um nível de desempenho satisfatório. Um processo de aprendizagem híbrido é uma maneira de superar essa limitação. Esse processo é dividido em dois estágios (Chen, Mulgrew, & McLaughlin, 1992):

- Estágio de aprendizagem auto-organizada, cujo objetivo é estimar a melhor localização dos centros na camada oculta;
- Estágio de aprendizagem supervisionada, que completa o treinamento estimando os pesos lineares da camada de saída.

Para o processo de aprendizagem auto-organizada, é necessário um algoritmo de agrupamento, "clustering", que subdivida o conjunto fornecido em grupos, tão homogêneo quanto possível.

3.3.3 Seleção Supervisionada de Centros

Nessa abordagem, os centros das funções como todos os outros parâmetros livres são parte do processo de aprendizagem supervisionada da rede, onde a RBF assume sua forma mais generalizada. A abordagem mais utilizada é a aprendizagem por correção do erro (Haykin, 2008). O primeiro passo no desenvolvimento deste procedimento é definir o valor da função de custo (Haykin, 2008)

$$E = \frac{1}{2} \sum_{j=1}^N e_j^2 \quad (3.6)$$

onde N é o tamanho da amostra de treinamento e e_j é o sinal de erro definido por

$$\begin{aligned} e_j &= d_j - F * (x_j) \\ &= d_j - \sum_{i=1}^M w_i G(\|x_j - t_i\|_{ci}) \end{aligned} \quad (3.7)$$

O objetivo é encontrar os parâmetros livres w_i , t_i e Σ_i^{-1} que minimizem E .

3.4 Processo de Treinamento

Conforme mostrado na seção anterior, existem três abordagens que podem ser usadas no processo de treinamento de uma RBF. Para este trabalho, foi escolhida a estratégia de treinamento por seleção auto-organizada dos centros, apresentadas na seção 3.3.2.

Nesta abordagem, o primeiro estágio, associado com o ajuste dos centros e dos pesos da camada intermediária, é feito através de um algoritmo não-supervisionado e é dependente apenas das características dos dados de entrada. Já o segundo estágio, vinculado os pesos sinápticos da camada de saída, utiliza um aprendizado supervisionado.

3.4.1 Primeiro Estágio de Treinamento

Essa primeira fase realiza o ajuste dos neurônios da camada intermediária. Estes neurônios são constituídos de funções de ativação do tipo base radial, podem ser usadas as funções representadas pelas Equações (3.8) a (3.12), mas a função gaussiana é a mais empregada (Silva et al., 2010).

$$\phi(r) = e^{-\frac{r^2}{2\sigma^2}}, \quad \text{gaussiana} \quad (3.8)$$

$$\phi(r) = r^2 \ln(r), \quad \text{thin-plate spline} \quad (3.9)$$

$$\phi(r) = \frac{1}{1 + e^{\frac{r^2}{\sigma^2} - \theta}}, \quad \text{função logística} \quad (3.10)$$

$$\phi(r) = (r^2 + \sigma^2)^{1/2}, \quad \text{multiquádrica} \quad (3.11)$$

$$\phi(r) = \frac{1}{(r^2 + \sigma^2)^{1/2}}, \quad \text{multiquádrica inversa} \quad (3.12)$$

Nessas equações, r denota a distância entre o valor recebido pela camada de entrada x até o centro c e σ^2 representa a variância das amostras, que indica o quão disperso estão os pontos x em relação ao centro da função c .

Considerando a expressão fornecida na Equação (3.8), os parâmetros livres a serem ajustados são o centro c e a variância σ^2 . O centro está diretamente associado aos seus próprios pesos, de acordo com a configuração dos neurônios escondidos da rede. Consequentemente, a saída de cada neurônio da camada intermediária, utilizando a função gaussiana, pode ser expressa por (Silva et al., 2010):

$$\phi(x) = e^{-\frac{\sum_{i=1}^n (x_i - W_{ji}^{(1)})^2}{2\sigma_j^2}} \quad (3.13)$$

O principal objetivo dos neurônios nessa camada é posicionar o centro de suas funções de bases radiais da forma mais apropriada possível. Um dos métodos utilizados para essa função é o algoritmo de *k-means* (Duda et al., 2001). O valor de k nesse algoritmo é igual ao número de neurônios na camada intermediária, pois cada uma das funções de ativação desses neurônios é uma função gaussiana, conforme a Equação (3.13) e os centros serão

representados pelos seus pesos (Silva et al., 2010).

Algoritmo 1: PRIMEIRO ESTÁGIO DE TREINAMENTO

```

1 início
2   Obter o conjunto de amostras de treinamento  $x^{(n)}$ 
3   Iniciar o vetor de pesos de cada neurônio da camada intermediária com os valores
   das  $n_1$  primeiras amostras de treinamento;
4   repita
5     para todas as amostras de treinamento  $x^{(n)}$  faça
6       Calcular as distâncias euclidianas entre  $x^{(n)}$  e  $W_{ji}^{(1)}$ , considerando-se cada
       j-ésimo neurônio por vez;
7       Selecionar o neurônio  $j$ , que contenha a menor distância, com o intuito de
       agrupar a referida amostra junto ao centro mais próximo;
8       Atribuir a amostra  $x^{(n)}$  ao grupo  $\Omega^{(j)}$ ;
9     fim
10    para todos  $W_{ji}^{(1)}$ , onde  $j = 1, \dots, n_1$  faça
11      Ajustar  $W_{ji}^{(1)}$  de acordo com as amostras em  $\Omega^{(j)}$ :  $W_{ji}^{(1)} = \frac{1}{m^{(j)}} \sum_{x^{(n)} \in \Omega^{(j)}} x^{(n)}$ 
      {  $m^{(j)}$  é o número de amostras em  $\Omega^{(j)}$  }
12    fim
13  até não haja mudanças nos grupos  $\omega^{(j)}$  entre as iterações;
14  para todos os  $W_{ji}^{(1)}$ , onde  $j = 1, \dots, n_1$  faça
15    Calcular a variância de cada uma das funções de ativação gaussiana pelo
    critério da distância quadrática média:  $\sigma_j^2 = \frac{1}{m^{(j)}} \sum_{x^{(n)} \in \Omega^{(j)}} \sum_{i=1}^n (x_i^{(n)} - W_{ji}^{(1)})^2$ 
16  fim
17 fim

```

O pseudocódigo, a seguir, foi proposto por Silva et. al (2010) e apresenta a sequência de procedimentos computacionais que visam o treinamento deste primeiro estágio das redes RBF utilizando o algoritmo de *k-means*.

Após finalizado esse estágio de treinamento, os pesos $W^{(1)}$ de cada um dos k neurônios da camada intermediária foram ajustados para seus valores mais adequados para o conjunto de amostras. Esses pesos representam os centros de cada um desses *clusters*, subgrupos de

amostras, que serão usados no próximo estágio de treinamento da rede.

3.4.2 Segundo Estágio de Treinamento

O segundo estágio de treinamento é responsável pelo ajuste dos pesos da camada de saída de uma rede RBF, este estágio é supervisionado e deve ser executado somente após a finalização do primeiro estágio de treinamento (Silva et al., 2010).

Nos experimentos feitos neste trabalho, foram utilizadas duas abordagens diferentes para esta fase. A primeira delas foi a técnica de otimização de pesos que foi baseada no *método de Galerkin* proposta por Haykin (2008) e a segunda foi a utilização da Regra Delta Generalizada proposta por Silva et. al. (2010).

3.4.2.1 Otimização de Pesos - adaptação do método de Galerkin

Essa abordagem procura uma solução subótima em um espaço de menor dimensionalidade. De acordo com (Haykin, 2008) este procedimento foi baseado em uma técnica conhecida como *método de Galerkin*. Para encontrar o vetor de pesos ótimos, entende-se que a solução aproximada que será fornecida pela rede, pode ser generalizada pela equação:

$$\hat{y}(x) = \sum_{i=1}^{m_1} w^{(2)} \phi_i(x) \quad (3.14)$$

onde $\{\phi_i(x) | i = 1, 2, \dots, m_1\}$ é um conjunto de funções de base radial e que $w^{(2)}$ constituem o conjunto de pesos sinápticos da camada de saída. Considerando as funções de base radial expressas nas Equações (3.8) a (3.12), temos:

$$\phi_i(x) = G(\|x - W^{(1)}\|), i = 1, 2, \dots, m_1 \quad (3.15)$$

onde G representa uma dessas funções de base radial e $\|x - W^{(1)}\|$ a distância euclidiana entre o padrão de entrada x e o peso $W^{(1)}$. Substituindo a Equação (3.15) na Equação (3.14), temos:

$$\hat{y}(x) = \sum_{i=1}^{m_1} w^{(2)} G(\|x - W^{(1)}\|) \quad (3.16)$$

O objetivo dessa fase é determinar o novo conjunto de pesos ótimos $w^{(2)}$ que minimizem o funcional de custo $E(\hat{y})$, definido por (Haykin, 2008) como

$$E(\hat{y}) = \sum_{i=1}^N (d_i - \sum_{j=1}^{m_1} w^{(2)} G(\|x - W^{(1)}\|))^2 + \lambda \|D\hat{y}\|^2 \quad (3.17)$$

Segundo (Haykin, 2008) o primeiro termo no lado direito da Equação (3.17) pode ser expresso como a norma euclidiana quadrada $\|d - Gw^{(2)}\|^2$, em que

$$d = [d_1, d_2, \dots, d_N]^T \quad (3.18)$$

$$G = \begin{bmatrix} G(x_1, W_1^{(1)}) & G(x_1, W_2^{(1)}) & \dots & G(x_1, W_{m_1}^{(1)}) \\ G(x_2, W_1^{(1)}) & G(x_2, W_2^{(1)}) & \dots & G(x_2, W_{m_1}^{(1)}) \\ \vdots & \vdots & & \vdots \\ G(x_N, W_1^{(1)}) & G(x_N, W_2^{(1)}) & \dots & G(x_N, W_{m_1}^{(1)}) \end{bmatrix} \quad (3.19)$$

$$w^{(2)} = [w_1^{(2)}, w_2^{(2)}, \dots, w_N^{(2)}]^T \quad (3.20)$$

onde d é o vetor de resposta desejado, G é a matriz que contém o conjunto de funções de base radiais (camada intermediária) e $w^{(2)}$ é o vetor de pesos sinápticos para a camada de saída. Verificamos que a matriz G têm dimensões N -por- m_1 e portanto não é simétrica.

A partir da Equação (3.16) notamos que a função aproximativa \hat{y} é uma combinação linear das funções de Green ¹ para o estabilizador D . Por isso, podemos expressar o segundo

¹Uma Função de Green é um tipo de função utilizada para resolver equações diferenciais não-homogêneas sujeitas a condições iniciais ou condições de contorno determinadas

termo do lado direito da Equação (3.17) como:

$$\begin{aligned}
\|D\hat{y}\|^2 &= (D\hat{y}, D\hat{y})_\Omega \\
&= \left[\sum_{i=1}^{m_1} w_i^{(2)} G(x, W_i^{(1)}), \tilde{D} D \sum_{i=1}^{m_1} w_i^{(2)} G(x, W_i^{(1)}) \right]_\Omega \\
&= \left[\sum_{i=1}^{m_1} w_i^{(2)} G(x, W_i^{(1)}), \sum_{i=1}^{m_1} w_i^{(2)} \delta W_i^{(1)} \right]_\Omega \\
&= \sum_{j=1}^{m_1} \sum_{i=1}^{m_1} w_j^{(2)} w_i^{(2)} G(W_j^{(1)}, W_i^{(1)}) \\
&= \mathbf{w}^T \mathbf{G}_0 \mathbf{w}
\end{aligned} \tag{3.21}$$

Na segunda e terceira linha da demonstração foi feito uso equação para diferenciação parcial das funções de Green (Haykin, 2008). A matriz \mathbf{G}_0 é simétrica de dimensão m_1 -por- m_1 , definida por:

$$\mathbf{G}_0 = \begin{bmatrix} G(W_1^{(1)}, W_1^{(1)}) & G(W_1^{(1)}, W_2^{(1)}) & \cdots & G(W_1^{(1)}, W_{m_1}^{(1)}) \\ G(W_2^{(1)}, W_1^{(1)}) & G(W_2^{(1)}, W_2^{(1)}) & \cdots & G(W_2^{(1)}, W_{m_1}^{(1)}) \\ \vdots & \vdots & & \vdots \\ G(W_{m_1}^{(1)}, W_1^{(1)}) & G(W_{m_1}^{(1)}, W_2^{(1)}) & \cdots & G(W_{m_1}^{(1)}, W_{m_1}^{(1)}) \end{bmatrix} \tag{3.22}$$

A minimização da Equação (3.17) em relação ao vetor de peso \mathbf{w} pode ser representada como:

$$(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{G}_0) \mathbf{w} = \mathbf{G}^T \mathbf{d} \tag{3.23}$$

resolvendo para \mathbf{w} :

$$\mathbf{w} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{G}_0)^{-1} \mathbf{G}^T \mathbf{d} \tag{3.24}$$

Se o parâmetro de regularização λ se aproxima de zero, o vetor e peso \mathbf{w} converge para a solução da pseudo-inversa do problema indeterminado de ajuste de dados por quadrados mínimos para $m_1 < N$, a matriz pseudo-inversa \mathbf{G}^+ da matriz \mathbf{G} pode ser definida como (Broomhead & Lowe, 1988):

$$\mathbf{G}^+ = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \tag{3.25}$$

Substituindo Equação (3.25) em Equação (3.24):

$$\mathbf{w} = \mathbf{G}^+ \mathbf{d}, \lambda = 0 \quad (3.26)$$

Com base na Equação (3.26) conclui-se que essa abordagem necessita somente de uma iteração para definição dos pesos ótimos \mathbf{w} que melhor irão se adaptar ao conjunto de dados inseridos no treinamento.

O pseudocódigo abaixo apresenta a sequência de procedimentos computacionais que visam o treinamento deste segundo estágio das redes RBF utilizando a técnica proposta por (Haykin, 2008).

Algoritmo 2: SEGUNDO ESTÁGIO DE TREINAMENTO - OTIMIZAÇÃO DE PESOS

```

1  início
2  |   Obter o conjunto de amostras de treinamento  $\{x\}$ 
3  |   Obter o conjunto de pesos  $\{W^{(1)}\}$ 
4  |   Obter o conjunto de saídas desejadas  $\{d\}$ 
5  |   para todas as amostras de treinamento  $x$  faça
6  |       |   Calcular os valores da matriz  $G$  de acordo com Equação (3.19)
7  |   fim
8  |   Calcular vetor de peso ótimo:  $\mathbf{w} = \mathbf{G}^+ \mathbf{d}$ 
9  fim

```

3.4.2.2 Regra Delta Generalizada

A segunda abordagem, para o segundo estágio de treinamento, proposta por Silva et al. (2010) utiliza a Regra Delta Generalizada e é portanto, supervisionada. De acordo com a Figura 3.1, os dados são propagados pela rede saindo da camada de entrada, passando pela camada intermediária, onde foi realizado o primeiro estágio de treinamento, e seguem para camada de saída.

O conjunto de treinamento para os neurônios da camada de saída será constituído por pares de entrada e saída, em que as entradas serão as respostas produzidas pelas funções de

ativação de bases radiais, como as expressas nas Equações (3.8) a (3.12), dos neurônios da camada intermediária e as saídas são as aquelas esperadas pela rede (Silva et al., 2010). Este conjunto é representado pela equação

$$\hat{y} = \sum_{i=1}^{n_1} w_{ji}^{(2)} \cdot \phi(x, W^{(1)}) - \theta_j, \text{ onde } j = 1, \dots, n_2 \quad (3.27)$$

onde $w_{ji}^{(2)}$ representa o vetor de pesos sinápticos e θ o limiar, ambos da camada de saída, porém a expressão $\phi(x, W^{(1)})$ representa os valores calculados pela função de ativação de base radial utilizando os pesos obtidos no primeiro estágio de treinamento.

Por se tratar de um treinamento supervisionado, cada padrão de entrada tem uma unidade de saída correspondente, representada pelo vetor-alvo. O erro proveniente da diferença entre os valores encontrados pela rede e a saída correspondente é utilizado para a atualização dos pesos $w^{(2)}$. A equação que representa esse erro é expressada por

$$E = \frac{1}{2} \cdot \sum_k \left(d_k - \hat{y}_k \right)^2 \quad (3.28)$$

A forma de ajuste do vetor de pesos $w^{(2)}$ para obtenção de seu valor ótimo pode ser expressa pelas equações:

$$z = \phi(x, W^{(1)}) \quad (3.29)$$

$$yin = z \cdot w^{(2)} \quad (3.30)$$

$$\delta_k = (d_k - \hat{y}_k) \cdot f(yin) \quad (3.31)$$

$$\Delta w^{(2)} = \alpha \cdot \delta_k \cdot z \quad (3.32)$$

em que:

- $\phi(x, W^{(1)})$: valores calculados pela função de ativação de base radial no primeiro estágio de treinamento;
- $w^{(2)}$: pesos sinápticos da camada de saída

- f : função de ativação dos neurônios da camada de saída.
- d_k : saída desejada
- \hat{y} : resultado encontrado pela rede
- α : taxa de aprendizagem da rede

Em sua modelagem tradicional, a função de ativação utilizada pelos neurônios da camada de saída é sempre linear. Nesse caso, os neurônios dessa camada somente irão realizar uma combinação linear das funções de ativação de base radial produzidas na camada anterior (Silva et al., 2010).

As instruções do pseudo-código proposta foram por (Silva et al., 2010) visando-se o segundo estágio de treinamento, são explicitadas como segue:

Algoritmo 3: SEGUNDO ESTÁGIO DE TREINAMENTO - REGRA DELTA GENERALIZADA

```

1  início
2  | Obter o conjunto de amostras de treinamento  $\{x\}$ 
3  | Obter o vetor de saídas desejadas  $\{d\}$  para cada amostra
4  | Iniciar  $w_{ji}^{(2)}$  com valores aleatórios  $[-0.5, 0.5]$ 
5  | Especificar a taxa de aprendizagem  $\alpha$  e precisão requerida  $\epsilon$ 
6  | para todas as amostras de treinamento  $x^{(k)}$  faça
7  | | Calcular  $\phi_j(x, W^{(1)})$  de acordo com Equação (3.15)
8  | | Assumir  $z^{(k)} = [\phi_1, \phi_2, \dots, \phi_n]^T$ 
9  | fim
10 | Iniciar o contador do número de épocas:  $epoca \leftarrow 0$ ;
11 | repita
12 | |  $E_{anterior} \leftarrow E$  conforme Equação (3.28)
13 | | para todos os pares de treinamento  $\{z^{(k)}, d^{(k)}\}$  faça
14 | | | Ajustar  $w_{ji}^{(2)}$  conforme especificado nas Equações (3.29) a (3.32)
15 | | | fim
16 | |  $E_{atual} \leftarrow E$  conforme Equação (3.28)
17 | |  $epoca \leftarrow epoca + 1$ 
18 | até  $E_{atual} - E_{anterior} \leq \epsilon$ ;
19 fim

```

3.4.2.3 Modificação na camada de saída

A partir da implementação baseada na Regra Delta Generalizada, foi feita uma modificação no neurônio da camada de saída da rede. Tradicionalmente, estes neurônios possuem uma função de ativação linear e ela foi substituída por uma função sigmoide, como a tangente hiperbólica e a logística bipolar, descritas na Tabela 2.1. A Figura 3.3 demonstra a arquitetura da rede após a modificação realizada.

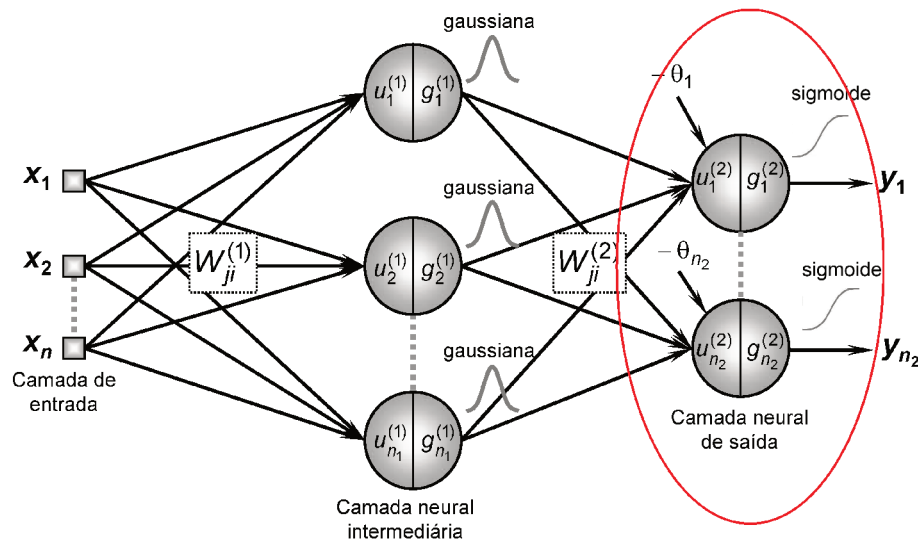


Figura 3.3: Arquitetura de uma rede do tipo RBF modificada

Fonte: (Silva et al., 2010) - modificada

O processo de treinamento realizado nesse modelo foi igual ao apresentado na Seção 3.4.2.2, sendo realizadas apenas as modificações referentes à função de ativação. Essa alteração na camada de saída, proporcionou uma nova transformação nesses dados, onde antes havia apenas uma combinação linear.

Os resultados experimentais demonstraram que essa alteração aumentou o desempenho da rede em cerca de 9% para os dígitos australianos e cerca de 3% nos demais conjuntos de dados testados. Todos os experimentos realizados utilizando a Regra Delta Generalizada, apresentados no Capítulo 5, foram feitos com a rede já modificada.

Capítulo 4

Alvos ortogonais em Redes de Funções

Bases Radiais

4.1 Definição de vetores-alvo

Convencionalmente, dois tipos de vetores-alvo são usados pelas RNAs em problemas de reconhecimento de padrões, o Vetor Binário (VBN) e o Vetor Bipolar Convencional (VBC). A proposta deste trabalho é a utilização de vetores-alvo que possuem a característica de serem mutuamente ortogonais e dimensão sempre equivalente a uma potência de 2, definida dessa forma. Por questões inerentes ao algoritmo de geração, determinado pela Seção 4.2.

Por isso, em aplicações, o Vetor Bipolar Ortogonal (VBO) pode apresentar dimensão maior que os vetores convencionais. Para fins de comparação, também foi utilizado vetores com a mesma característica dos vetores convencionais, tendo a mesma dimensão dos vetores ortogonais, definidos no trabalho como Vetor Não Ortogonal (VNO). Os tipos de vetores-alvo são definidos como:

- *Vetores Binários (VBN)*: são vetores constituídos por n componentes, onde n corresponde à quantidade de padrões a serem classificados. Representados por uma matriz, definida pela Equação (4.1), cada linha i desta matriz corresponde ao i -ésimo VBN contendo o componente “1” para $i = j$ e o componente “0” para os

outros elementos.

$$\vec{V}_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \quad (4.1)$$

- *Vetores Bipolares Convencionais (VBC)*: são constituídos por n componentes e sua dimensão depende da quantidade de padrões a serem classificados, de forma semelhante aos VBN. Porém, na representação matricial, definida pela Equação (4.2), contém os componentes “1” e “-1”.

$$\vec{V}_{ij} = \begin{cases} 1 & \text{for } i = j \\ -1 & \text{for } i \neq j \end{cases} \quad (4.2)$$

- *Vetores Bipolares Ortogonais (VBO)*: são caracterizados por serem mutuamente ortogonais e por questões matemáticas inerentes ao algoritmo de geração, seu tamanho é sempre uma potência de 2. Para geração dos VBOs, é usado o teorema apresentado na Seção 4.2.
- *Vetores Não Ortogonais (VNO)*: possuem a mesma característica dos vetores do tipo VBCs e a mesma dimensão dos VBOs. Eles foram utilizados neste trabalho apenas com o objetivo de propiciar uma comparação justa de vetores. Para obter VNOs, os VBCs são complementados com o termo “-1” de modo a atingir o mesmo tamanho dos VBOs.

4.2 Algoritmo de geração de vetores bipolares ortogonais

O método de geração de VBO foi proposto por Fausett & Hall (1994), nesse algoritmo um vetor, chamado vetor de sementes, é utilizado para gerar uma nova sequência de vetores até a determinação final dos VBOs. O número de componentes de cada VBO depende do número de vetores desejados e do número de componentes escolhido para o vetor de sementes. Essa relação pode ser descrita pela fórmula

$$n = 2^k m \quad (4.3)$$

onde 2^k é o número de vetores ortogonais entre si para $k > 0$ e m é o número de componentes em um vetor de sementes. É interessante observar que o número de vetores será sempre uma potência de base 2. Logicamente, o interessado constrói seu conjunto, observando esse detalhe, e, após a obtenção dos vetores, seleciona somente a quantidade de que necessitar.

Assim sendo, um conjunto com 2^k VBOs é construído com $2^k m$ componentes. O pseudo-código que apresenta a sequência de passos que visam a geração dos vetores-alvos ortogonais está descrita no Algoritmo 4.

Algoritmo 4: GERAÇÃO DE VETORES BIPOLARES ORTOGONAIS

```

1  início
2  | Iniciar o valor de  $\{m\}$  {número de componentes do vetor de sementes}
3  | Iniciar o valor de  $\{k\}$  {número vetores ortogonais a serem gerados}
4  | Iniciar o vetor de sementes  $V$  de acordo com a equação  $V_m^0 = \overbrace{(1, 1, \dots, 1)}^m$ 
5  | Concatenar o vetor  $V$  para construção de novos vetores:
6  | início
7  | | A função de concatenação é definida por:
8  | |  $f_{cc}(U, W) = (u_1, u_2, \dots, u_n, w_1, w_2, \dots, w_n)$ 
9  | | São gerados os vetores:  $V_{2m}^1 = f_{cc}(V_m^0, V_m^0)$  e  $V_{2m}^2 = f_{cc}(V_m^0, -V_m^0)$ 
10 | fim
11 | Concatenar os vetores obtidos na linha 9 para construção de novos vetores:
12 | início
13 | | Os vetores construídos são:
14 | |  $V_{4m}^1 = f_{cc}(V_{2m}^1, V_{2m}^1)$ ,  $V_{4m}^2 = f_{cc}(V_{2m}^1, -V_{2m}^1)$ ,
15 | |  $V_{4m}^3 = f_{cc}(V_{2m}^2, V_{2m}^2)$  e  $V_{4m}^4 = f_{cc}(V_{2m}^2, -V_{2m}^2)$ 
16 | fim
17 | repita
18 | | Concatenar os vetores existentes para a construção de novos vetores.
19 | até que exista  $2^k$  vetores ortogonais;
20 fim

```

Para fins ilustrativos, considere o exemplo em que se deseja construir 8 VBOs. O vetor semente escolhido contém um componente $\{V_m^0 = (1)\}$, dessa forma, de acordo com a

Equação (4.3) temos 8 componentes em cada vetor, conforme demonstrado

$$n = 2^k m = 8 \cdot 1 = 8 \quad (4.4)$$

O próximo passo do algoritmo é a concatenação do vetor de sementes para geração de novos vetores, que são representados por:

$$V_{2m}^1 = fcc(V_m^0, V_m^0) = (1, 1) \quad (4.5)$$

$$V_{2m}^1 = fcc(V_m^0, -V_m^0) = (1, -1) \quad (4.6)$$

Em seguida, um novo conjunto de vetores é obtido pela execução da linha 11 do algoritmo, esses vetores são definidos como:

$$V_{4m}^1 = fcc(V_{2m}^1, V_{2m}^1) = (1, 1, 1, 1) \quad (4.7)$$

$$V_{4m}^2 = fcc(V_{2m}^1, -V_{2m}^1) = (1, 1, -1, -1) \quad (4.8)$$

$$V_{4m}^3 = fcc(V_{2m}^2, V_{2m}^2) = (1, -1, 1, -1) \quad (4.9)$$

$$V_{4m}^4 = fcc(V_{2m}^2, -V_{2m}^2) = (1, -1, -1, 1) \quad (4.10)$$

Seguindo o algoritmo, em uma possível execução da linha 18, obtêm-se os vetores

representados:

$$V_{8m}^1 = f_{cc}(V_{4m}^1, V_{4m}^1) = (1, 1, 1, 1, 1, 1, 1, 1) \quad (4.11)$$

$$V_{8m}^2 = f_{cc}(V_{4m}^1, -V_{4m}^1) = (1, 1, 1, 1, -1, -1, -1, -1) \quad (4.12)$$

$$V_{8m}^3 = f_{cc}(V_{4m}^2, V_{4m}^2) = (1, 1, -1, -1, 1, 1, -1, -1) \quad (4.13)$$

$$V_{8m}^4 = f_{cc}(V_{4m}^2, -V_{4m}^2) = (1, 1, -1, -1, -1, -1, 1, 1) \quad (4.14)$$

$$V_{8m}^5 = f_{cc}(V_{4m}^3, V_{4m}^3) = (1, -1, 1, -1, 1, -1, 1, -1) \quad (4.15)$$

$$V_{8m}^6 = f_{cc}(V_{4m}^3, -V_{4m}^3) = (1, -1, 1, -1, -1, -1, 1, -1) \quad (4.16)$$

$$V_{8m}^7 = f_{cc}(V_{4m}^4, V_{4m}^4) = (1, -1, -1, 1, 1, -1, -1, 1) \quad (4.17)$$

$$V_{8m}^8 = f_{cc}(V_{4m}^4, -V_{4m}^4) = (1, -1, -1, 1, -1, -1, 1, 1) \quad (4.18)$$

Se for realizado o produto interno tomando cada par de vetores, verificar-se-á que serão iguais a zero. Dessa forma, são obtidos oito VBOs, e cada qual com oito componentes.

4.3 Observações sobre os vetores

A distância euclidiana entre dois VBCs é igual a $2\sqrt{2}$ independente de sua dimensão, porém a distância euclidiana entre dois VBOs aumenta à medida que a dimensão desses vetores também aumenta. Por essa característica, para vetores de dimensão n , metade nas n diferenças se anulará e a outra metade será igual a $2^2 = 4$. Por isso, a Equação (2.4) pode ser reescrita como:

$$d_{U,W} = \sqrt{\frac{n}{2} \cdot 4} = \sqrt{2n} \quad (4.19)$$

O gráfico da Figura 4.1 mostra a evolução da distância euclidiana baseado no aumento da dimensão dos vetores-alvo. Com VBC e VNO, a distância vale sempre $\sqrt{2}$ e para VBO, a

distância é sempre crescente. Conforme demonstrado pelas Equações (4.20) a (4.23).

$$\lim_{x \rightarrow \infty} \sqrt{2n} = \lim_{x \rightarrow \infty} \sqrt{2} \cdot \sqrt{n} \quad (4.20)$$

$$\lim_{x \rightarrow \infty} \sqrt{2n} = \sqrt{2} \cdot \lim_{x \rightarrow \infty} \sqrt{n} \quad (4.21)$$

$$\lim_{x \rightarrow \infty} \sqrt{2n} = \sqrt{2} \cdot \infty \quad (4.22)$$

$$\lim_{x \rightarrow \infty} \sqrt{2n} = \infty \quad (4.23)$$

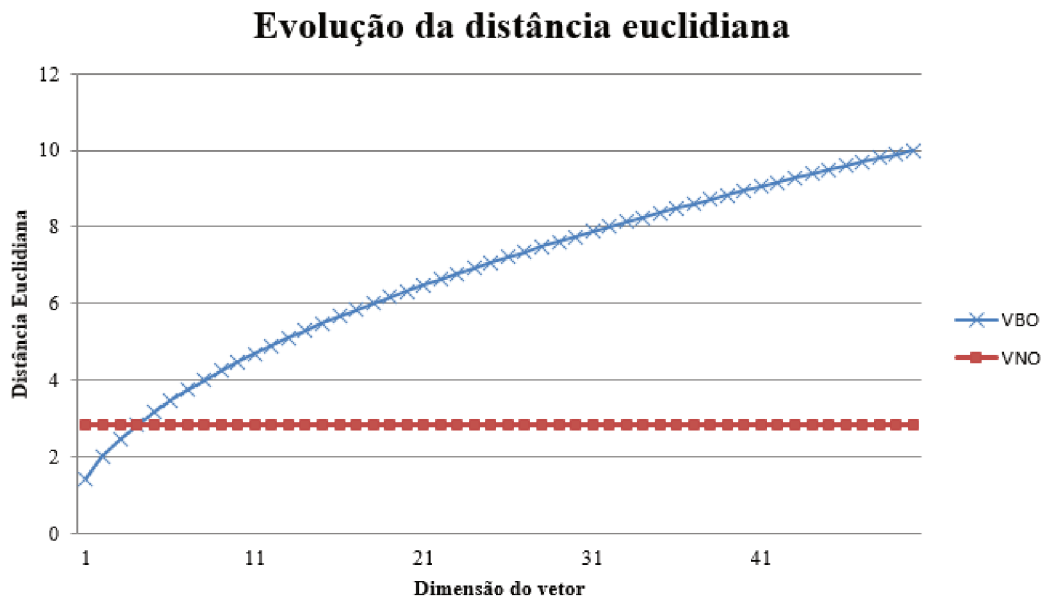


Figura 4.1: Distância Euclidiana de vetores-alvo

4.4 Efeito dos alvos ortogonais no processo de treinamento

Conforme visto nas seções anteriores, os VBOs alteram o espaço de saída da rede, deixando-o mais disperso pois aumenta a distância entre um alvo e outro. Porém, sua eficácia em melhorar o desempenho de uma rede, depende da técnica utilizada para realização dos estágios de treinamento.

4.4.1 Técnica de Otimização dos pesos

Conforme descrito na Seção 3.4.2.1, nessa abordagem a camada de saída da rede é linear e calculada através do uso da matriz pseudo-inversa dos resultados das funções de base radial utilizadas no primeiro estágio de treinamento. Este trabalho utilizou como função de base radial, as funções gaussianas, definidas após vários experimentos, pois foram as que apresentaram as melhores taxas de reconhecimento dos padrões.

A função gaussiana é definida pela Equação (3.8) e é representada graficamente pela Figura 4.2. Nela a variável σ representa o valor do desvio padrão das amostras, de cada sub-grupo, essa é uma medida de dispersão que indica a regularidade de um conjunto de dados em função da média. Nas gaussianas, ela define a amplitude da função, conforme mostrado na Figura 4.2.

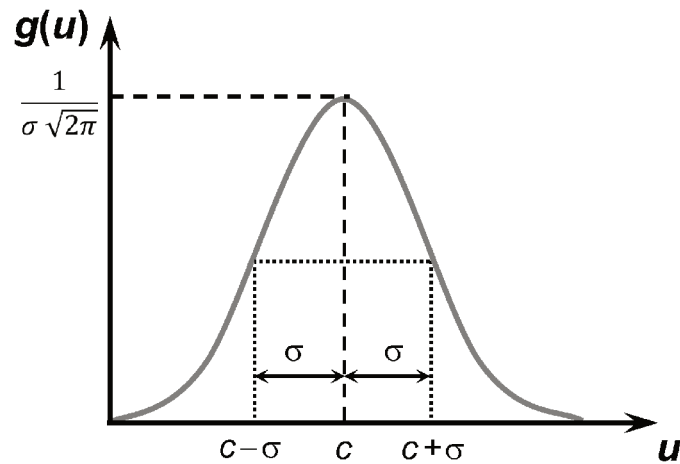


Figura 4.2: Função de base radial do tipo gaussiana
Fonte: (Silva et al., 2010)

A variância σ^2 também é uma medida de dispersão e indica “o quão longe” os valores se encontram do valor esperado, a Equação (4.24) define uma variância da população de um conjunto de dados. O desvio padrão é obtido através do cálculo da raiz quadrada positiva da variância.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (4.24)$$

onde N é a quantidade de padrões e μ representa a média.

No processo de RP, as fronteiras delimitadoras das RBF são definidas por campos receptivos hipersféricos. Consequentemente, a classificação de padrões levará em conta a distância radial em que os os padrões se encontram em relação aos centros das hiperesferas. Já a MLP computa as fronteiras de delimitação por intermédio de uma combinação de hiperplanos, conforme exemplificado pela Figura 4.5 (Silva et al., 2010).

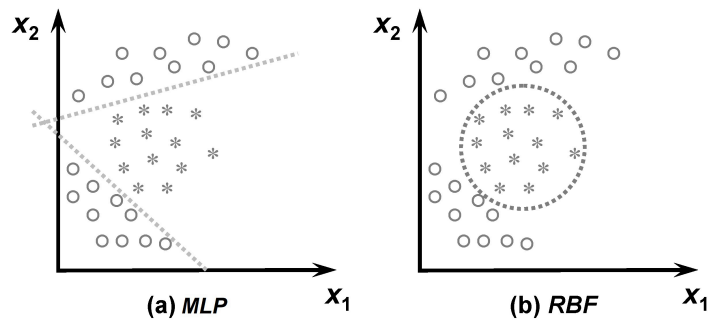


Figura 4.3: Fronteira de separabilidade das redes MLP e RBF

Fonte: (Silva et al., 2010)

Com base nessas características, quando utilizado um VBO no processo de classificação, os alvos ficarão mais distantes, porém durante o cálculo da gaussiana para o processo de clusterização, essa dispersão será refletida no valor da variância das amostras. Com os dados dispersos, a variância das amostras será maior, aumentando o desvio padrão e consequentemente a amplitude da gaussiana também será maior para se ajustar aos dados pertencentes a essa distribuição, o que não terá nenhum impacto no processo de clusterização.

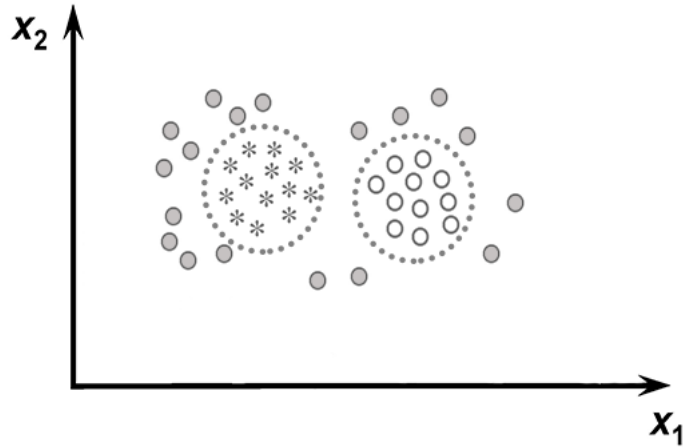


Figura 4.4: Clusterização de amostras com VNO

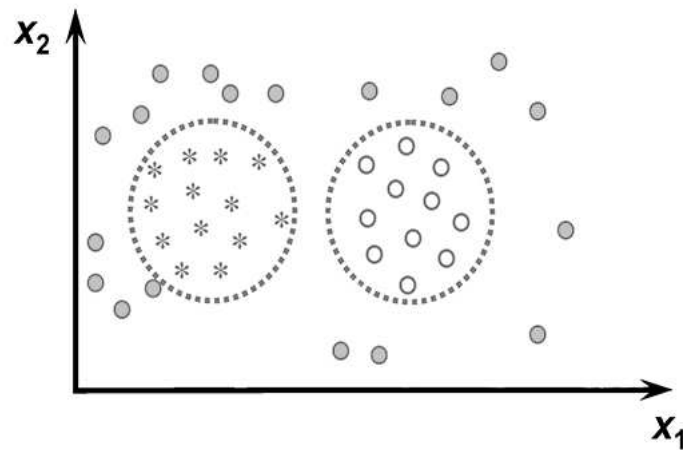


Figura 4.5: Clusterização de amostras com VBO

A segunda fase do treinamento nessa abordagem é linear, utilizando os valores obtidos nos cálculos das gaussianas para se obter o vetor de pesos sinápticos otimizado para o conjunto de dados treinado. Portanto, conclui-se que a utilização de outros vetores-alvo, nessa abordagem, não altera o desempenho da rede.

4.4.2 Regra Delta Generalizada

O algoritmo da Regra Delta é deduzido por meio da função erro obtida pela diferença entre a saída encontrada e a saída desejada (Fausett & Hall, 1994). Foi deduzido por Rumelhart et al (1985) com o objetivo de se reduzir o valor do erro quadrático médio. Assim, a função erro

é submetida a um algoritmo de gradiente descendente.

A comprovação matemática do uso dos VBOs no algoritmo da Regra Delta foi proposta por Manzan (2016). Para uma discussão matemática apropriada, é necessário fornecer algumas fórmulas.

Considere x_i como o valor da i -ésima entrada, d_k como o valor da k -ésima saída desejada, z_j como o j -ésimo valor que chega à camada oculta e y_k como a k -ésima saída encontrada. Considere $W_{ij}^{(1)}$ como o peso sináptico entre a i -ésima entrada e o j -ésimo neurônio da camada oculta, $w_{ij}^{(2)}$ como o peso sináptico entre o j -ésimo neurônio da camada oculta e o k -ésimo neurônio da camada de saída e w_{0ij} como o peso sináptico do tipo bias correspondente ao k -ésimo neurônio da camada de saída. A Equação (4.25) mostra o cálculo de z , nela ϕ representa a função de base radial utilizada no primeiro estágio de treinamento e as Eq. 4.26 e 4.27 mostram o cálculo de y_k (Fausett & Hall, 1994).

$$z_j = \phi(x_i, W^{(1)}) \quad (4.25)$$

$$yin_k = \sum_j z_j w_{jk}^{(2)} \quad (4.26)$$

$$y_k = f(yin_k) \quad (4.27)$$

A função erro é mostrada pela Equação (4.28), na qual d_k é a saída desejada e y_k é a saída encontrada. As Equações (4.29) a (4.33) mostram o cálculo do gradiente descendente da função erro em relação aos pesos da camada de saída. A Equação (4.34) mostra a fórmula de

atualização dos pesos da camada de saída.

$$E = \frac{1}{2} \sum_k [d_k - y_k]^2 \quad (4.28)$$

$$\frac{\partial E}{\partial w_{jk}^{(2)}} = \frac{\partial}{\partial w_{jk}^{(2)}} \frac{1}{2} \sum_k [d_k - y_k]^2 \quad (4.29)$$

$$\frac{\partial E}{\partial w_{jk}^{(2)}} = \frac{\partial}{\partial w_{jk}^{(2)}} \frac{1}{2} [d_k - f(y_{in_k})]^2 \quad (4.30)$$

$$\frac{\partial E}{\partial w_{jk}^{(2)}} = -[d_k - y_k] \frac{\partial}{\partial w_{jk}^{(2)}} f(y_{in_k}) \quad (4.31)$$

$$\frac{\partial E}{\partial w_{jk}^{(2)}} = -[d_k - y_k] f'(y_{in_k}) \frac{\partial}{\partial w_{jk}^{(2)}} (y_{in_k}) \quad (4.32)$$

$$\frac{\partial E}{\partial w_{jk}^{(2)}} = -[d_k - y_k] f'(y_{in_k}) z_j \quad (4.33)$$

$$\delta_k = [d_k - y_k] f'(y_{in_k}) \quad (4.34)$$

É possível expressar matematicamente a função distância euclidiana em relação à função erro. Se a distância euclidiana diminui, o erro também diminui. Se a distância euclidiana aumenta, o erro também aumenta. A função distância euclidiana é mostrada pela Eq. 4.35.

$$D = \left(\sum_k [t_k - y_k]^2 \right)^{\frac{1}{2}} \quad (4.35)$$

Assim, a função distância euclidiana é a raiz quadrada do dobro da função erro $D = \sqrt{2E}$. Aplicando o gradiente descendente para a função distância euclidiana, obtém-se

$$\delta_k = \frac{[t_k - y_k] f'(y_{in_k})}{\sqrt{\sum_k [t_k - y_k]^2}} \quad (4.36)$$

$$\delta_j = \sum_k \delta_k w_{jk} f'(z_{in_j}) \quad (4.37)$$

Assim, as fórmulas obtidas pela aplicação do gradiente descendente para a função erro também estão relacionadas com as fórmulas obtidas pela aplicação do gradiente descendente para a função da distância euclidiana. Pode-se deduzir que, se os pontos do espaço de saída, intitulados “alvos” da rede, estão mais distantes uns dos outros, as chances de uma saída obtida pela inserção de um determinado padrão se aproximar de alvos correspondentes a outros padrões são menores.

Os pontos do espaço de saída são equidistantes para qualquer tipo de vetor no espaço R^n , são considerados dois vetores-alvo distintos. Considere-se também a representação desses pontos no plano. Ao longo do treinamento, cada saída obtida é projetada no espaço R^n . Nessa ilustração, cada saída é projetada no plano. Com a evolução dos ciclos, as saídas projetadas formam uma região de convergência em torno da saída desejada. A Figura 4.6 ilustra essa discussão.

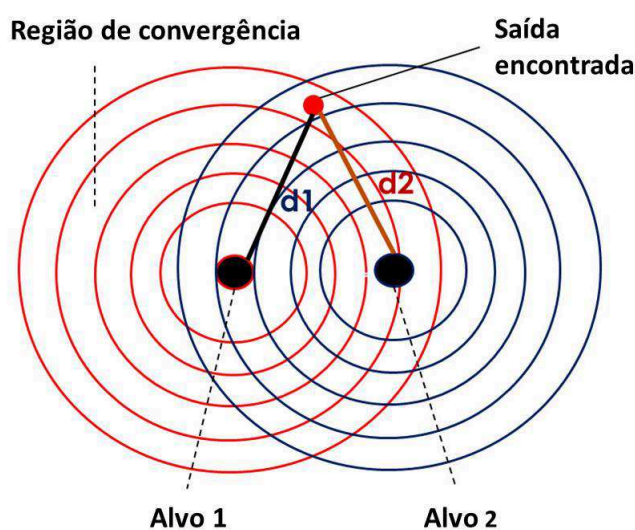


Figura 4.6: Ilustração de regiões de convergência
Fonte: (Manzan, 2016)

Nessa ilustração está clara a existência de uma grande quantidade de pontos pertencentes às duas regiões de convergência. Esses pontos dentro da intersecção das regiões de convergência estão mais propensos a estarem mais próximos de alvos correspondentes a outros padrões, ou seja, mais próximos de alvos incorretos, fazendo com que a rede os classifique

incorretamente.

Contudo, se os alvos estão mais distantes uns dos outros, a intersecção entre as regiões de convergência é bem menor. Assim, a taxa de classificação é beneficiada, porque há menos saídas propensas a erros de classificação. A Figura 4.7 ilustra dois alvos dispostos a uma maior distância euclidiana.

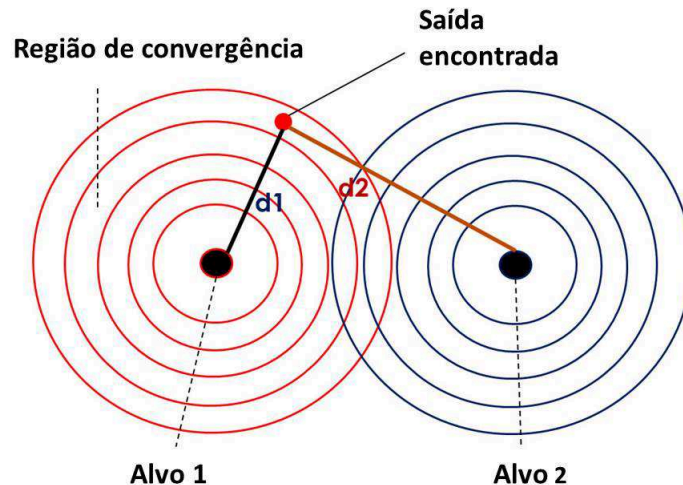


Figura 4.7: Ilustração de regiões de convergência com alvos distantes

Fonte: (Manzan, 2016)

Isso explica por que as redes que utilizam essa metodologia treinadas com VBOs têm melhor desempenho global. Também explica a superioridade de VBOs com poucos ciclos de treinamento. A característica dos VBOs de possuírem maior distância euclidiana reduz a interferência entre as regiões de convergência.

Capítulo 5

Procedimentos Experimentais

5.1 Dados experimentais

Os experimentos realizados para treinamento e teste da RBF foram feitos com três conjuntos de dados diferentes disponibilizados por repositório de dados públicos. Em cada um dos conjuntos de dados, foram feitos testes com diferentes tipos de vetores-alvos e dimensões.

5.1.1 Íris humana

O conjunto de dados para os experimentos realizados com a íris humanas foi obtido na Chinese Academy of Sciences - Institute of Automation database denominada CASIA (Casia, 2010). O conjunto de dados contém íris de 108 indivíduos e para 70 deles há um conjunto completo de sete imagens. De acordo com o repositório CASIA, essas imagens foram obtidas com o uso de luz infravermelha para obter melhor contraste e nitidez.

Cada imagem possui 18 círculos concêntricos, e o primeiro passo para o processamento das imagens é a detecção das circunferências que foi feita utilizando a transformada de Hough Circular (Pereira & Veiga, 2005). Em seguida, a região da íris, que tem formato de anel, é normalizada de modo a ser representada como uma matriz retangular.

A extração das características da íris é realizada no próximo passo, neste trabalho, foi realizada por convolução da imagem normalizada com o chamado filtro de Log Gabor. A

filtragem resulta em coeficientes complexos, cujas fases são quantizadas para um dos quatro quadrantes do plano complexo. Cada quadrante é referenciado por dois bits, e um modelo binário é criado (Daugman, 1993; Negin et al., 2000; Manzan, Yamanaka, & Nomura, 2011).

Para cada imagem existem 8640 pixels dispostos em 18 círculos concêntricos, cada um contendo 480 pixels. Porém, para reduzir a interferência dos cílios, foi utilizado somente as cinco primeiras circunferências. Dessa forma, cada um dos padrões de treinamento corresponde a um conjunto de $5 \times 480 = 2400$ pixels, que são dispostos em um vetor linha onde os pixels brancos são representados por “-1”, e os pixels pretos, por “1”.

5.1.2 Dígitos manuscritos

O repositório internacional Semeion Handwritten Digit of Machine Learning Repository (Lichman, 2013) disponibiliza o conjunto de dados com imagens escaneadas de dígitos manuscritos. Segundo informações disponibilizadas pelo repositório, para obtenção desses dados, cerca de 80 pessoas foram convidadas a escrever os números de 0 a 9 duas vezes:

1. Escrever os dígitos calmamente, primando pela perfeição;
2. Escrever os dígitos rapidamente, sem se preocupar com a legibilidade.

Cada figura foi escaneada em uma imagem contendo 256 pixels no formato de 16 linhas e 16 colunas, essa matriz foi transformada em um vetor linha de 256 componentes, sendo que cada linha foi posicionada imediatamente à direita de sua linha superior na matriz (Lichman, 2013). Para cada pixel correspondente ao fundo da imagem foi atribuído o valor -1 e para cada pixel correspondente à imagem foi atribuído o valor 1.

5.1.3 Signos da linguagem de sinais australiana

Finalmente, foram realizados experimentos com signos da linguagem de sinais australiana (Australian linguagem gestual) obtidos no UCI Machine Learning Repository (Lichman, 2013). De acordo com o repositório, foram capturadas 27 amostras de cada um dos 95 signos australianos usando-se rastreadores de posição de alta qualidade de indivíduos nativos.

Os dados foram capturados utilizando-se luvas de quinta dimensão, rastreadores magnéticos, entre outros equipamentos de alta qualidade. Porém alguns dados obtidos não são totalmente precisos. Foram registrados os seguintes dados:

- *Posição x*: posição relativa, medida em metros, em relação a um ponto ajustado ligeiramente abaixo do queixo.
- *Posição y*: posição relativa, em metros, em relação a um ponto ajustado ligeiramente abaixo do queixo.
- *Posição z*: posição relativa, em metros, em relação a um ponto ajustado ligeiramente abaixo do queixo.
- *Roll* $[-0.5, 0.5]$: 0 sendo palma para baixo, valor positivo significa que a palma gira no sentido horário a partir da perspectiva do pronunciador.
- *Pitch* $[-0.5, 0.5]$: 0 sendo palma plana (horizontal), um valor positivo significa que a palma está apontando para cima.
- *Yaw* $[-1, 1]$: 0 corresponde a palma para a frente e o valor positivo significa movimento no sentido horário ambos a partir da perspectiva acima do pronunciador.
- *Medida da curva do polegar* $[0, 1]$: O valor 0 significa totalmente plana, e 1 significa totalmente dobrado.
- *Medida da curva do dedo indicador* $[0, 1]$: 0 indica o dedo totalmente plano e 1 indica que o dedo está totalmente dobrado.
- *Medida da curva do dedo médio* $[0, 1]$: 0 indica o dedo totalmente plano e o valor 1 indica o dedo totalmente dobrado.
- *Medida da curva do dedo anelar* $[0, 1]$: 0 indica o dedo totalmente plano e o valor 1 indica o dedo totalmente dobrado.
- *Medida da curva do dedo mindinho* $[0, 1]$: 0 indica o dedo totalmente plano, e o valor 1 indica o dedo totalmente dobrado.

5.2 Vetores-alvo

Para cada conjunto de dados testados, foram usados os 3 tipos de vetores-alvo descritos na Seção 4.1, o tamanho utilizado para treinamento da rede em cada conjunto de dados, está descrito da Tabela 5.1.

Tabela 5.1: Dimensão dos vetores-alvo utilizados

Conjunto de dados	VBC	VNO	VBO
Íris Humana	70	128	128
		256	256
		512	512
		1024	1024
		2048	2048
Dígitos Manuscritos	10	16	16
Signos Australianos	95	128	128

5.3 Planejamento experimental

Os experimentos foram realizados com os três conjuntos de dados descritos na Seção 5.1 e para cada um desses conjuntos foram implementados dois modelos de RBF. O primeiro modelo foi desenvolvido utilizando a abordagem proposta por Haylin (2008) de otimização de pesos, descrita na Seção 3.4.2.1, no segundo estágio de seu treinamento, referenciado apenas como MH a partir de agora.

Já o segundo modelo, em sua fase supervisionada utilizou a abordagem proposta por Silva et al. (2010) e empregou a Regra Delta Generalizada para atualização dos pesos da camada de saída, a partir desta seção, este modelo será referenciado apenas por MRD.

Para cada um dos conjuntos de dados, foram feitos experimentos com 10 valores dife-

rentes de neurônios na camada intermediária (quantidade de subgrupos/clusters), além disso, as dimensões especificadas na Seção 5.2 dos vetores-alvo, definidos na Seção 4.1, também foram utilizadas. Todos os experimentos feitos com o emprego da Regra Delta Generalizada utilizou taxa de aprendizagem adaptativa (Duffner & Garcia, 2007) com valor inicial em 0.05 e termo *momentum* $[0, 0.9]$ definido como $\beta = 0.9$ (Silva et al., 2010).

Os experimentos foram repetidos 100 vezes para cada combinação desses valores, com inicialização de pesos aleatórios nas duas fases do algoritmo de treinamento. Os pesos sinápticos iniciais foram gerados aleatoriamente entre -0.5 e 0.5 . Os resultados apresentados neste trabalho representam a média das 100 execuções de cada um dos experimentos feitos.

Todos os modelos experimentais foram desenvolvidos e testados utilizando o software Matlab ®R2016a. Os testes estatísticos realizados durante o trabalho foram feitos com o software de análises estatísticas R (R Core Team, 2014).

5.4 Parâmetros analisados

Cada conjunto de dados foi dividido em três subgrupos que representavam os dados para treinamento, validação e teste. Nos experimentos feitos com a base de dados da íris humana, o conjunto de treinamento continha quatro amostras de cada indivíduo, totalizando 280 amostras.

Para o conjunto de 10 dígitos manuscritos foram utilizadas no total 450 amostras. Finalmente, foram utilizadas 9 amostras de cada um dos 95 tipos de sinais de signos australianos, totalizando 855 amostras para cada um dos conjuntos (treinamento, validação e teste). Todos os parâmetros de desempenho foram obtidos por meio dos conjuntos de teste.

O MH é executado em apenas um ciclo de treinamento e por isso, foi analisado apenas o desempenho médio e o desempenho máximo para os experimentos. A título de comparação das técnicas, a implementação de MRD também foi analisada seguindo os mesmos parâmetros.

Considerando que em (Manzan et al., 2016) estes conjuntos de dados foram utilizados para experimentos em redes MLP, o modelo MRD foi analisado utilizando os mesmos parâ-

metros utilizados para a rede MLP para fins comparatórios. Os parâmetros analisados em cada experimento foram:

1. Desempenho máximo global
2. Desempenho máximo em 50 ciclos
3. Desempenho após o primeiro ciclo
4. Desempenho médio dos cinco primeiros ciclos

5.5 Resultados Experimentais

A Tabela 5.2 mostra os resultados do teste de normalidade de Kolmogorov-Smirnov (Conover, 1999) realizados com os resultados obtidos nos experimentos com dígitos manuscritos. A estatística desse teste é representada pela letra D e usando nível de significância $\alpha = 0.05$, os dados não se ajustam à distribuição normal.

O mesmo fenômeno ocorre para os resultados obtidos nos experimentos com íris humana e signos australianos, como pode ser visto nas Tabelas 5.3 e 5.4. Por essa razão, o teste não paramétrico de Mann-Whitney (Conover, 1999) foi utilizado para comparação de significância entre as amostras dos três tipos de conjuntos de dados.

Tabela 5.2: Teste de normalidade de Kolmogorov-Smirnov - Dígitos Manuscritos

Ciclo	1	5	10	15	20	25	30	35	40	45	50
D	0.170	0.130	0.119	0.112	0.168	0.214	0.210	0.208	0.192	0.153	0.168

Tabela 5.3: Teste de normalidade de Kolmogorov-Smirnov - Íris Humana

Ciclo	1	5	10	15	20	25	30	35	40	45	50
D	0.415	0.452	0.338	0.354	0.362	0.352	0.333	0.319	0.289	0.249	0.201

Tabela 5.4: Teste de normalidade de Kolmogorov-Smirnov - Sinais Australianos

Ciclo	1	5	10	15	20	25	30	35	40	45	50
D	0.369	0.365	0.348	0.313	0.301	0.273	0.233	0.225	0.194	0.181	0.145

A Tabela 5.5 mostra os resultados do teste estatístico de Mann-Whitney de todas as comparações com o uso de diferentes vetores-alvo nos experimentos com dígitos manuscritos. A hipótese nula é que não existem diferenças significativas entre os vetores, e a hipótese alternativa é que existem diferenças significativas entre os vetores. Para cada comparação é gerada uma estatística do teste representada como “valor-p”.

Entende-se por “valor-p” sob a hipótese nula (considerada como verdadeira) a probabilidade de obtenção de um valor igual ou mais extremo do que o valor obtido na amostra. De modo análogo, as Tabelas 5.6 e 5.7 mostram, respectivamente, os resultados do teste estatístico de Mann-Whitney referentes aos experimentos com íris humana e signos australianos.

As comparações com os demais vetores-alvo utilizados nos experimentos feitos com a íris humana estão dispostos no Apêndice A. Os valores representados com asteriscos, nas tabelas posteriores, indicam que não existem diferenças ao nível de 1% de significância.

Tabela 5.5: Teste de Mann-Whitney - Dígitos Manuscritos

Ciclo	Comparação		
	$VBC_{10} \times VNO_{16}$	$VBC_{10} \times VBO_{16}$	$VNO_{10} \times VBO_{16}$
1	0.35112	*	*
5	0.27479	*	*
10	0.22780	*	*
15	0.23129	*	*
20	0.18167	*	*
25	0.33009	*	*
30	0.16136	*	*
35	0.06641	*	*
40	0.26642	*	*
45	0.05782	*	*
50	0.35112	*	*

Tabela 5.6: Teste de Mann-Whitney - Íris Humana

Ciclo	Comparação		
	$VBC_{70} \times VNO_{128}$	$VBC_{70} \times VBO_{128}$	$VNO_{128} \times VBO_{128}$
1	0.720	*	*
5	0.123	*	*
10	0.201	*	*
15	0.923	*	*
20	0.323	*	*
25	0.452	*	*
30	0.701	*	*
35	0.359	*	*
40	0.074	*	*
45	0.468	*	*
50	0.865	*	*

Tabela 5.7: Teste de Mann-Whitney - Sinais Australianos

Ciclo	Comparação		
	$VBC_{95} \times VNO_{128}$	$VBC_{95} \times VBO_{128}$	$VNO_{128} \times VBO_{128}$
1	0.5857	*	*
5	0.2570	*	*
10	0.7572	*	*
15	0.5047	*	*
20	0.7759	*	*
25	0.8969	*	*
30	0.1486	*	*
35	0.1469	*	*
40	0.4853	*	*
45	0.3775	*	*
50	0.4639	*	*

Em sua camada intermediária, redes RBF utilizam funções de ativação do tipo base radial, como as expressas nas Equações (3.8) a (3.12), a Figura 5.1 mostra uma comparação do desempenho obtido no treinamento realizado com a base de dados da Íris Humana utilizando três das principais funções. A partir dos resultados apresentados, pode-se concluir que o desempenho da função gaussiana foi superior aos demais em todos os ciclos do treinamento e por isso, nos demais experimentos ela foi utilizada.

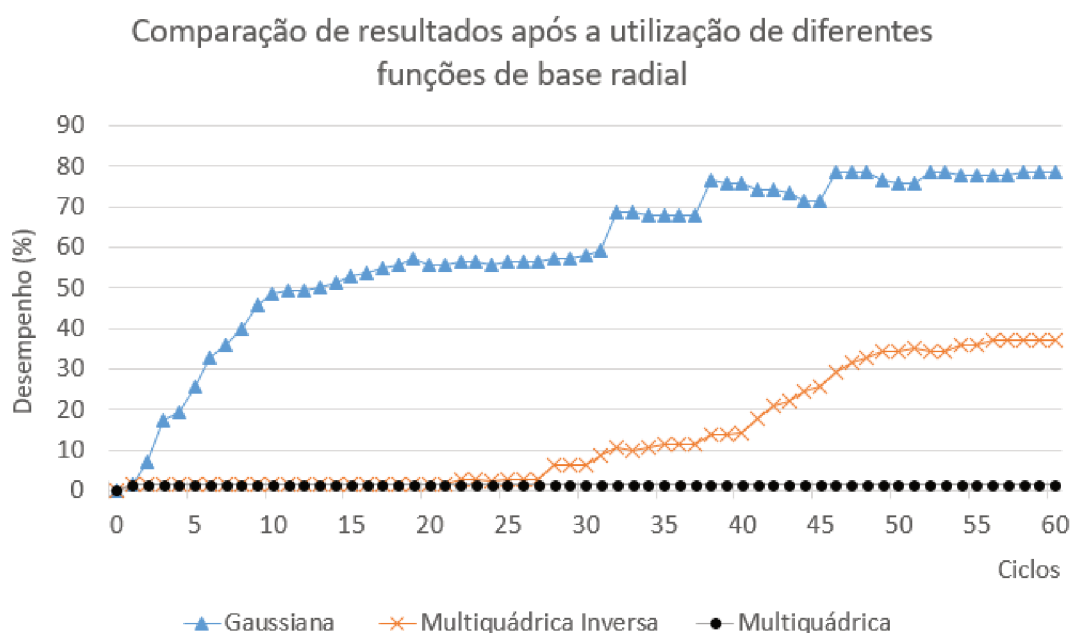


Figura 5.1: Comparação de resultados após a utilização de diferentes funções de base radial

Durante a escolha da estratégia de aprendizagem a ser utilizada, foram feitos experimentos utilizando a abordagem de Centros Fixos Seleccionados ao Acaso, descritos na Seção 3.3.1, e por Seleção Auto-organizada de Centros, apresentada na Seção 3.3.2.

Os resultados encontrados nesses dois experimentos demonstraram que a abordagem de treinamento híbrido da rede, utilizando a seleção auto-organizada de centros, apresentou média de desempenho superior à outra estratégia em cerca de 20%. Isso acontece porque na seleção auto-organizada, os clusters e os centros de cada um deles são ajustados com maior precisão.

A Tabela 5.8 mostra o valor médio dos resultados obtidos com implementação de MH. Devido à sua forma de implementação, os pesos são ajustados em apenas um ciclo, por isso, para fins comparatórios entre as duas implementações, a Tabela 5.9 apresenta o valor médio dos desempenhos máximos de cada experimento na modelagem MRD.

Tabela 5.8: Desempenho obtido com a rede MH

Dados Experimentais	Vetores-alvo	Desempenho médio	Desempenho máximo
Íris Humana	VBC ₇₀	87.69	93.57
	VNO ₁₂₈	87.44	93.57
	VNO ₂₅₆	87.46	93.78
	VNO ₅₁₂	87.33	93.57
	VNO ₁₀₂₄	87.56	93.45
	VNO ₂₀₄₈	87.31	93.57
	VBO ₁₂₈	87.31	93.78
	VBO ₂₅₆	87.91	93.30
	VBO ₅₁₂	87.55	92.14
	VBO ₁₀₂₄	87.12	93.45
	VBO ₂₀₄₈	87.56	93.57
Dígitos Manuscritos	VBC ₁₀	87.51	89.38
	VNO ₁₆	87.72	89.56
	VBO ₁₆	87.73	89.51
Sinais Australianos	VBC ₉₅	55.85	57.76
	VNO ₁₂₈	54.02	55.90
	VBO ₁₂₈	54.30	55.97

Tabela 5.9: Desempenho obtido com a rede MRD

Dados Experimentais	Vetores-alvo	Desempenho médio	Desempenho máximo
Íris Humana	VBC ₇₀	79.08	87.14
	VNO ₁₂₈	79.87	87.85
	VNO ₂₅₆	91.19	94.28
	VNO ₅₁₂	91.37	93.57
	VNO ₁₀₂₄	79.58	87.14
	VNO ₂₀₄₈	79.66	86.42
	VBO ₁₂₈	90.75	91.42
	VBO ₂₅₆	95.22	96.42
	VBO ₅₁₂	95.12	96.42
	VBO ₁₀₂₄	90.89	91.42
	VBO ₂₀₄₈	90.82	91.42
Dígitos Manuscritos	VBC ₁₀	83.33	90.86
	VNO ₁₆	83.45	91.11
	VBO ₁₆	81.89	90.12
Sinais Australianos	VBC ₉₅	66.08	76.04
	VNO ₁₂₈	66.16	77.07
	VBO ₁₂₈	68.34	73.68

Os resultados obtidos por (Manzan et al., 2016) em seus experimentos com estes conjuntos de dados utilizando redes do tipo MLP estão dispostos nas Tabelas 5.10, 5.11, 5.12. De forma análoga, as Tabelas 5.13, 5.14, 5.15 mostram os resultados obtidos nos experimentos feitos com a implementação MRD utilizando os parâmetros para fins comparatórios das duas redes.

Tabela 5.10: Resultados MLP - Dígitos Manuscritos

Parâmetros analisados	VBC₁₀	VNO₁₆	VBO₁₆
Desempenho após o primeiro ciclo	56.13	50.79	69.59
Desempenho médio dos cinco primeiros ciclos	63.34	59.79	74.54
Desempenho médio em 50 ciclos	76.91	76.52	82.45
Desempenho máximo global	79.49	79.34	84.93

Tabela 5.11: Resultados MLP - Íris Humana

Parâmetros analisados	VBC₇₀	VNO₁₂₈	VBO₁₂₈
Desempenho após o primeiro ciclo	7.01	5.16	77.13
Desempenho médio dos cinco primeiros ciclos	18.51	12.22	85.48
Desempenho médio em 50 ciclos	83.48	82.13	89.18
Desempenho máximo global	83.88	82.72	90.44

Tabela 5.12: Resultados MLP - Sinais Australianos

Parâmetros analisados	VBC₉₅	VNO₁₂₈	VBO₁₂₈
Desempenho após o primeiro ciclo	2.82	2.68	28.56
Desempenho médio dos cinco primeiros ciclos	2.88	2.79	47.06
Desempenho médio em 50 ciclos	62.77	60.06	78.84
Desempenho máximo global	76.39	75.26	82.63

O gráfico representado na Figura 5.2, representa os resultados obtidos na Tabela 5.13 e mostra uma comparação entre os três tipos de vetores-alvo apresentados neste trabalho. Os valores apresentados, correspondem a média dos resultados dos 100 experimentos realizados com 50 ciclos na fase de treinamento. Da mesma maneira, os gráficos das Figuras 5.3 e 5.4 representam os valores sintetizados nas Tabelas 5.14 e 5.15, respectivamente.

Tabela 5.13: Resultados RBF - MRD - Dígitos Manuscritos

Parâmetros analisados	VBC₁₀	VNO₁₆	VBO₁₆
Desempenho após o primeiro ciclo	24.45	23.96	50.75
Desempenho médio dos cinco primeiros ciclos	62.64	62.02	71.19
Desempenho médio em 50 ciclos	83.33	83.45	81.89
Desempenho máximo global	90.86	91.11	90.12

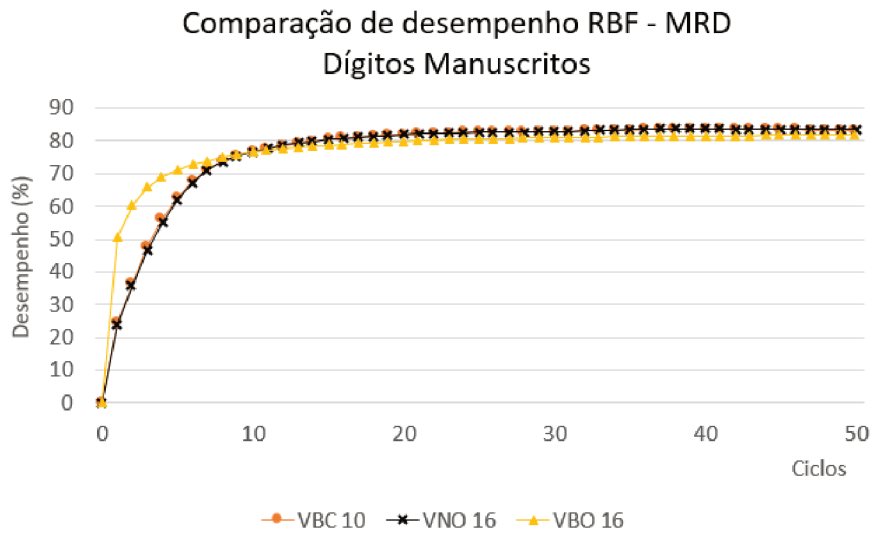


Figura 5.2: Gráfico para comparação de desempenho de diferentes vetores alvo em RBF - MRD: Dígitos Manuscritos

Tabela 5.14: Resultados RBF - MRD - Íris Humana

Parâmetros analisados	VBC ₇₀	VNO ₁₂₈	VBO ₁₂₈
Desempenho após o primeiro ciclo	3.43	3.37	10.02
Desempenho médio dos cinco primeiros ciclos	6.12	6.27	70.75
Desempenho médio em 50 ciclos	63.47	63.55	88.2
Desempenho máximo global	87.17	87.85	91.42

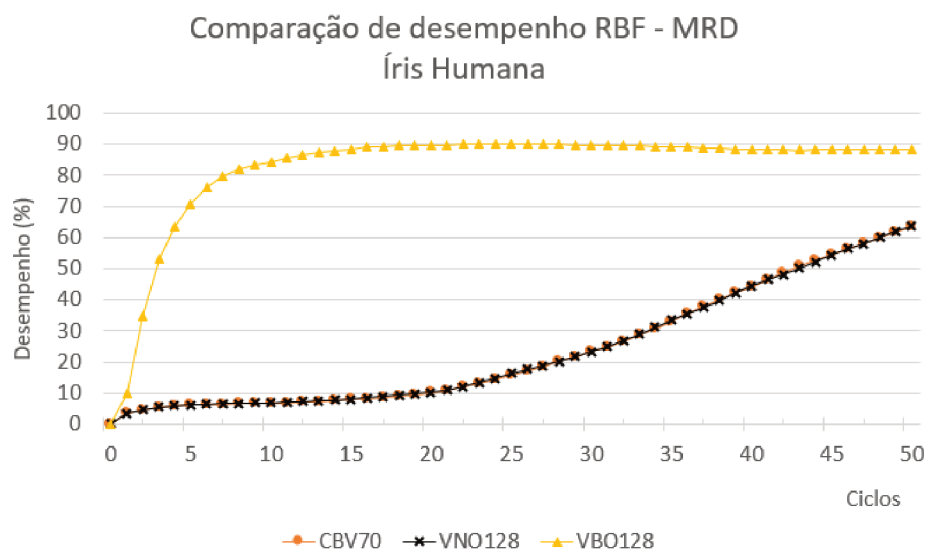


Figura 5.3: Gráfico para comparação de desempenho de diferentes vetores alvo em RBF - MRD: Íris Humana

Tabela 5.15: Resultados RBF - MRD - Sinais Australianos

Parâmetros analisados	VBC ₉₅	VNO ₁₂₈	VBO ₁₂₈
Desempenho após o primeiro ciclo	4.44	4.57	41.55
Desempenho médio dos cinco primeiros ciclos	11.61	12.00	59.33
Desempenho médio em 50 ciclos	66.08	66.16	68.34
Desempenho máximo global	76.07	77.07	76.68

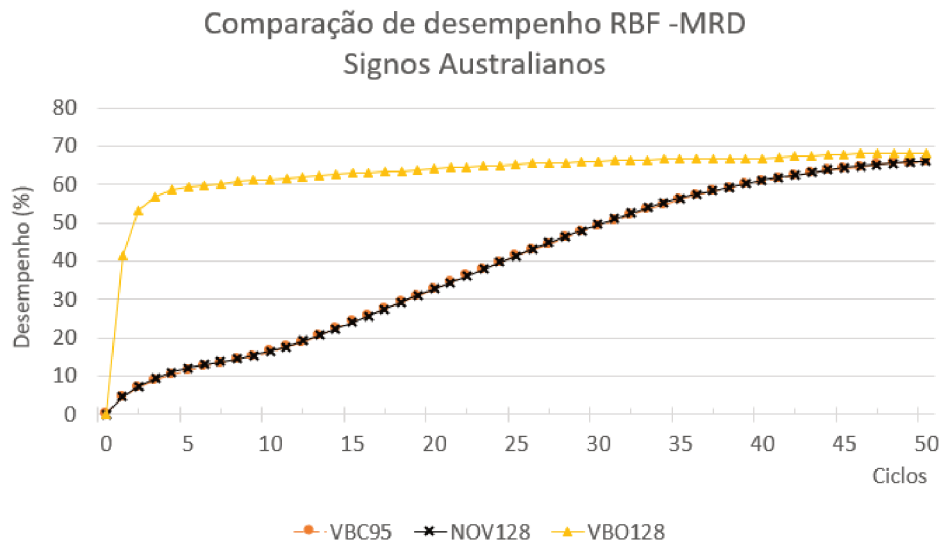


Figura 5.4: Gráfico para comparação de desempenho de diferentes vetores alvo em RBF - MRD: Sinais Australianos

5.6 Discussão

A análise estatística do trabalho foi feita utilizando o teste de Mann-Whitney. Neste teste, o valor de “p” obtido como resultado das comparações, indica se existem diferenças ou não entre os dados. Se os valores forem abaixo de 0.05, demonstram que não existem diferenças consideráveis entre as amostras. Porém, valores maiores indicam que o resultado da comparação de dados com diferenças significantes.

Os resultados do teste de Mann-Whitney demonstram que não existem diferenças significativas entre os VBCs e os VNOs em nenhum dos conjuntos de testes. Porém, na comparação entre os VNOs com os VBOs e dos VBCs com os VBOs os valores de “p” encontrados

foram todos menores que $2.2E-16$, representando a diferença que existe entre os vetores comparados.

Os resultados obtidos pela modelagem MH na Tabela 5.8, quando comparados com os obtidos pela modelagem MRD presentes pela Tabela 5.9, mostram que na utilização dos VNOs as duas redes possuem desempenho equiparáveis, porém, quando os VBOs são utilizados como vetores-alvo nas redes MRD, o desempenho delas passa a ser superior em todos os conjuntos de dados experimentados nesta pesquisa.

De acordo com a Tabela 5.8, os dados treinados com a íris, utilizando VBO_{128} , obtiveram desempenho médio de 87.31%, enquanto que na modelagem MRD com os mesmos vetores-alvo, o desempenho médio da rede chegou a 90.75%. Já com a utilização de VBO_{256} a diferença é ainda maior, a modelagem MH obteve desempenho médio de 87.91%, enquanto que na modelagem MRD o desempenho médio da rede foi de 95.22%.

Com o conjunto de dados referente aos dígitos manuscritos, a melhora da performance não foi significativa. Obteve-se com a RBF - MRD desempenho máximo de 90.12% e na modelagem MH, o resultado foi de 89.51%. Na base de dados dos Signos Australianos, nota-se novamente uma melhor performance utilizando a Regra Delta Generalizada. Os resultados do desempenho médio da modelagem MH para os sinais australianos foram de 54.30% de acerto, enquanto que com a rede MRD, obteve-se 68.34% de taxa de acerto.

As Figuras 5.2, 5.3 e 5.4 representam os resultados obtidos pela execução da rede MRD e os gráficos demonstram a superioridade dos VBOs em relação aos demais vetores utilizados, sendo notável nos 3 conjuntos de dados que nos 5 primeiros ciclos de treinamento os VBOs conseguiram atingir uma taxa de acerto maior que os demais. As Tabelas 5.13, 5.14 e 5.15 também demonstram as mesmas características visualizadas nos gráficos.

Nos experimentos feitos com as imagens de íris, no 1º ciclo a rede apresentava 10.02% de acerto com VBO_{128} , enquanto usando VBC_{70} estava com apenas 3.43% de acerto e com VNO_{128} com 3.37%. No 5º ciclo, a taxa de acerto com o VBO_{128} foi de 70.75%, com a utilização do VBC_{70} a performance foi de 6.12% e com VNO_{128} a taxa de acerto foi de 6.27%. O melhor desempenho dos VBOs continua nos ciclos posteriores, sendo que, no 10º ciclo a rede apresenta

6.92% de acerto com os VBC_{70} , 6.97% com os VNO_{128} e 84.30% utilizando-se VBO_{128} .

Os resultados com os Sinais Australianos também apresentam a mesma característica. No 5º ciclo a modelagem da rede que utiliza VBO_{128} como vetor-alvo possui 59.33% de acerto, enquanto que com VBC_{95} a taxa de acerto é 11.61% e usando VNO_{128} a porcentagem de acerto é de 12%. Nos dígitos manuscritos, logo no 1º ciclo a taxa de acerto da rede modelada com VBO_{16} apresenta 50.75% de acerto, enquanto as modelagens que possuem VBC_{10} e VNO_{16} apresentam porcentagem de acerto em torno de 24%. Comprovando dessa forma, a eficácia dos novos vetores para estes problemas de classificação de padrões, utilizando como abordagem de treinamento a Regra Delta Generalizada, uma vez que o desempenho foi maior em um número de ciclos reduzido.

Comparando os resultados apresentados nas Tabelas 5.10, 5.11 e 5.12, que apresenta os valores obtidos com a rede MLP, com os apresentados pelas Tabelas 5.13, 5.14 e 5.15, respectivamente, cujos valores são aqueles obtidos pela execução da rede com modelagem MRD, nota-se que com o conjunto de dados da Íris Humana, a RBF apresentou uma taxa de acerto bem próxima que a MLP, porém com base de dados dos Signos Australianos que a porcentagem de acerto da MLP foi 5.95% maior que a da RBF. Já no conjunto de dados dos Dígitos Manuscritos, a RBF acertou 5.19% mais padrões que a MLP.

Utilizando a base de dados das imagens da íris, o desempenho máximo obtido pela MLP foi de 90.44%, utilizando-se os VBOs, já a taxa de acerto da RBF para o mesmo conjunto de dados foi de 91.42%. Nos dígitos manuscritos, o desempenho da RBF foi de 90.12% enquanto que o apresentado para a MLP foi de 84.93%. Utilizando a base de dados dos Signos Australianos, o desempenho da RBF foi de 76.68% e a MLP apresentou resultado de 82.63%. Essa semelhança na média dos resultados das comparações, demonstra que as duas redes possuem desempenho equivalente para os padrões testados. Além disso, nota-se que os VBOs melhoram o desempenho dos dois tipos de RNAs.

Capítulo 6

Conclusão

Este trabalho propôs a utilização de novos vetores-alvo em RNAs do tipo RBF. A característica de ortogonalidade mútua desses vetores faz com que estejam geometricamente a uma maior distância euclidiana. Essa maior distância entre os pontos do espaço de saída oferece à rede maior capacidade de generalização, redução do esforço computacional e robustez.

Foi demonstrado que o distanciamento dos pontos, melhora o desempenho para a modelagem realizada com o algoritmo da Regra Delta Generalizada, porque ao longo do treinamento, a saída obtida pela rede para cada amostra aproxima-se do seu alvo correspondente. Se os alvos estão mais distantes uns dos outros, haverá menor probabilidade de as saídas geradas pela rede em tempo de treinamento estarem mais próximas de alvos incorretos. Consequentemente, a tarefa de classificação dos padrões será facilitada.

Porém na modelagem utilizando o método de otimização de pesos proposto por Haykin (2008), a utilização desses vetores não impacta o resultado da rede devido a sua forma de ajuste de pesos. Esse comportamento acontece devido à forma de agrupamento/clusterização realizada, onde os grupos se ajustam de acordo com os membros que pertencem a ele, conforme explicado na Seção 4.4.1. Essa característica também foi demonstrada pelos experimentos realizados durante o trabalho e sintetizados na tabela 5.8, onde a variação do desempenho das redes com diferentes vetores-alvo não passou de 1%.

As duas modelagens propostas para redes RBF foram implementadas e experimental-

mente analisadas para o reconhecimento de dígitos manuscritos, íris humana e signos australianos. A comparação feita entre as duas redes, analisando seu desempenho médio e máximo demonstrou que, na utilização de vetores convencionais, ambas possuem desempenho equiparável. Porém na utilização de VBO, objeto de pesquisa deste trabalho, a rede modelada com a Regra Delta Generalizada apresentou melhores resultados para taxa de acerto dos padrões experimentados.

Estes resultados também nos mostram que a modificação realizada na camada de saída, alterando a função de ativação de linear para sigmoide melhorou o desempenho da rede em cerca de 9% para os dígitos australianos e cerca de 3% nos demais conjuntos de dados testados. O aumento no desempenho ocorre porque essa alteração na camada de saída proporcionou uma nova transformação nos dados onde antes havia apenas uma combinação linear.

Os dois tipos de redes neurais utilizados nesta pesquisa são tradicionalmente utilizados em aplicações para reconhecimento de padrões, aproximação funcional, entre outros. As redes se diferenciam quanto características em sua arquitetura. A comparação das redes RBF e MLP demonstrou que, a RBF necessita de um conjunto de dados de entrada maior para realizar o treinamento e ajuste de pesos, enquanto a MLP consegue resultados semelhantes com menos dados de entrada. Em contrapartida, o tempo de convergência da RBF é menor quando comparado à MLP.

Os experimentos realizados para comparação das redes, consistiram na comparação do desempenho em vários momentos do treinamento usando-se VBCs, VNOs e VBOs. Eles demonstraram que, para o conjunto de padrões utilizados, as duas redes são equiparáveis na tarefa de reconhecimento de padrões.

Outro aspecto importante que foi verificado, foi que as redes treinadas com VBOs têm desempenho superior em todos esses momentos do treinamento. Foi verificado também que o aumento da dimensão dos vetores convencionais não melhora o desempenho, em alguns casos, o desempenho com vetores de dimensões maiores foi inferior. Essa mesma característica, foi observada na utilização dos VBOs, onde os vetores com dimensões muito grandes, como 1024 e 2048, apresentaram menor desempenho que os demais.

Destaca-se que, além do aumento do desempenho global do treinamento, o uso de VBOs como alvos de RBFs permite inclusive que a rede alcance um nível satisfatório de desempenho com pouco treinamento. Isso é especialmente importante pelo fato de reduzir o esforço computacional.

Os experimentos também demonstraram que o processo de clusterização feito na camada intermediária é de extrema importância para o bom desempenho na tarefa de classificação de padrões. Em trabalhos futuros, espera-se realizar experimentos com outros tipos de algoritmos para clusterização dos dados na camada intermediária para analisar e comparar o desempenho da rede com outras bases de dados.

Apêndice A

Teste de Mann-Whitney

Vetores	1	5	10	15	20	25	30	35	40	45	50
CBV70 x NOV128	0.512	0.914626	0.550437	0.917541	0.275391	0.12363	0.677306	0.427429	0.695998	0.489651	0.964828
CBV70 X NOV256	0.08738	*	*	*	*	*	*	*	*	*	*
CBV70 x NOV512	0.003688	*	*	*	*	*	*	*	*	*	*
CBV70 x NOV1024	0.6808	0.262312	0.132572	0.312186	0.713009	0.518463	0.914952	0.250881	0.38246	0.295546	0.250846
CBV70 X NOV2048	0.6713	0.289763	0.449735	0.999017	0.425711	0.826974	0.565621	0.369999	0.41511	0.524193	0.651508
CBV70 X OBV128	0.9261	*	*	*	*	*	*	*	*	*	*
CBV70 X OBV256	*	*	*	*	*	*	*	*	*	*	*
CBV70 X OBV512	*	*	*	*	*	*	*	*	*	*	*
CBV70 X OBV1024	0.6296	*	*	*	*	*	*	*	*	*	*
CBV70 X OBV2048	8412	*	*	*	*	*	*	*	*	*	*
NOV128 X NOV256	0.02711	*	*	*	*	*	*	*	*	*	*
NOV128 X NOV512	*	*	*	*	*	*	*	*	*	*	*
NOV128 X NOV1024	0.8445	0.342471	0.338783	0.232548	0.480911	0.02468	0.700863	0.780616	0.263475	0.090504	0.230253
NOV128 X NOV2048	0.8002	0.320668	0.90695	0.855552	0.068396	0.17899	0.482815	0.849288	0.690534	0.151067	0.727144
NOV128 X OBV128	0.7987	*	*	*	*	*	*	*	*	*	*
NOV128 X NOV256	*	*	*	*	*	*	*	*	*	*	*
NOV128 X OBV512	*	*	*	*	*	*	*	*	*	*	*
NOV128 X OBV1024	0.8118	*	*	*	*	*	*	*	*	*	*
NOV128 X OBV2048	0.9689	*	*	*	*	*	*	*	*	*	*
NOV256 X NOV512	0.2562	0.551231	0.188105	0.465768	0.765582	0.481723	0.367597	0.014044	0.189381	0.288308	0.639414
NOV256 X NOV1024	0.04028	*	*	*	*	*	*	*	*	*	*
NOV256 X NOV2048	0.0438	*	*	*	*	*	*	*	*	*	*
NOV256 X OBV128	0.01634	*	*	*	*	*	*	*	*	*	*
NOV256 X OBV256	*	*	*	*	*	*	*	*	*	*	*
NOV256 X OBV512	*	*	*	*	*	*	*	*	*	*	*
NOV256 X OBV1024	*	*	*	*	*	*	*	*	*	*	*
NOV256 X OBV2048	*	*	*	*	*	*	*	*	*	*	*
NOV512 X NOV1024	*	*	*	*	*	*	*	*	*	*	*
NOV512 X NOV2048	*	*	*	*	*	*	*	*	*	*	*
NOV512 X OBV128	*	*	*	*	*	*	*	*	*	*	*
NOV512 X OBV256	*	*	*	*	*	*	*	*	*	*	*
NOV512 X OBV512	*	*	*	*	*	*	*	*	*	*	*
NOV512 X OBV1024	*	*	*	*	*	*	*	*	*	*	*
NOV512 X OBV2048	*	*	*	*	*	*	*	*	*	*	*
NOV1024 X NOV2048	0.9714	0.975426	0.447827	0.357813	0.229267	0.398091	0.73948	0.924778	0.133925	0.705093	0.133882
NOV1024 X OBV128	0.8948	*	*	*	*	*	*	*	*	*	*
NOV1024 X OBV256	*	*	*	*	*	*	*	*	*	*	*
NOV1024 X OBV512	*	*	*	*	*	*	*	*	*	*	*
NOV1024 X OBV1024	0.6012	*	*	*	*	*	*	*	*	*	*
NOV1024 X OBV2048	0.7535	*	*	*	*	*	*	*	*	*	*
NOV2048 X OBV128	0.669	*	*	*	*	*	*	*	*	*	*
NOV208 X OBV256	*	*	*	*	*	*	*	*	*	*	*
NOV2048 X OBV512	*	*	*	*	*	*	*	*	*	*	*
NOV2048 X OBV1024	0.8278	*	*	*	*	*	*	*	*	*	*
NOV2048 X OBV2048	0.6221	*	*	*	*	*	*	*	*	*	*
OBV128 X OBV256	*	*	*	*	*	*	*	*	*	*	*
OBV128 X OBV512	*	*	*	*	*	*	*	*	*	*	*
OBV128 X OBV1024	0.1338	*	*	0.012977	*	0.568592	0.087761	0.721882	0.839393	*	*
OBV128 X OBV2048	0.249	*	*	0.036136	*	0.92724	0.129083	0.890327	0.732381	*	*
OBV256 X OBV512	0.05484	0.000106	0.35547	0.086972	0.645765	0.20082	0.435129	0.003475	0.551962	0.514455	*
OBV256 X OBV1024	*	*	*	*	*	*	*	*	*	*	*
OBV256 X OBV2048	*	*	*	*	*	*	*	*	*	*	*
OBV512 X OBV1024	*	*	*	*	*	*	*	*	*	*	*
OBV512 X OBV2048	*	*	*	*	*	*	*	*	*	*	*
OBV1024 X OBV2048	0.09817	0.679253	0.489327	0.553595	0.24638	0.612953	0.777025	0.682764	0.322174	*	0.084822

Os asteriscos indicam que não existem diferenças significativas ao nível de 1% de significância.

Referências Bibliográficas

- Araújo, J. R. (2005). *Previsão regionalizada de vazão sazonal utilizando redes neurais artificiais* (Unpublished master's thesis). Centro de Tecnologia e Recursos Naturais - UFCG.
- Bianchi, M. F. (2006). *Extração de características de imagens de faces humanas através de Wavelets, PCA e IMPCA* (Unpublished master's thesis). Escola de Engenharia de São Carlos - USP.
- Birlutiu, A., d'Alche Buc, F., & Heskes, T. (2014). A bayesian framework for combining protein and network topology information for predicting protein-protein interactions. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, PP(99), 1-1. doi: <https://doi.org/10.1109/TCBB.2014.2359441>
- Broomhead, D. S., & Lowe, D. (1988). *Multivariable functional interpolation and adaptive networks* (Vol. 2). Complex Systems.
- Casia. (2010). *Human iris, database of 756. greyscale eye images*. Retrieved from <http://www.cbsr.ia.ac.cn/IrisDatabase.htm>
- Chen, S., Mulgrew, B., & McLaughlin, S. (1992). Adaptive bayesian feedback equalizer based in a radial basis function network. In (Vol. 3, pp. 1267–1271).
- Connel, S. D., & Jain, A. K. (2001). Template-based online character recognition. , 34, 1–14. doi: [https://doi.org/10.1016/S0031-3203\(99\)00197-1](https://doi.org/10.1016/S0031-3203(99)00197-1)
- Conover, W. (1999). *Practical nonparametric statistics*. IE-Wiley.
- Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Eletronic Computers, EC-14*, 326–334. doi: <https://doi.org/10.1109/PGEC.1965.264137>

- Daugman, J. G. (1993, November). High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11), 1148–1161. Retrieved from <http://dx.doi.org/10.1109/34.244676> doi: 10.1109/34.244676
- Du, K.-L., & Swamy, M. (2013). *Neural networks and statistical learning*. doi: 10.1007/978-1-4471-5571-3
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. 2nd.
- Duffner, S., & Garcia, C. (2007). An online backpropagation algorithm with validation error-based adaptive learning rate. In *Artificial neural networks–icann 2007* (pp. 249–258). Springer. doi: https://doi.org/10.1007/978-3-540-74690-4_26
- Eberhart, R., & Shi, Y. (2007). *Computational intelligence: Concepts to implementations*. Elsevier. doi: <https://doi.org/10.1016/B978-155860759-0/50002-0>
- Fausett, L. V., & Hall, P. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications* (Vol. 40). Prentice-Hall Englewood Cliffs.
- Golbabai, A., Mammadov, M., & Seifollahi, S. (2019). Solving a system of nonlinear integral equations by an RBF network. doi: <https://doi.org/10.1016/j.camwa.2009.03.038>
- Gonzalez, R. C. (1992). RE woods, digital image processing. *Addison–Wesely Publishing Company*.
- Haykin, S. (2008). *Redes neurais: Princípios e prática*. Bookman.
- Healy, E. W., Yoho, S. E., Wang, Y., & Wang, D. (2013). An algorithm to improve speech recognition in noise for hearing-impaired listeners. *The Journal of the Acoustical Society of America*, 134(4), 3029-3038. Retrieved from <http://scitation.aip.org/content/asa/journal/jasa/134/4/10.1121/1.4820893> doi: <http://dx.doi.org/10.1121/1.4820893>
- Hebb, D. (1968). *O.(1949) the organization of behavior*. Wiley, New York.
- Huang, G., Huang, G.-B., Song, S., & You, K. (2015). Trends in extreme learning machines: A review. *Neural Networks*, 61(0), 32 - 48. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0893608014002214> doi: <http://dx.doi.org/10.1016/>

j.neunet.2014.10.001

- Kuo, R., Hu, T.-L., & Chen, Z.-Y. (2009). Application of radial basis function neural network for sales forecasting. In (p. 325-328). doi: <https://doi.org/10.1109/CAR.2009.97>
- Lichman, M. (2013). *UCI machine learning repository*. Retrieved from <http://archive.ics.uci.edu/ml>
- Manzan, J. R. G. (2016). *Análise de desempenho de redes neurais do tipo multilayer perceptron por meio do distanciamento dos pontos do espaço de saída* (Unpublished doctoral dissertation). UFU - Faculdade de Engenharia Elétrica.
- Manzan, J. R. G., Nomura, S., & Filho, J. B. D. (2014). EEG signal classification by improved MLPs with new target vectors. In *Proceedings on the international conference on artificial intelligence (ICAI)* (p. 1).
- Manzan, J. R. G., Nomura, S., & Yamanaka, K. (2011). A melhoria no desempenho de MLP com o uso de novos vetores alvo. In *10th brazilian congress on computational intelligence (CBIC 2011)*.
- Manzan, J. R. G., Nomura, S., & Yamanaka, K. (2012). Mathematical evidence for target vector type influence on MLP learning improvement. In *Proceedings on the international conference on artificial intelligence (ICAI)* (p. 1).
- Manzan, J. R. G., Nomura, S., Yamanaka, K., Carneiro, M. B. P., & Veiga, A. C. P. (2012). Improving iris recognition through new target vectors in MLP artificial neural networks. In *Artificial neural networks in pattern recognition* (pp. 115–126). Springer. doi: https://doi.org/10.1007/978-3-642-33212-8_11
- Manzan, J. R. G., Yamanaka, K., & Nomura, S. (2011). Improvement in performance of MLP using new target vectors (in portuguese). In: *X Brazilian Congress on Computational Intelligence - Fortaleza*.
- Manzan, J. R. G., Yamanaka, K., Peretta, I. S., Pinto, E. R., Oliveira, T. E. C., & Nomura, S. (2016). A mathematical discussion concerning the performance of multilayer perceptron-type artificial neural networks through use of orthogonal bipolar vectors. *Computational and Applied Mathematics*, 1–22. doi: <http://dx.doi.org/10.1007/s40314-016-0377-x>

- Martins, G. A., & Fonseca, J. S. (2006). Curso de estatística. *Atlas, 6ª Edição*.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133. doi: <https://doi.org/10.1007/BF02478259>
- Muda, A. K., Choo, Y.-H., Abraham, A., & Srihari, S. N. (2014). *Computational intelligence in digital forensics: Forensic investigation and applications*. Springer. Retrieved from <https://doi.org/10.1007/978-3-319-05885-6>
- Negin, M., Chmielewski, T. A., Salganicoff, M., Camus, T. A., Cahn, U. M. V. S., Venetianer, P. L., & Zhang, G. G. (2000, February). An iris biometric system for public and personal use. *Computer*, 33(2), 70–75. doi: <https://doi.org/10.1109/2.820042>
- Nomura, S., Manzan, J. R. G., & Yamanaka, K. (2010). An experimentation with improved target vectors for MLP in classifying degraded patterns. *Learning and Nonlinear Models*, 8(4), 240–252. doi: <https://doi.org/10.21528/LNLM-vol8-no4-art5>
- Nomura, S., Manzan, J. R. G., & Yamanaka, K. (2011). Análise experimental de novos vetores alvo na melhoria do desempenho de MLP. *IX Conferência de estudos em engenharia elétrica (CEEL 2011)*.
- Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., & Shiose, T. (2004). A new approach to improving math performance of artificial neural networks (in portuguese).
- Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., & Shiose, T. (2005). Improved MLP learning via orthogonal bipolar target vectors. *JACIII*, 9(6), 580–589. doi: <https://doi.org/10.20965/jaciii.2005.p0580>
- Pereira, M. B., & Veiga, A. C. P. (2005). Application of genetic algorithms to improve the reliability of an iris recognition system. In *Machine learning for signal processing, 2005 IEEE workshop on* (pp. 159–164). doi: <https://doi.org/10.1109/MLSP.2005.1532892>
- R Core Team. (2014). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <http://www.R-project.org/>
- Sbarufatti, C., Corbetta, M., Giglio, M., & Candini, F. (2017). Adaptive prognosis of lithium-ion batteries based on the combination of particle and radial basis function neural networks..

doi: <https://doi.org/10.1016/j.jpowsour.2017.01.105>

- Silva, I. N., Spatti, D. H., & Flauzino, R. A. (2010). Redes neurais artificiais para engenharia e ciências aplicadas curso prático. *Artliber*.
- Theodoridis, S., & Koutroumbas, K. (1999). *Pattern classification*. Academic Press.
- Vieira, F. H. T., Lemos, R. P., & Ling, L. L. (2003). Aplicação de redes neurais rbf treinadas com algoritmos rols e análise wavelet na predição de tráfego em redes ethernet. In (p. 145-150).
- Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., & Li, J. (2014). Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the ACM international conference on multimedia* (pp. 157–166). New York, NY, USA: ACM. doi: <http://doi.acm.org/10.1145/2647868.2654948>
- Xu, Y., Xu, D., Lin, S., Han, T. X., Cao, X., & Li, X. (2012, June). Detection of sudden pedestrian crossings for driving assistance systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3), 729-739. doi: <https://doi.org/10.1109/TSMCB.2011.2175726>