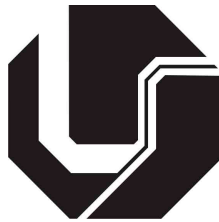


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



**AVANÇADA ANÁLISE DO USO DE NOVOS
VETORES-ALVO EM MLPs DE ALTA
PERFORMANCE**

JOSÉ RICARDO GONÇALVES MANZAN

SETEMBRO
2012

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

AVANÇADA ANÁLISE DO USO DE NOVOS
VETORES-ALVO EM MLPs DE ALTA PERFORMANCE

Dissertação apresentada por José Ricardo Gonçalves Manzan à Universidade Federal de Uberlândia perante a banca de examinadores abaixo, como requisito parcial para obtenção do título de Mestre em Ciências.

Keiji Yamanaka, Phd (UFU) - Orientador

Shiguo Nomura, Dr (UFU) - Co-Orientador

Edmilson Rodrigues Pinto, Dr (UFU)

Celso Gonçalves Camilo Junior, Dr (UFG)

M296a Manzan, José Ricardo Gonçalves, 1984-
2012 Avançada análise do uso de novos vetores-alvo em MLPs de
alta performance / José Ricardo Gonçalves Manzan. - 2012.
114 p. : il.

Orientador: Keiji Yamanaka.
Co-orientador: Shiguo Nomura.

Dissertação (mestrado) ó Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Redes neurais artificiais - Teses.
3. Reconhecimento de padrões - Teses. 3. Vetores - Análise - Teses. I.
Yamanaka, Keiji. II. Nomura, Shiguo. III. Universidade Federal de
Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. IV.
Título.

AVANÇADA ANÁLISE DO USO DE NOVOS VETORES-ALVO EM MLPs DE ALTA PERFORMANCE

JOSÉ RICARDO GONÇALVES MANZAN

Dissertação apresentada por José Ricardo Gonçalves Manzan à Universidade Federal de Uberlândia como requisito parcial para obtenção do título de Mestre em Ciências.

Keiji Yamanaka, Phd
Orientador

Alexandre Cardoso, Dr
Coordenador do Curso de Pós-Graduação

Aos meus pais Antônio Ricardo e Geralda, pelos exemplos e ensinamentos de perseverança, de respeito, honestidade e simplicidade. E à Ana Paula, minha esposa, pelo companheirismo, carinho e dedicação no desenvolvimento deste curso.

Agradecimentos

À Deus que esteve comigo em todos os momentos, principalmente naqueles em que houve maior dificuldade me mantendo firme frente aos obstáculos. À ele que é o mestre dos mestres.

Ao meu orientador Professor Keiji Yamanaka por me confiar este tema de pesquisa, pelo grande incentivo, pela amizade e motivação, pela sapiência e pelos valiosos ensinamentos que me apoiaram neste trabalho.

Ao meu co-orientador Professor Shiguelo Nomura por me apoiar durante todo o desenvolvimento da pesquisa. Seu conhecimento, paciência, amizade, ensinamentos e experiência foram fundamentais para o meu crescimento acadêmico e profissional.

Aos meus amados pais, que serão sempre meus eternos professores. 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 amada esposa. Pela paciência durante toda a jornada. Por me apoiar em todos os momentos durante essa etapa da vida. Por me encorajar frente aos problemas e obstáculos. Por toda a dedicação, amor e carinho.

À Cinara Fagundes Paranhos Mattos, pela simpatia e presteza na secretaria da Pós-Graduação.

Aos companheiros da turma do MINTER que tanto me ajudaram com conhecimentos básicos da área de computação que não é minha área de formação. E também pela amizade e companheirismo, combustíveis importantíssimos nesse desafio de vida.

À CAPES, à UFU e ao IFTM pela parceria do programa MINTER que me possibilitou o acesso ao sonho de me tornar Mestre.

Resumo

MANZAN, José R. G. *Avançada análise do uso de novos vetores-alvo em MLPs de alta performance*, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2012.

O presente trabalho propõe a análise avançada para a melhoria de desempenho de MLP através do uso de novos vetores-alvo. Primeiramente, por meio de um estudo matemático, avalia-se a influência dos VBOs sobre o treinamento das MLPs quando são utilizados como vetores-alvo. Os VBOs possuem a maior distância euclidiana possível entre si, o que leva a supor que melhora o treinamento e a capacidade de generalização da rede em teste. A hipótese é a de que a provocação de uma maior distância entre os pontos de saída da rede pode ter relação direta com a melhoria na classificação dos padrões. Os diferentes tipos de vetores-alvo tais como VBNs, VBCs e VNOs são utilizados para o treinamento de MLPs e os seus desempenhos são comparados com a rede treinada adotando-se os VBOs. As evidências matemáticas da melhoria de desempenho foram encontradas no refinamento da atualização dos pesos, etapa denominada no algoritmo como retro propagação do erro. Esse refinamento característico do treinamento com VBOs age no sentido de preservar as características de cada padrão, reduzindo o ruído de interferência do treinamento de um padrão para outro. Seguindo-se ao estudo matemático, realiza-se uma análise experimental mais avançada da utilização dos VBOs por meio de duas bases de dados para reconhecimento de padrões. A primeira base de dados é a de dígitos manuscritos para comparar os desempenhos de MLPs treinadas com VBCs e VNOs com aquelas treinadas com VBOs. Os resultados mostraram taxas de classificação superiores para a MLP treinada com VBOs. A segunda base de dados é formada por imagens de íris humana com o propósito de realizar a comparação dos desempenhos de MLPs treinadas com vetores-alvo convencionais e novos vetores-alvo representados pelos VBOs. Além da alta performance nas taxas de reconhecimento das MLPs treinadas com VBOs, observou-se que com o uso desses novos vetores-alvo, é possível obter elevadas taxas de reconhecimento com pouco rigor nas épocas de treinamento, reduzindo-se conseqüentemente a carga computacional de processamento dos padrões.

Palavras-chave: Estudo matemático, reconhecimento de padrões, multilayer perceptron, vetores-alvo, vetores bipolares ortogonais.

Abstract

MANZAN, José R. G. *Advanced analysis of using new target vectors on high performance MLPs*, Uberlândia, Faculty of Electric Engineering - UFU, 2012.

This work proposes an advanced analysis for MLP performance improvement by adopting new target vectors. Firstly, a mathematical study is done to evaluate the influence of VBOs used as target vectors on MLP training. The VBOs provide the largest possible Euclidean distance between them to improve the training and generalization capability of MLPs. The largest distance inducement between points from output space leads to direct correspondence on pattern classification improvement. The various types of target vectors such as VBNs, VBCs and VNOs are adopted for training of MLP models and their performances are compared with the model trained by using VBOs. The mathematical evidences of performance improvement were found on weight updating refinement from backpropagation error stage of the algorithm. This particular refinement for training with VBOs is useful to preserve the features of each pattern due to noise interference reduction during the training process from a pattern to another. Following the mathematical study, more advanced experimental analysis using VBOs with two databases for pattern recognition is performed. The first database is related to the handwritten digits for comparing the performances of MLPs trained by adopting VBCs and VNOs with the performance of MLP trained by adopting VBOs. The results showed higher classification rates for the MLP trained with VBOs. The second database is constituted by human iris images in order to perform the comparison of MLP performances using conventional target vectors and new target ones represented by VBOs. Besides the high performance of MLPs trained with VBOs on recognition rates, it was concluded that the use of new target vectors provides high recognition rates with low tolerance for epoch trainings leading to the consequent low computational load for pattern processing.

Keywords: Mathematical study, pattern recognition, artificial neural networks, multilayer perceptron, target vectors, orthogonal bipolar vectors.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xii
Lista de Abreviaturas	xiii
1 Introdução	15
1.1 Estado da arte	15
1.2 Justificativa	17
1.3 Originalidade	19
1.4 Estrutura do trabalho	21
2 Trabalhos relacionados	23
2.1 Mudança na escolha dos pesos sinápticos	23
2.2 Aperfeiçoamento da arquitetura da rede	24
2.3 Aperfeiçoamento do algoritmo de treinamento <i>backpropagation</i>	24
2.4 Análise de sensibilidade do sinal	25
3 Fundamentos teóricos	26
3.1 Fundamentos de reconhecimento de padrões	26
3.1.1 Conceitos de um sistema de reconhecimento de padrões	26
3.1.2 Etapas de um sistema de reconhecimento de padrões	27
3.1.3 Técnicas para classificação de padrões	29
3.2 Técnicas de redes neurais artificiais	31
3.2.1 Neurônio biológico	33
3.2.2 Neurônio artificial	34
3.2.3 Rede neural artificial	34
3.2.4 Funções de ativação	40
3.3 Treinamento de redes perceptron multicamadas - algoritmo <i>backpropagation</i>	44
3.3.1 Propagação de um padrão no algoritmo de treinamento	45
3.3.2 Retropropagação do erro	47
3.4 Fundamentos de vetores-alvo	48
3.4.1 Vetores-alvo na aprendizagem de uma MLP	48
3.4.2 Algoritmo de geração de vetores bipolares ortogonais	50

4	Método Proposto	55
4.1	Hipótese de comparação de vetores-alvo - Intersecção entre regiões de convergência	55
4.1.1	Posições relativas entre circunferências	58
4.1.2	Procedimento experimental sobre regiões de convergência	60
4.1.3	Resultados experimentais sobre áreas de convergência	62
5	Análise matemática	68
5.1	Melhoria do desempenho de MLP com o uso de VBOs como vetores-alvo da rede	68
5.2	Atualização dos pesos entre a camada intermediária e a camada de saída	70
5.3	Atualização dos pesos entre a camada de entrada e a camada intermediária	74
5.4	Confirmação da hipótese formulada	83
6	Análise experimental - Classificação de dígitos manuscritos	85
6.1	Introdução	85
6.2	Dificuldades	86
6.3	Motivação	87
6.4	Procedimento experimental	87
6.5	Resultados experimentais	89
6.6	Estatística de comparação	93
6.7	Discussão dos resultados	94
7	Análise experimental - Classificação de imagens de íris humana	96
7.1	Introdução	96
7.2	Dificuldades	97
7.3	Motivação	98
7.4	Procedimento experimental	98
7.4.1	Topologias adotadas	98
7.4.2	Caracterização dos dados experimentais	99
7.4.3	Extração de características	99
7.4.4	Processo de treinamento das redes	100
7.5	Resultados experimentais	103
7.6	Discussão dos resultados experimentais	103
8	Conclusão e sugestões para trabalhos futuros	106
8.1	Conclusão	106
8.2	Agradecimentos	107
8.3	Trabalhos futuros	108
9	Lista de publicações	109
	Referências Bibliográficas	111

Lista de Figuras

1.1	Subdivisão da área de Inteligência Computacional	16
1.2	Fatores envolvidos no treinamento de RNA para RP	18
1.3	Figura ilustrativa da originalidade do trabalho 1	20
1.4	Figura ilustrativa da originalidade do trabalho 2	21
3.1	Exemplo de separador de classes perfeito	27
3.2	Exemplo de separador de classes próximo do ideal	28
3.3	Ilustração das etapas em um sistema de reconhecimento de padrões	29
3.4	Ilustração das principais técnicas de reconhecimento de padrões	31
3.5	Ilustração de um neurônio biológico	33
3.6	Ilustração de um neurônio artificial	35
3.7	Arquitetura de redes de camadas simples	36
3.8	Função lógica OU	37
3.9	Função lógica XOR	37
3.10	Arquitetura de redes multicamadas com realimentação	38
3.11	Arquitetura de rede em reticulado	39
3.12	Função de ativação degrau	40
3.13	Função de ativação degrau bipolar	41
3.14	Função de ativação degrau rampa	42
3.15	Função de ativação logística binária	43
3.16	Função de ativação logística bipolar	43
3.17	Função de ativação tangente hiperbólica bipolar	44
3.18	Função de ativação gaussiana	45
3.19	Distância euclidiana de vetores-alvo	54
4.1	Treinamento de redes com vetores comuns	56
4.2	Treinamento de redes com VBO	56
4.3	Circunferências concêntricas	58
4.4	Circunferências secantes	58
4.5	Circunferências tangentes internas	59
4.6	Circunferências tangentes externas	59
4.7	Circunferências externas	59
4.8	Comparação das invasões entre áreas de convergência	64
4.9	Áreas de convergência para VBC (10)	65
4.10	Áreas de convergência para VNO (16)	65
4.11	Áreas de convergência para VNO (32)	66
4.12	Áreas de convergência para VBO (16)	66

4.13	Áreas de convergência para VBO (32)	66
6.1	Amostra de dígitos manuscritos	88
6.2	Resultados do reconhecimento de dígitos	93
7.1	Early stopping para redes com 71 saídas e VBCs	101
7.2	Early stopping para redes com 128 saídas e VNOs	101
7.3	Early stopping para redes com 128 saídas e VBOs	102
7.4	Comparação do desempenho dos vetores-alvo VBCs, VNOs e VBOs	105

Lista de Tabelas

4.1	Análise estatística da comparação das áreas de convergência de vetores-alvo . . .	63
6.1	Análise do desempenho da MLP, usando VBCs, alfa igual a 0,01 e 200 neurônios escondidos	90
6.2	Análise do desempenho da MLP, usando VNOs, alfa igual a 0,01 e 200 neurônios escondidos	90
6.3	Análise do desempenho da MLP, usando VBOs, alfa igual a 0,01 e 200 neurônios escondidos	90
6.4	Análise do desempenho da MLP, usando VBCs, alfa igual a 0,01 e 100 neurônios escondidos	90
6.5	Análise do desempenho da MLP, usando VNOs, alfa igual a 0,01 e 100 neurônios escondidos	90
6.6	Análise do desempenho da MLP, usando VBOs, alfa igual a 0,01 e 100 neurônios escondidos	91
6.7	Análise do desempenho da MLP, usando VBCs, alfa igual a 0,005 e 200 neurônios escondidos	91
6.8	Análise do desempenho da MLP, usando VNOs, alfa igual a 0,005 e 200 neurônios escondidos	91
6.9	Análise do desempenho da MLP, usando VBOs, alfa igual a 0,005 e 200 neurônios escondidos	91
6.10	Análise do desempenho da MLP, usando VBCs, alfa igual a 0,005 e 100 neurônios escondidos	91
6.11	Análise do desempenho da MLP, usando VNOs, alfa igual a 0,005 e 100 neurônios escondidos	92
6.12	Análise do desempenho da MLP, usando VBOs, alfa igual a 0,005 e 100 neurônios escondidos	92
7.1	Desempenho da rede treinada com VBCs de 71 componentes	103
7.2	Desempenho da rede treinada com VNOs de 128 componentes	104
7.3	Desempenho da rede treinada com VBOs de 128 componentes	104

Lista de Abreviaturas

RNA	Rede neural artificial
MLP	Rede perceptron multicamada
VBO	Vetor bipolar ortogonal
VC	Vetor convencional
VBN	Vetor binário
VBC	Vetor bipolar convencional
VNO	Vetor não ortogonal
RP	Reconhecimento de padrão
VC	Vetor convencional

Capítulo 1

Introdução

1.1 Estado da arte

A inteligência computacional é uma área da ciência que vem se consagrando como um conjunto de ferramentas eficientes e capazes de resolver problemas que antes não podiam ser resolvidos. Dentro desse conjunto de ferramentas podemos citar a computação evolucionária cujo principal representante são os algoritmos genéticos, a lógica nebulosa (também conhecida como lógica *fuzzy*) e as Redes Neurais Artificiais (RNA)s. No caso das RNAs, o grande salto em seu desenvolvimento ocorreu após a década de 1980, quando recebeu importantes contribuições por parte dos pesquisadores e desde então têm se firmado como uma poderosa ferramenta para o reconhecimento de padrões, aproximação de funções e até mesmo para a previsão de séries temporais.

Essa poderosa ferramenta da área de inteligência computacional tem sido aplicada na resolução de diversos problemas. Citar todas as possibilidades de aplicações de RNAs seria um ato demasiadamente ambicioso. Sem a pretensão de fazer injustiças, pode-se citar que as redes do tipo *Multilayer Perceptron* (MLP) são usadas, por exemplo, no reconhecimento de letras e dígitos (GUINGO, RODRIGUES, & THOMÉ, 2002), no reconhecimento de sinais biológicos (SANTOS, SEIXAS, PEREIRA, & MEDRONHO, 2005), de imagens de radiografia (KINOSHITA, MARQUES, & FRÈRE, 2000), de tomografia (ALVES & ARAÚJO, 2005), de

sinais sonoros (ABREU, 2003), de sinais elétricos (PASSOS, 2006) e outros (GALO, 2000), (FERNEDA, 2006) e (GUARNIERI, 2006). Um exemplo bastante interessante é o reconhecedor automático de códigos de placas de veículos de imagens reais captadas por câmeras de controle de tráfego (GUINGO et al., 2002). A Figura 1.1 representa essa subdivisão das técnicas de inteligência computacional.

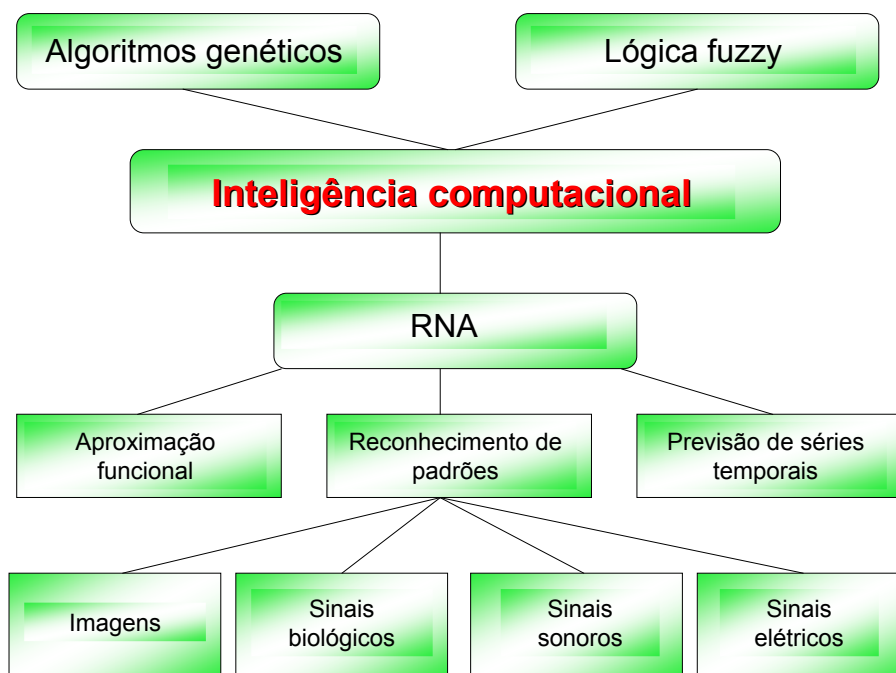


Figura 1.1: Subdivisão da área de Inteligência Computacional

A cognição biológica permite que um indivíduo consiga reconhecer e distinguir padrões, mesmo que eles tenham um elevado grau de degradação em suas características (KOHONEN, 1977), (FU, 1974) e (COOPER, 1973). As RNAs, que têm sua concepção alicerçada no princípio de funcionamento dos neurônios biológicos, buscam a resolução de tarefas que antes eram resolvidas apenas pelo cérebro humano. Uma rede neural artificial, devidamente modelada e com seus parâmetros escolhidos de forma correta, consegue realizar um treinamento de forma generalizada, dando ao modelo a flexibilidade necessária para reconhecer corretamente um padrão que tenha sofrido elevada interferência de ruídos.

1.2 Justificativa

Reconhecimento de padrões (RP) são técnicas estatísticas ou conexionistas que tem por função a separação de elementos em classes ou subconjuntos a partir de determinadas características. Os padrões podem ser oriundos de diversas aplicações, como por exemplo, a íris do olho humano, digitais do dedo humano, face humana, imagens de radiografia, dígitos manuscritos ou não, a fala humana e mensagens de correio eletrônico para citar alguns exemplos.

O conjunto de dados utilizados no reconhecimento de padrões pode ser de boa qualidade, facilitando a tarefa do reconhecimento, ou pode ser de baixa qualidade, quando há a interferência nos dados das características que chamamos de ruídos. Os ruídos constituem-se em um grande problema pelo fato de dificultarem a capacidade da classificação correta dos padrões nas suas respectivas classes.

Os fatores ocasionadores de ruídos são diversos e dependem de cada situação. Em alguns casos eles podem ter origem no próprio sistema de aquisição de características, como por exemplo, dispositivos de captura de imagem com baixa qualidade de resolução. Também há casos em que ainda não existem técnicas mais precisas no modo de captação dos dados. Pode-se citar o caso da captura de imagens de íris de seres humanos, onde a distância da câmera, o deslocamento da posição do olho em relação ao foco ideal e a rotação do rosto da pessoa são variáveis onde há grande dificuldade de controle.

Outro exemplo da interferência de ruídos ocorre na aquisição de imagens de placas de veículos. Por problemas climáticos, de qualidade da imagem ou mesmo por questões de luminosidade, os padrões a serem processados no reconhecimento podem conter uma grande quantidade de ruídos.

Assim sendo, um dos grandes desafios das técnicas de reconhecimento de padrões tem sido o tratamento dos ruídos. No caso das RNAs isso não é diferente. Fazer com que a rede consiga superar esse problema é uma tarefa árdua que exige grande conhecimento e experiência por parte do pesquisador. No caso das RNAs, o ajuste correto é muito importante para dar

flexibilidade ao modelo. Isso é justificado pela própria característica das redes neurais. Como elas agem no sentido de tentativa e ajuste com base no erro, há de se entender que quanto menor ele for mais próximo do padrão de treinamento estará a convergência. Dessa maneira, redes que tenham sido exaustivamente treinadas, terão no processo de classificação pouca tolerância com relação a padrões que apresentem características com diferenças do que ela espera. Se o padrão for muito ruidoso, essas diferenças farão com que o sistema deixe de reconhecê-lo corretamente. Esse problema dentro da área de RNA é conhecido como *over-training*, ou seja, excesso de treinamento. A Figura 1.2 ilustra os fatores envolvidos no treinamento de uma RNA para RP.

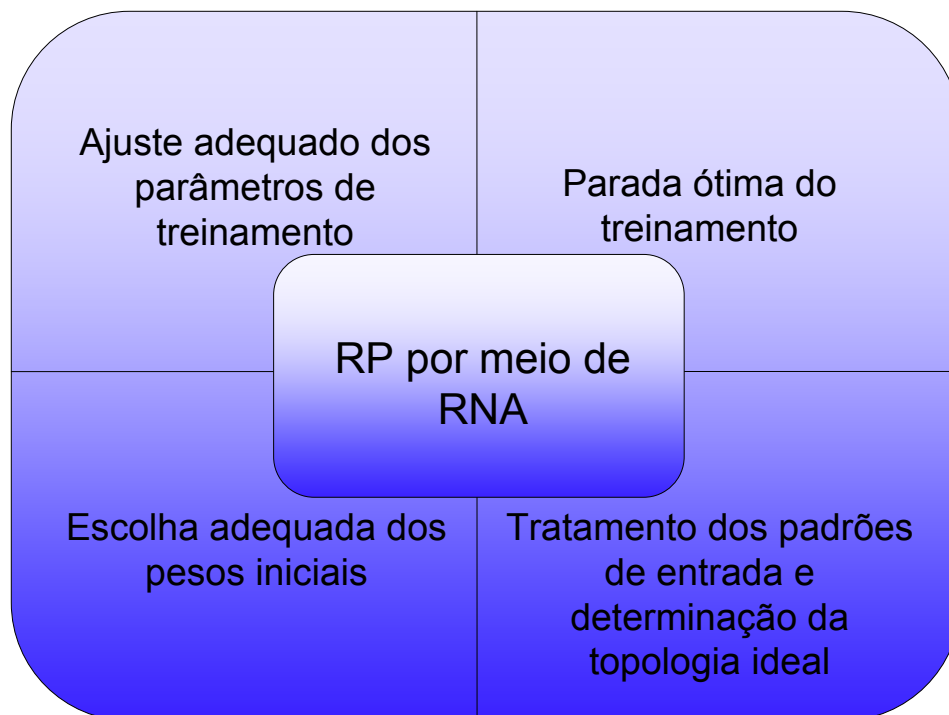


Figura 1.2: Fatores envolvidos no treinamento de RNA para RP

Nesse sentido, vários pesquisadores (WANG, CHANG, & DU, 2004), (COSTA, BRAGA, & MENEZES, 2002/2003), (LEE, YANG, & HO, 2006), (KIM, 2005) e (CHEN & PUNG, 2008) têm concentrado esforços para aperfeiçoar as RNAs bem como para obter melhores desempenhos. Esses esforços ocorrem de maneiras variadas e em diferentes pontos.

Melhorar os padrões de entrada, aperfeiçoar a arquitetura da RNA, selecionar melhores pesos sinápticos iniciais e aperfeiçoar o algoritmo de aprendizagem, são exemplos de procedimentos realizados pela comunidade científica. Todas essas iniciativas têm como objetivo a obtenção de redes mais flexíveis, robustas e eficientes no tratamento dos ruídos. A importância dessas pesquisas é facilmente justificada ante as mais variadas aplicações, as quais essa importante ferramenta se destina.

1.3 Originalidade

Nesse cenário algumas pesquisas já realizadas (NOMURA, YAMANAKA, KATAI, KAWAKAMI, & SHIOSE, 2004), (NOMURA, YAMANAKA, KATAI, KAWAKAMI, & SHIOSE, 2005), (NOMURA, YAMANAKA, & MANZAN, 2010) e (MANZAN, NOMURA, & YAMANAKA, 2011) tem se dedicado a estudar o espaço de saída de RNAs do tipo MLP. Os estudos em questão apontam que se os vetores-alvo da rede forem Vetores Bipolares Ortogonais (VBOs), a rede terá um treinamento mais eficiente. Isso se torna evidente ao observarmos que esse tipo de vetor possui maior distância entre si, facilitando então o mapeamento dos padrões a serem reconhecidos. Também verifica-se um estudo sobre a influência de vetores-alvo na velocidade do treinamento de RNAs do tipo MLP para o reconhecimento de voz (ROBERTS, 1989).

Se os vetores-alvo forem associados a pontos no espaço R^n , que é o espaço de saída da rede, quanto maior for a distância entre eles, mais fácil será a tarefa do treinamento e classificação. Vetores-alvo próximos entre si sofrem com o problema da grande interseção entre suas regiões de convergência. Reduzir essa interseção torna o treinamento mais simples e fácil para a MLP. Essa maior distância indica um menor grau de similaridade o que no paradigma das redes neurais, significa maior diferenciação dos padrões de treinamento (KOHONEN, 1977), (FU, 1974) e (COOPER, 1973). Contudo, tais pesquisas apresentam apenas resultados experimentais.

Além dessas pesquisas não têm sido encontradas contribuições de estudos que se des-

tinem a estudar a influência dos vetores-alvo na aprendizagem da rede. Este trabalho propõe a utilização dos VBOs como vetores-alvo de Redes Perceptron Multicamadas (MLPs), de maneira a estudar matematicamente as evidências que explicam a melhoria do seu desempenho através dessa nova abordagem. Além de evidências matemáticas, também é proposto o estudo de novas análises experimentais. As Figuras 1.3 e 1.4 ilustram a originalidade do trabalho.

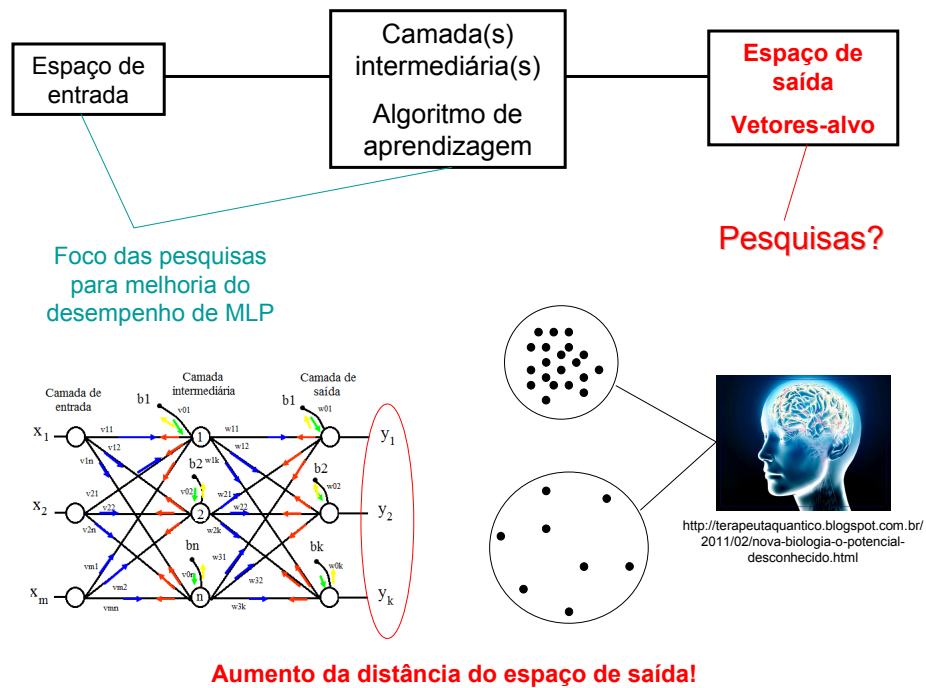


Figura 1.3: Figura ilustrativa da originalidade do trabalho 1

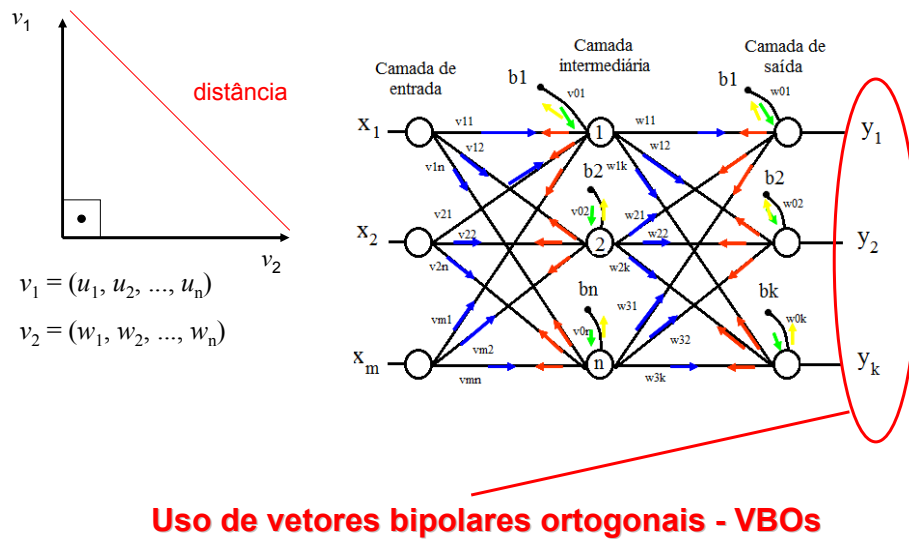


Figura 1.4: Figura ilustrativa da originalidade do trabalho 2

1.4 Estrutura do trabalho

Resultados experimentais relativos a essa nova abordagem já foram apresentados por Nomura et al. (2004,2005), Nomura, Manzan e Yamanaka (2010) e Manzan, Yamanaka e Nomura (2011). Eles tem comprovado a melhoria da performance da rede de maneira experimental. Neste trabalho, apresentam-se evidências matemáticas e novos resultados experimentais que explicam essa melhora de desempenho da MLP. A abordagem deste trabalho é um estudo matemático que mostra evidências para a hipótese levantada, ou seja, a melhora no desempenho de redes do tipo MLP com a utilização de vetores bipolares e ortogonais.

No capítulo 2 são apresentados os trabalhos relacionados. No capítulo 3 encontram-se os fundamentos teóricos de reconhecimento de padrões e de redes neurais artificiais. No capítulo 4 são apresentadas as hipóteses formuladas para o problema juntamente com o método proposto. No capítulo 5 são mostrados os argumentos matemáticos. Nos capítulos 6 e 7 são mostrados resultados experimentais. Finalmente, a conclusão, os agradecimentos e as sugestões

para trabalhos futuros são mostrados no capítulo 8.

Capítulo 2

Trabalhos relacionados

2.1 Mudança na escolha dos pesos sinápticos

Trabalhos destinados à otimização do treinamento de RNAs vêm sendo realizados, sempre na tentativa de melhorar a capacidade de reconhecimento da rede. Uma das propostas destina-se a uma nova metodologia de escolha dos pesos sinápticos iniciais (KIM, 2005) para redes *Multilayer Perceptron*. Para esta rede, os pesos sinápticos iniciais devem ser valores aleatórios pequenos, onde alguns textos sugerem que estejam entre - 1 e 1 e outros indicam a escolha entre - 0,5 e 0,5. Dependendo dos valores que são sorteados a rede pode ter um mal ou bom desempenho. Isso é explicado pelo fato de que alguns pesos fazem com que a curva do erro quadrático médio seja conduzida a mínimos globais, elevando o número de épocas e consequentemente reduzindo a capacidade de generalização da rede. Dessa maneira, a abordagem proposta por Kim (2005) realiza cinco treinamentos de uma mesma rede com pesos diferentes para cada um deles. São selecionados os treinamentos e é calculada a diferença entre os dois melhores resultados. Se a diferença for menor que 0,01, os pesos da rede de maior desempenho serão selecionados. Caso contrário, o algoritmo é repetido até que se obtenha essa condição.

2.2 Aperfeiçoamento da arquitetura da rede

A topologia da MLP também afeta de forma significativa o seu desempenho. Nesse ponto pode-se destacar avanços no trabalho de Costa et al. (2003). Treinamentos demorados afetam de forma significativa a curva do erro e, por conseguinte o desempenho da rede no sentido de reconhecer corretamente os padrões de teste. Essa demora pode ter várias origens e uma delas é a topologia que quando é muito grande pode delongar o treinamento. Esse trabalho utiliza as funções do erro quadrático médio, da função normal e do número de neurônios escondidos para estabelecer uma função multi-objetivo que permita a determinação ideal da topologia a ser utilizada. Os resultados demonstram a simplificação da topologia sem perda da capacidade de generalização.

Outra proposta é a determinação do número de neurônios da camada intermediária (CHEN & PUNG, 2008). Nela, os autores apontam a utilização de um algoritmo denominado de *convex incremental algorithm* (CI-ELM) e a sua evolução. O algoritmo acrescenta aleatoriamente neurônios na camada escondida, e em seguida determina os pesos sinápticos que ligam essa camada à camada de saída. A cada acréscimo os pesos sinápticos são congelados e o algoritmo evolui até que se obtenha melhoras no desempenho. Além disso, é sugerida uma adaptação do método com a substituição de todos os pesos sinápticos que, de acordo com resultados experimentais, tem mostrado melhores quedas do erro residual.

2.3 Aperfeiçoamento do algoritmo de treinamento *backpropagation*

Há também iniciativas de mudança no algoritmo de treinamento *backpropagation* Lee et al. (2006). No algoritmo usual, o erro é obtido através da comparação entre a saída da rede com a saída esperada, ou seja, no processo de supervisão do treinamento. Esse erro é propagado da camada de saída até a camada de entrada com o intuito de ajustar os pesos sinápticos da rede.

Na abordagem citada, a camada intermediária é subdividida em duas sub-MLPs e

cada uma das sub-MLPs é otimizada pelo próprio algoritmo. Dessa maneira, toda a MLP é ajustada por dois algoritmos independentes. Essa nova abordagem é usada com o intuito de lidar com o problema da equalização de canal. Os resultados apresentam melhoras significativas, especialmente em ambientes com distorção não-linear, desvio de frequência e erros de fase e temporização.

2.4 Análise de sensibilidade do sinal

Uma metodologia de análise de sensibilidade do sinal é proposta por Wang et al. (2004). Em redes MLPs comuns, o ruído pode causar grandes interferências no sinal de saída. Se a rede não é robusta o suficiente para lidar com este problema, o erro proveniente desse ruído causará grandes efeitos no processo de classificação dos padrões.

O objetivo dessa proposta consiste na seleção da rede MLP ideal de um conjunto de redes MLPs com diferentes parâmetros, ou seja, selecionar a MLP com maior precisão e robustez para problemas de controle. A hipótese defendida é que a sensibilidade do sinal depende da variação do erro de saída oriunda dos ruídos nos sinais de entrada da MLP. Nessa nova metodologia é utilizado um método para o cálculo de perturbações na curva de convergência nas várias redes MLPs no intuito de selecionar o melhor modelo de treinamento. Assim sendo, os ruídos são minimizados e a RNA terá condições de obter a capacidade necessária de generalização na classificação de padrões.

Capítulo 3

Fundamentos teóricos

3.1 Fundamentos de reconhecimento de padrões

3.1.1 Conceitos de um sistema de reconhecimento de padrões

Entende-se por reconhecimento de padrões RPs, o conjunto de técnicas capaz de separar objetos em conjuntos ou classes. 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 & WOODS, 1992). Essa descrição pode ser feita por uma ou mais medidas que são denominadas como atributos ou características do padrão. Um conjunto de padrões com características semelhantes é denominado como classe (GONZALEZ & WOODS, 1992).

Uma técnica de reconhecimento de padrões tem por concepção a seleção de características de conjuntos de objetos e a separação dos objetos em suas devidas classes. De acordo com Duda et al. (2001), são técnicas que 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 problema simples ou complexo, busca-se sempre chegar próximo do separador ideal. Entretanto

em boa parte dos problemas de classificação a determinação do separador ideal exige muito esforço, sendo, em alguns casos, inviável. A Figura 3.1 representa um reconhecedor ideal capaz de separar duas classes 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 3.2 representa um reconhecedor próximo do ideal, capaz de separar duas classes com alguns erros de separação.

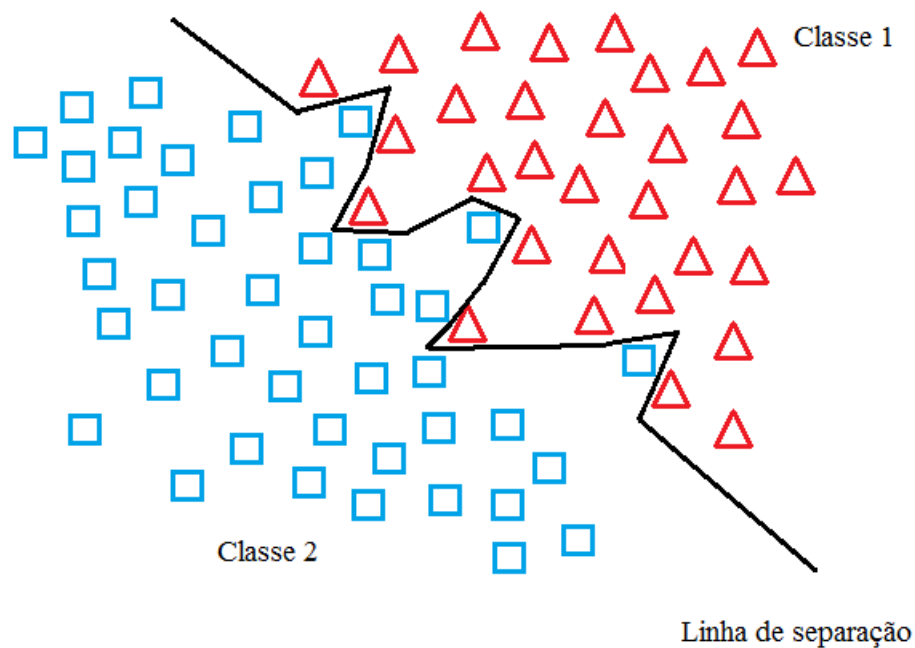


Figura 3.1: Exemplo de separador de classes perfeito
Fonte: (DUDA, HART, & STORK, 2001)

3.1.2 Etapas de um sistema de reconhecimento de padrões

De acordo com Duda et al. (2001) os sistemas de RPs seguem normalmente um conjunto de etapas como é sugerido pela Figura 3.3.

- A primeira etapa consiste na aquisição das características dos padrões de treinamento. As técnicas de aquisição são inúmeras e sua escolha depende principalmente do tipo de aplicação que está sendo trabalhada;
- A segunda etapa consiste na segmentação que nada mais é do que a separação das

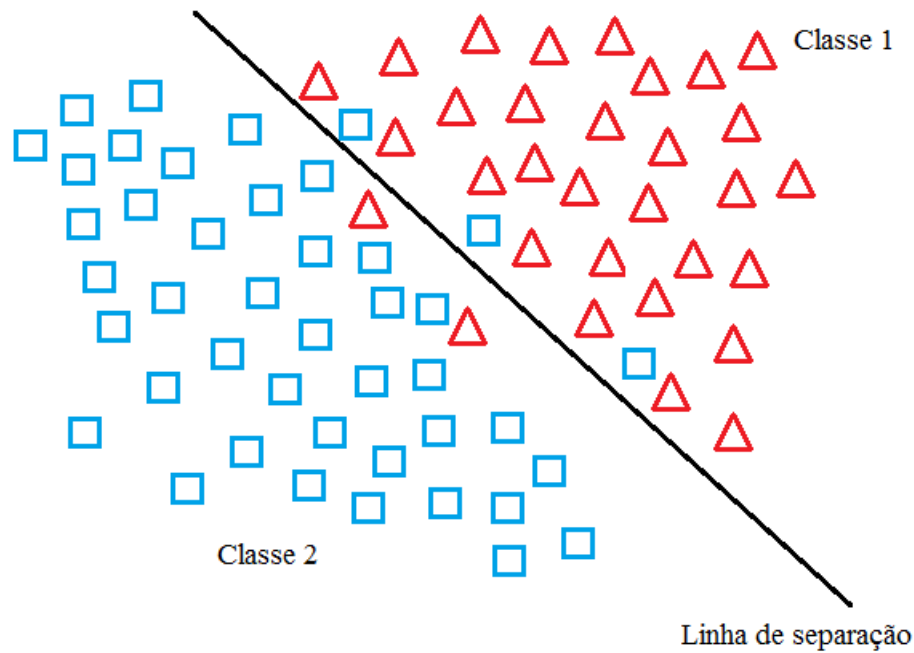


Figura 3.2: Exemplo de separador de classes próximo do ideal
 Fonte: (DUDA et al., 2001)

informações que tem relevância para o sistema das informações irrelevantes. Essa é uma das etapas mais complexas do sistema de RPs (DUDA et al., 2001);

- Na etapa de extração de características que é a terceira, de acordo com o modelo proposto, são formados os conjuntos de medidas ou valores que formarão as classes de objetos. Um bom processo de extração de características poderá facilitar e muito a tarefa do classificador. Uma boa extração é aquela capaz de resumir ao máximo as características relevantes e de fácil extração.
- A quarta etapa consiste na classificação dos padrões. Essa etapa pode ser simples ou complexa dependendo da natureza dos dados. Grandes diferenças entre objetos de uma mesma classe, aliada a uma pequena diferença entre classes diferentes, pode fazer com que essa etapa tenha um alto grau de dificuldade. Isso justifica a importância da execução correta das etapas anteriores. Também deve-se citar que é muito importante a escolha correta do tipo de classificador para o problema em questão (DUDA et al., 2001);

- A quinta etapa corresponde ao pré-processamento;
- A sexta etapa, inexistente em alguns sistemas corresponde à tomada de decisão a partir da tarefa de classificação.

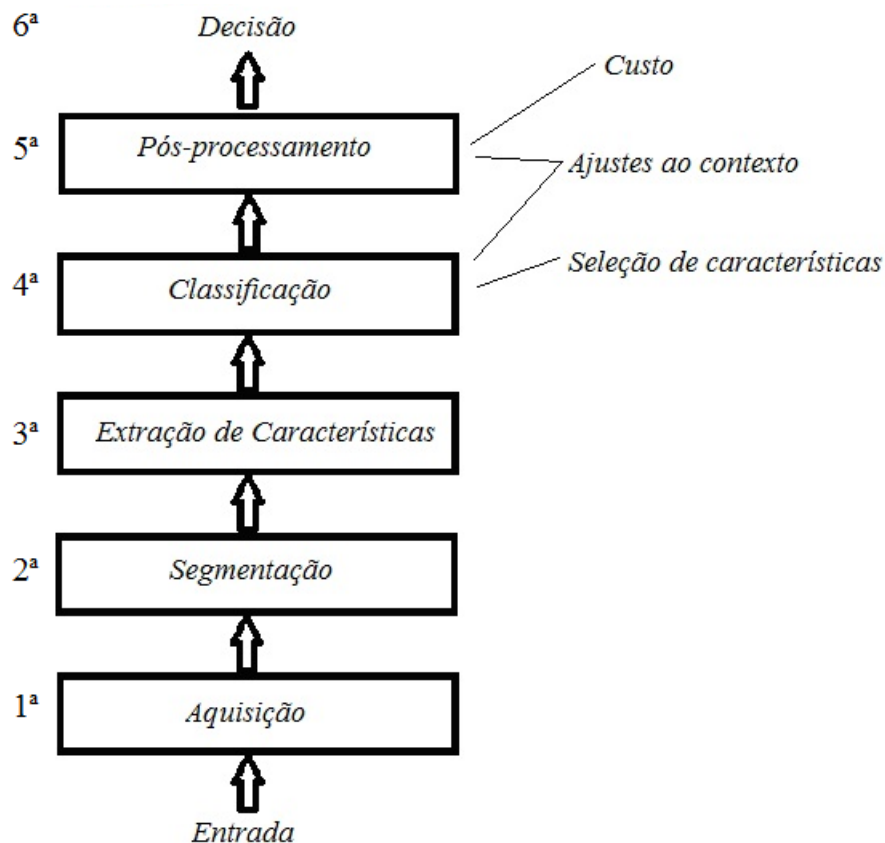


Figura 3.3: Ilustração das etapas normalmente envolvidas em um sistema de reconhecimento de padrões
 Fonte: (DUDA et al., 2001)

3.1.3 Técnicas para classificação de padrões

Existem dois grandes grupos de técnicas para classificação de padrões: treinamento supervisionado e treinamento não-supervisionado. Quando as classes de um conjunto de dados já são conhecidas a técnica de RPs deve apontar no treinamento a saída esperada para cada padrão de treinamento. Assim, o conjunto de dados de treinamento representa exemplos que farão o treinamento do sistema por experiência. No caso da classificação não-supervisionada, as classes não estão bem definidas e o próprio sistema agirá no sentido de separar essas classes

levantando as características mais evidentes.

Existe ainda uma etapa que poderia ser acrescentada ao modelo proposto da Figura 3.3 que consiste na validação do sistema. Essa etapa é importante para verificar se os resultados obtidos atendem à aplicação em questão. Na validação, os dados de teste são submetidos ao classificador a fim de se avaliar preliminarmente o desempenho do sistema ainda que de forma super-estimada. Caso a validação seja bem sucedida, pode-se passar para a etapa de classificação, onde o sistema será capaz de mapear objetos da mesma natureza nos dados de treinamento.

Dentre as técnicas de reconhecimento de padrões mais utilizadas, podemos citar a abordagem estatística, sendo a paramétrica e não-paramétrica e a abordagem conexionista que utiliza as redes neurais artificiais. 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. Dessa maneira, 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 e os histogramas (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, o que de acordo com Duda et al. (2001) é um tipo de classificador ótimo, pois minimiza a probabilidade média de erro na classificação. Existem também a função de discriminação e a regra Naive de

Bayes.

No caso da abordagem conexionista utiliza-se as RNAs. Trata-se de 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. Trabalhos destinados ao estudo do reconhecimento de padrões, tanto no que se refere à descrição, quanto para classificação, vem ganhando grande destaque no campo da computação. Atualmente, as redes neurais artificiais têm se tornado uma técnica amplamente utilizada em razão de resultados bastante satisfatórios e promissores (DUDA et al., 2001). A Figura 3.4 resume as principais técnicas de reconhecimento de padrões.

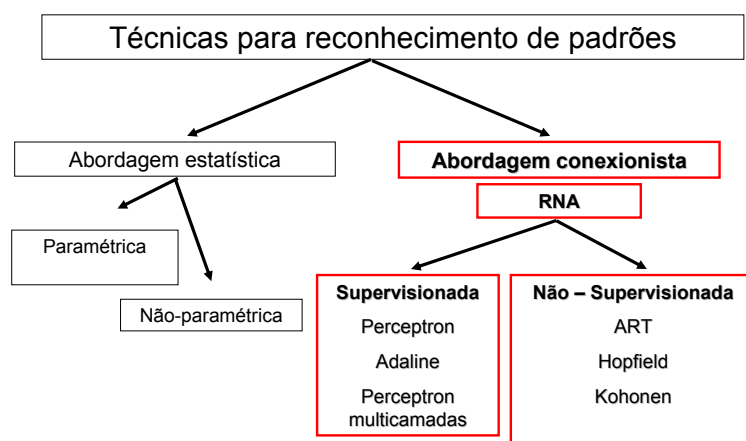


Figura 3.4: Ilustração das principais técnicas de reconhecimento de padrões

3.2 Técnicas de redes neurais artificiais

De acordo com Silva et al. (2010) o primeiro registro do surgimento das RNAs se deu com uma publicação de um artigo de McCulloch e Pitts em 1943 (MCCULLOCH & PITTS, 1943). Essa nova área da computação tinha por concepção o funcionamento dos neurônios biológicos. Segundo Silva et al. (2010) em 1949 Hebb apresentou o primeiro método de treina-

mento para redes neurais artificiais (HEBB, 1949). Entre 1957 e 1958 Frank Rosenblat desenvolveu o primeiro neurocomputador e no período de 1958 a 1962 criou uma grande classe de redes neurais denominada como perceptrons (FAUSETT, 1994).

Em 1960 Widrow e Hoff desenvolveram um novo tipo de rede neural artificial denominada como ADALINE (Adaptive Linear Element) que posteriormente recebeu aperfeiçoamentos correspondentes a associação de múltiplas redes Adaline, resultando no nome de MADALINE (FAUSETT, 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, 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, FLAUZINO, & ANDRADE, 2010). Nesse período de relativa turbulência das pesquisas envolvendo redes neurais, foi implementada a rede ART (Adaptive Resonance Theory). Num trabalho de Grossberg em 1980, foi realizada a formulação de mapas auto-organizáveis por 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 redes neurais artificiais conseguiu realmente se estabelecer a partir da publicação do livro de Rumelhart, Hinton e Williams com o título de *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).

3.2.1 Neurônio biológico

A Figura 3.5 montada com base na Figura 1.1 de Silva et al. (2010) ilustra um neurônio biológico. O neurônio biológico é dividido em três partes principais que são os dendritos, a soma ou corpo celular e o axônio.

Os dendritos captam sinais elétricos oriundos de outros neurônios ou mesmo do meio externo ao qual 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).

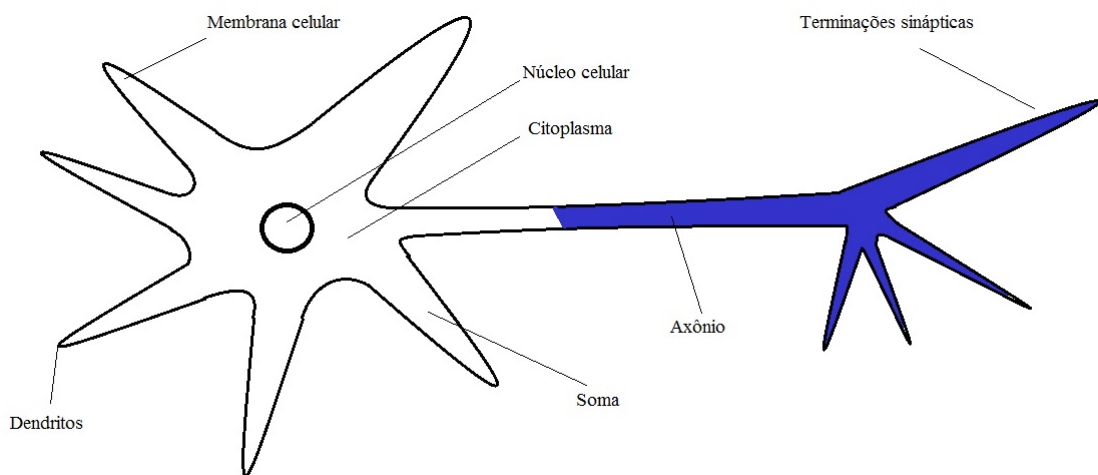


Figura 3.5: Ilustração de um neurônio biológico

Fonte: (SILVA et al., 2010)

O axônio constitui-se em um único prolongamento que conduz impulsos elétricos para outros neurônios. No axônio existem ramificações que denominamos como terminações sinápticas. O envio de informação das sinapses para os dendritos de outros neurônios ocorre através de substâncias neurotransmissoras o que explica o fato de que entre neurônios distintos não existe ligação física. Essas substâncias neurotransmissoras além de transmitir os impulsos elétricos, realizam a ponderação da informação (FAUSETT, 1994).

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

3.2.2 Neurônio artificial

Um neurônio artificial corresponde a um modelo bem simplificado do neurônio biológico. De modo semelhante ao que ocorre com neurônios biológicos, os neurônios artificiais recebem informação, processam de acordo com seu papel dentro da rede e enviam uma nova informação para outros neurônios ou para a saída do sistema (SILVA et al., 2010). O modelo proposto por MucCulloch e Pitts tem em sua concepção de funcionamento o processamento paralelo da informação com alta conectividade. Esse tipo de modelo é ainda o mais utilizado nos modelos de RNAs. A Figura 3.6 representa um neurônio artificial e foi elaborada com base na Figura 1.5 proposta por Silva et al. (2010).

De forma análoga ao neurônio biológico a informação chega ao neurônio com valores que são simbolizados pela variável x para uma das suas n entradas. Outra semelhança ocorre no fato de que o neurônio artificial também pondera a informação recebida através de valores denominados como pesos sinápticos. Aqui esses valores estão simbolizados pela letra v para as n entradas do neurônio. Toda essa informação é reunida através de um somatório que processa junto com ela um valor denominado como *bias* (FAUSETT, 1994). Após a reunião, essa informação recebe a ação da função de ativação, cuja existência também é análoga ao neurônio biológico. Finalmente, após a ação da ativação, o neurônio envia essa informação para outro neurônio ou para a saída.

3.2.3 Rede neural artificial

As abordagens clássicas da Inteligência Artificial trabalham com processamento sequencial. As redes neurais utilizam um modo de aprendizagem cujo processamento é distribuído e paralelo. Conforme o exposto anteriormente, sua metodologia de treinamento é baseada no funcionamento dos neurônios biológicos, onde a aprendizagem ocorre através de exemplos, fazendo com que a tentativa e o erro desencadeie a apropriação da habilidade de diferenciar padrões. RNAs realizam trabalho semelhante quando um grande número de neurônios envia sinais inibitórios ou excitatórios a outros neurônios da rede.

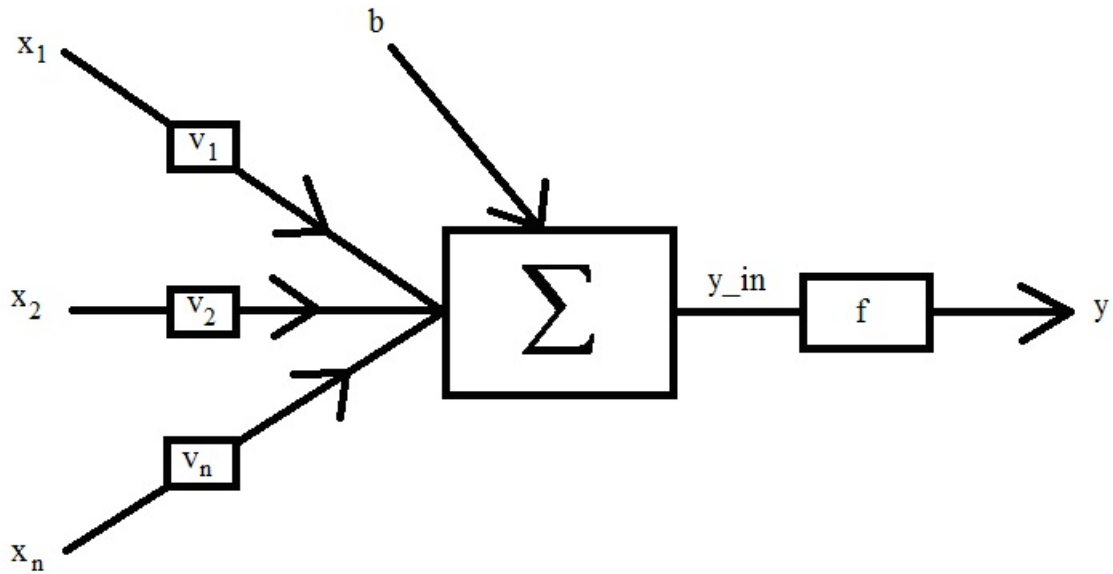


Figura 3.6: Ilustração de um neurônio artificial com base na Figura 1.5 de Silva et al. (2010)
 Fonte: (SILVA et al., 2010)

Basicamente, pode-se entender as RNAs como mecanismos capazes de receber o sinal de determinado padrão na sua entrada, analisá-lo e então nos informar sobre qual classe ele pertence. Isso é possível após o treinamento da rede que pode acontecer de forma supervisionada ou não-supervisionada em várias possibilidades de arquitetura e algoritmos de treinamento.

Há vários tipos de redes neurais artificiais. Algumas possuem uma arquitetura mais simples onde 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 camadas intermediárias ou escondidas. Nessas redes a camada de entrada tem a função do recebimento dos sinais advindos dos padrões de treinamento. A associação das características, bem como a separação das classes, é feita pelos neurônios da(s) camada(s) intermediária(s). Fica a cargo da camada de saída a apresentação dos resultados finais da rede.

Nas redes neurais mais simples a camada de separação das classes é a própria camada de saída como mostra a Figura 3.7. Esses tipos de redes são hábeis para problemas que tem classes linearmente separáveis. São representantes dessa abordagem as redes Perceptron e Adaline.

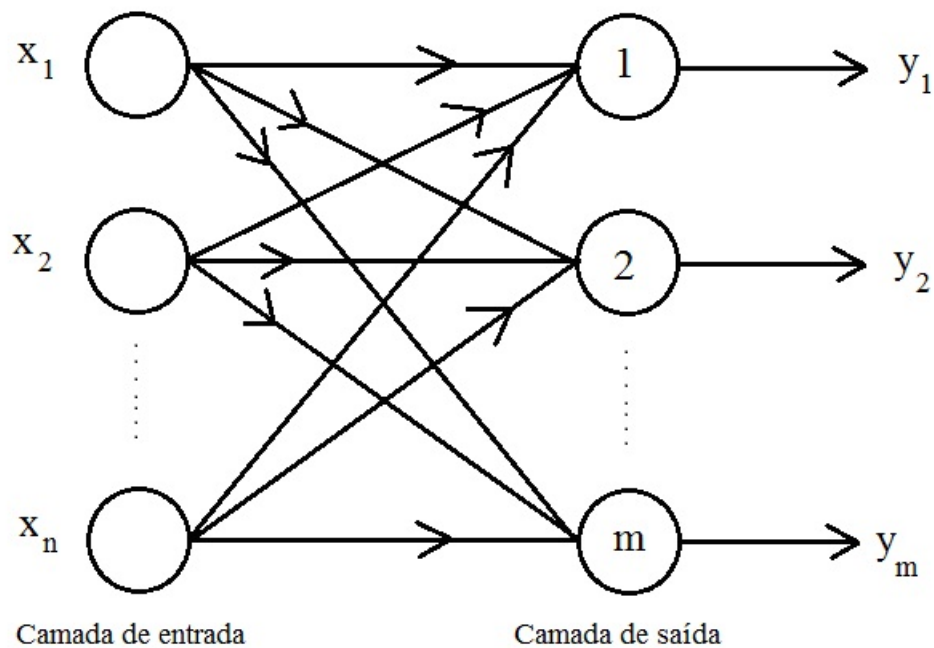


Figura 3.7: Arquitetura de redes de camadas simples
 Fonte: (FAUSETT, 1994)

Veja um exemplo simples da função lógica “ou” na Figura 3.8. Considere-se que “1” seja o valor lógico para verdadeiro e que “-1” seja o valor lógico para falso. Essa operação é realizada sempre entre dois valores lógicos que podem ser verdadeiro ou falso. O resultado dessa operação é sempre verdadeiro exceto para o caso em que a operação “ou” seja realizada entre dois valores lógicos falsos, que no nosso caso figuram como “1” ou “-1”. Resultados verdadeiros estão representados em quadrado e o único resultado falso está representado em triângulo. Com uma reta, conseguimos separar as duas classes, ou seja, os valores lógicos verdadeiros do valor lógico falso.

Entretanto, há problemas em que as classes não são linearmente separáveis. Dessa maneira, redes como a Perceptron e Adaline são incapazes de realizar a classificação. Isso foi mostrado por Minsk e Papert, ao qual já foi citado anteriormente. Consideremos a função lógica “ou exclusivo” também denominada por XOR. Diferentemente da operação lógica “ou” mencionada anteriormente, essa função terá por resultado valores lógicos verdadeiros, quando os valores lógicos da operação forem diferentes. Quando os valores lógicos envolvidos pela operação forem iguais, o resultado será o valor lógico falso. Pode-se perceber que na Figura

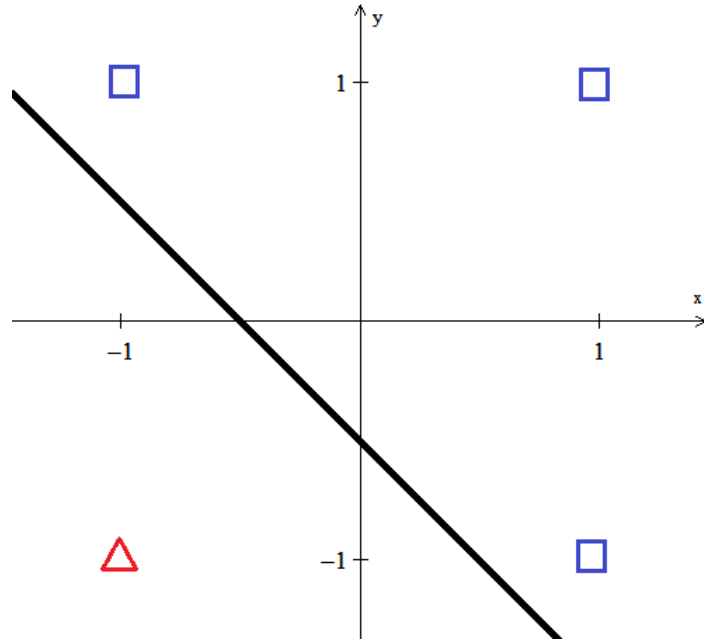


Figura 3.8: Representação da função lógica “ou” e a reta de separação das classes

3.9 que não é possível obter uma reta que separe a classe dos valores lógicos positivos da classe dos valores lógicos negativos.

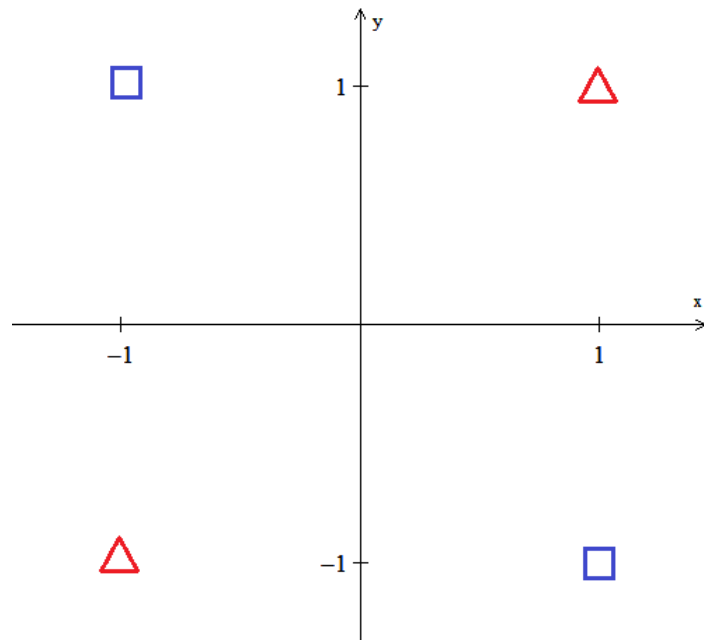


Figura 3.9: Representação da função lógica “ou exclusivo” (XOR)

Problemas dessa alçada requerem redes com mais de duas camadas, ou seja, redes que tenham pelo o menos uma camada intermediária. Os exemplos mais conhecidos de redes

com múltiplas camadas são as redes perceptron multicamadas, popularmente conhecidas como MLPs e as redes de Hopfield. Essas redes usam o princípio da realimentação dos sinais da saída para a atualização dos pesos sinápticos. A Figura 3.10 ilustra a arquitetura desse tipo de rede com a indicação da propagação e retro-propagação do sinal. A inclusão de uma ou mais camadas intermediárias possibilita a separação não-linear das classes. Isso resolve o problema do XOR e muitos outros de complexidade ainda maior. Como consequência, as redes com esse tipo de arquitetura tornaram-se as mais utilizadas pelos pesquisadores.

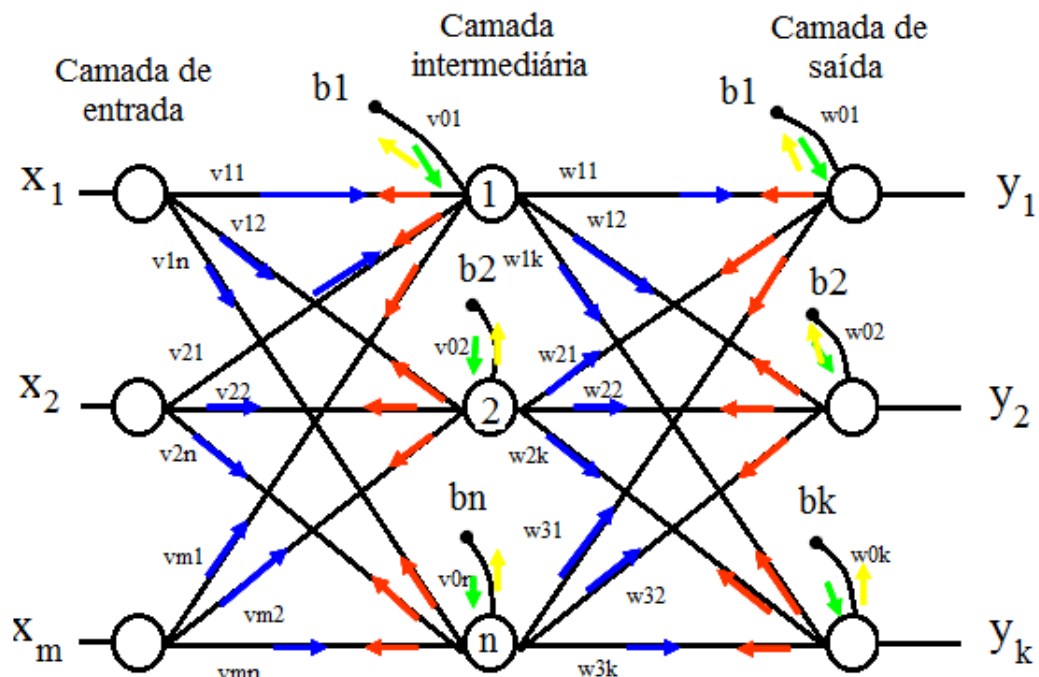


Figura 3.10: Arquitetura de redes multicamadas com realimentação
 Fonte: (SILVA et al., 2010)

Em contrapartida essas redes tem algoritmos bem mais complexos de serem implementados quando comparamos com as redes Perceptron e Adaline e exigem muito mais experiência e atenção do pesquisador para que produzam bons resultados.

Há também redes neurais com uma estrutura denominada como reticulada, cuja representante principal é a rede de Kohonen (SILVA et al., 2010). Nessa configuração os neurônios ficam dispostos no espaço bidimensional como ilustrado na Figura 3.11.

Além da arquitetura e dos tipos de redes neurais existentes, um ponto de extrema

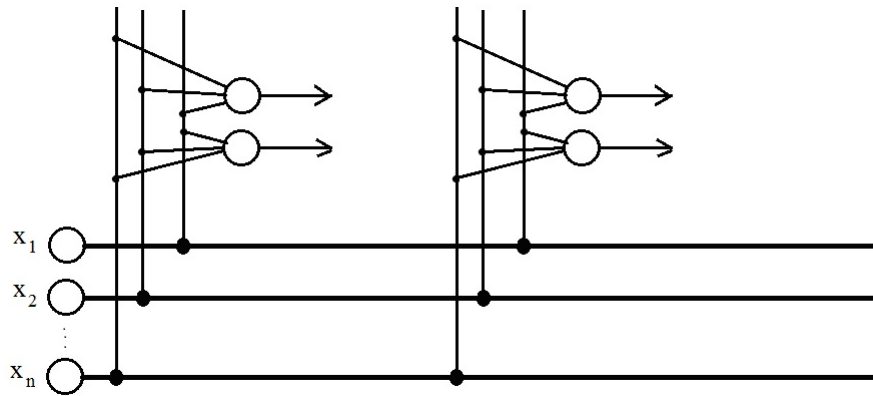


Figura 3.11: Arquitetura de rede em reticulado
 Fonte: (SILVA et al., 2010)

importância é a forma de aprendizado das redes. Uma rede neural pode ter seu treinamento executado de duas maneiras: supervisionada e não-supervisionada como citado anteriormente. No caso de redes supervisionadas, para cada padrão de treinamento há uma saída esperada, ou um alvo a ser atingido. Por causa desse princípio, as saídas desejadas são denominadas como vetores-alvo da rede.

Redes supervisionadas executam o treinamento de maneira a reduzir a diferença entre a saída obtida e a saída esperada para aquele padrão em cada passo da execução. Em outras palavras, diz-se que a cada execução, época ou ciclo, o treinamento procura reduzir o erro proveniente da comparação entre alvo e saída. Após atingir um erro tolerável que irá depender de cada problema, o treinamento é encerrado. Redes perceptron, Adaline e perceptron multicamadas são exemplos de redes supervisionadas.

Opondo-se ao treinamento supervisionado, no treinamento não-supervisionado não há saídas esperadas. A rede se auto-organiza identificando durante o treinamento as similaridades entre elementos. A partir dessas similaridades, subconjuntos são criados de maneira que os pesos sinápticos são ajustados com o intuito de propiciar a separação interna dos elementos de cada subconjunto. Representantes desse tipo de aprendizado são as redes de Hopfield e de Kohonen.

3.2.4 Funções de ativação

As funções de ativação são responsáveis por limitarem o sinal da saída a um intervalo de interesse (SILVA et al., 2010). Algumas funções têm imagem limitada ao intervalo real $[0,1]$ enquanto outras ao intervalo real $[-1,1]$. Também são utilizadas funções com mais de uma sentença matemática quando o tipo de rede realiza a limiarização para o sinal de saída. O tipo de função de ativação depende do tipo de rede adotado. A Figura 3.12 representa o gráfico da função degrau da equação 3.1. Essa função direciona saída “1” para valores líquidos maiores do que zero e saída “0” para valores líquidos menores do que “0”.

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (3.1)$$

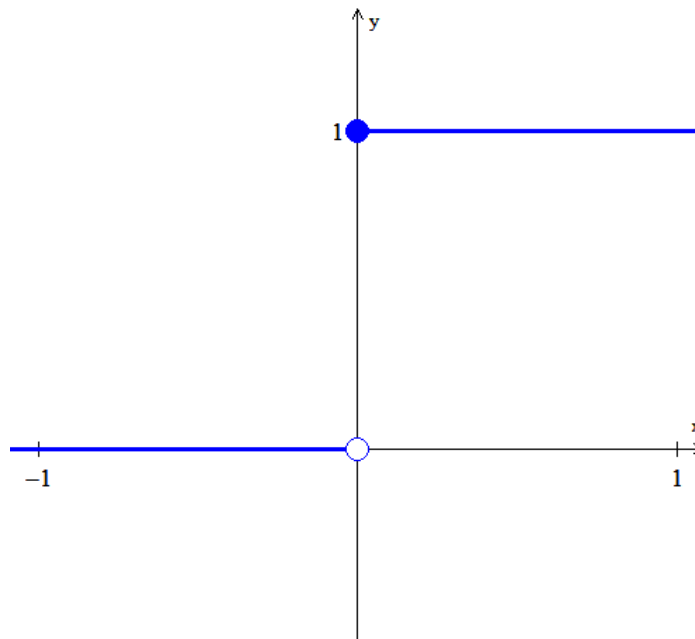


Figura 3.12: Função de ativação degrau
Fonte: (SILVA et al., 2010)

Outra função definida por mais de uma sentença matemática é a função degrau bipolar dada pela equação 3.2 e representada pela Figura 3.13. Nessa função, valores líquidos menores do que zero, recebem saída “-1” e valores líquidos maiores ou iguais a zero recebem saída “1”.

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases} \quad (3.2)$$

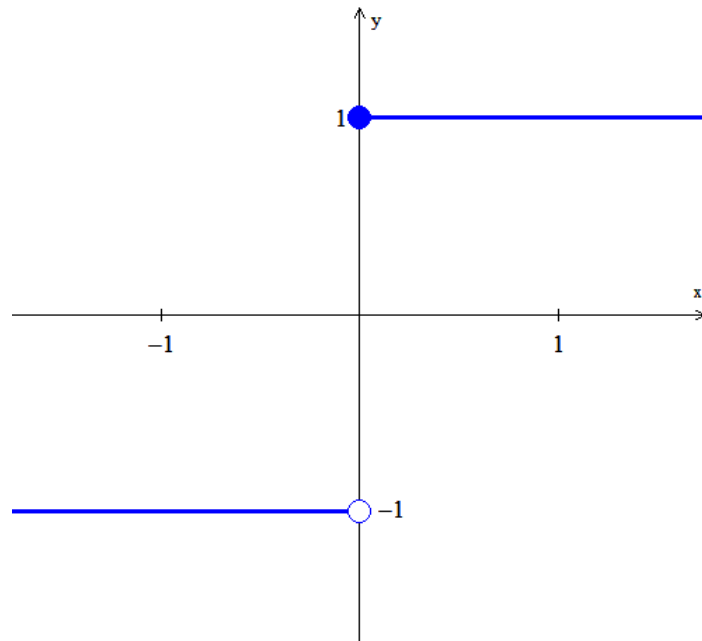


Figura 3.13: Função de ativação degrau bipolar
Fonte: (SILVA et al., 2010)

Em alguns casos é necessário utilizar outros intervalos de saída. A função rampa dada pela equação 3.3 permite que de acordo com a aplicação, definam-se os limites do intervalo de saída. Além disso, essa função considera como saída da rede os próprios valores líquidos quando os mesmos estão dentro dos limites estipulados. A Figura 3.14 representa o gráfico dessa função.

$$f(x) = \begin{cases} -a & \text{se } x < a \\ x & \text{se } -a \leq x \leq a \\ -a & \text{se } x > a \end{cases} \quad (3.3)$$

As funções mostradas anteriormente são deriváveis nos intervalos onde não ocorre o degrau ou a mudança de lei matemática. As funções que serão mostradas a seguir são deriváveis ou diferenciáveis em todo o intervalo real. A primeira função com essa característica é a logística binária dada pela equação 3.4. Essa função assume valores reais entre “0” e “1” e

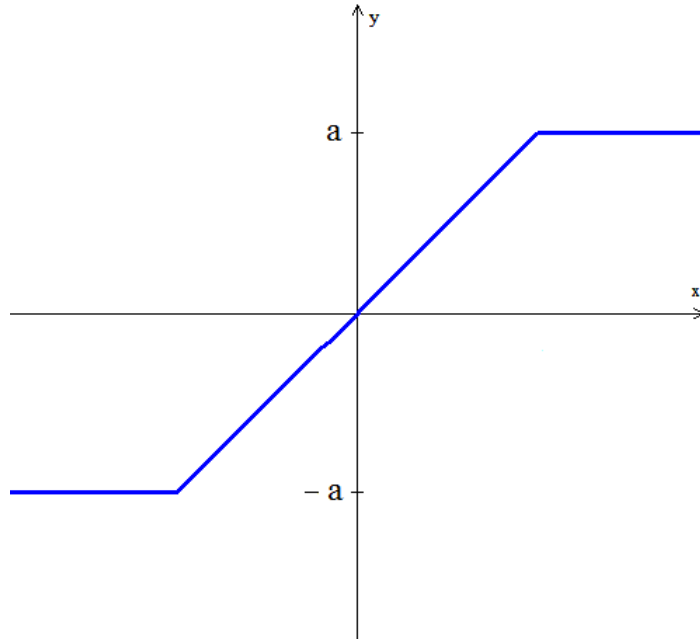


Figura 3.14: Função de ativação degrau rampa
 Fonte: (SILVA et al., 2010)

β interfere no nível de inclinação dessa função em relação ao ponto de inflexão como mostra a Figura 3.15.

$$f(x) = \frac{1}{1 + e^{-\beta \cdot x}} \quad (3.4)$$

Há também a função logística bipolar dada pela equação 3.5. A curvatura é semelhante à logística binária, porém essa função tem a imagem limitada no intervalo real $[-1,1]$. Seu gráfico está representado na Figura 3.16.

$$f(x) = \frac{2}{1 + e^{-\beta \cdot x}} - 1 \quad (3.5)$$

Outra função de ativação com imagem limitada pelo intervalo real $[-1,1]$ é a função tangente hiperbólica dada pela equação 3.6. Seu gráfico é mostrado na Figura 3.17.

$$f(x) = \frac{1 - e^{-\beta \cdot x}}{1 + e^{-\beta \cdot x}} \quad (3.6)$$

Ainda temos a função de ativação gaussiana dada pela equação 3.7 que de acordo com

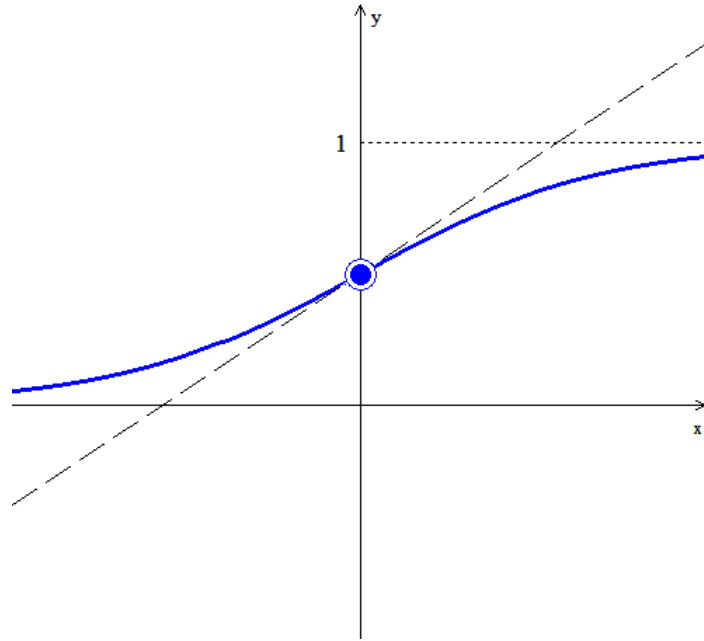


Figura 3.15: Função de ativação logística binária
 Fonte: (SILVA et al., 2010)

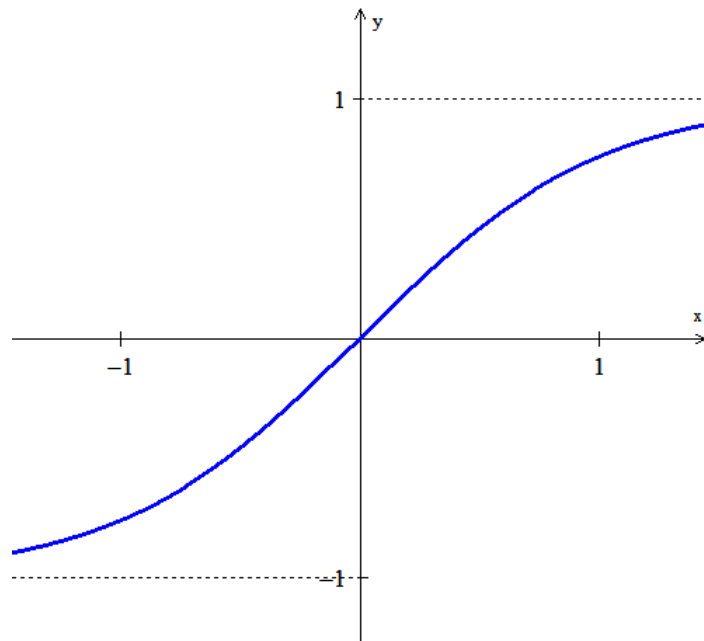


Figura 3.16: Função de ativação logística bipolar
 Fonte: (SILVA et al., 2010)

Silva et al. (2010) produzirá resultados iguais para aqueles valores de potencial de ativação $\{x\}$ que fiquem a uma mesma distância da média que é o centro da distribuição. Essa curva depende do desvio-padrão da distribuição. A Figura 3.18 é uma ilustração gráfica para o caso em que

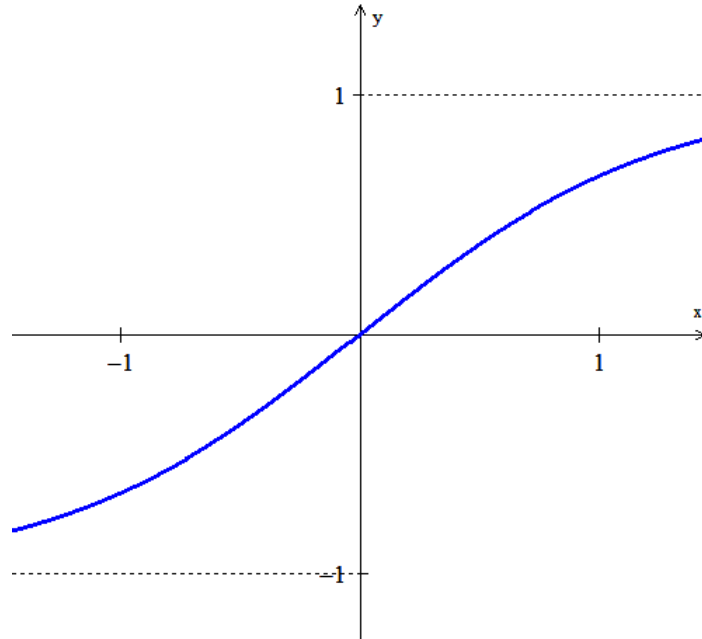


Figura 3.17: Função de ativação tangente hiperbólica bipolar
 Fonte: (SILVA et al., 2010)

$c = 1$. Geometricamente nota-se que a curva é simétrica em relação à reta vertical que passa pela média.

$$f(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (3.7)$$

3.3 Treinamento de redes perceptron multicamadas - algoritmo *backpropagation*

O algoritmo de aprendizagem de uma rede MLP mais conhecido é denominado *backpropagation* que em uma tradução para o português poderia ser chamado de retro-propagação. Esse nome é justificado pelo fato de que o erro proveniente da diferença entre a saída estimada da rede com a saída esperada é usado no ajuste de todos os pesos sinápticos da rede. Cada padrão tem seu sinal propagado da entrada até a saída, e o sinal proveniente do erro é retro-propagado da saída para a entrada.

Nesse algoritmo são definidas algumas variáveis como a taxa de aprendizagem da rede

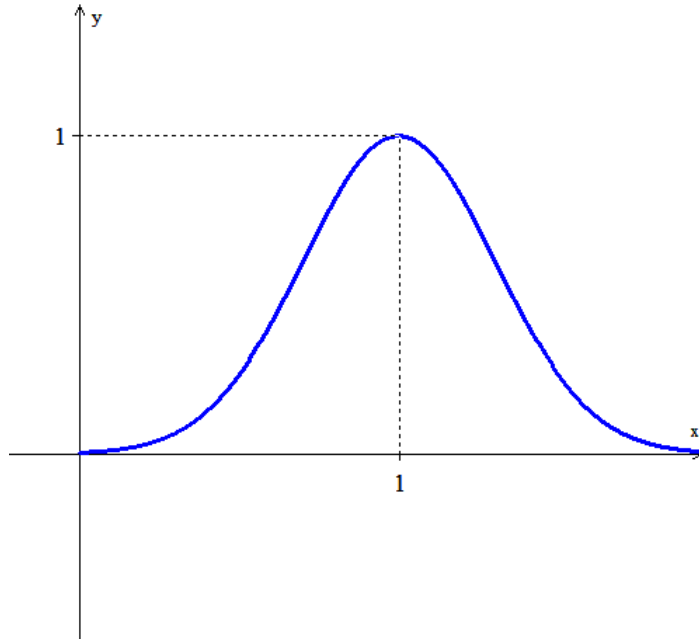


Figura 3.18: Função de ativação gaussiana
 Fonte: (SILVA et al., 2010)

“ α ”, o número de neurônios da camada intermediária e o critério de parada que pode ser feito, por exemplo, com um número limite de iterações ou pela determinação de um erro máximo admissível. A seguir temos os passos para a execução do algoritmo *backpropagation* de acordo com Fausset (1994):

3.3.1 Propagação de um padrão no algoritmo de treinamento

Passo 0: Inicialização dos pesos sinápticos que devem ser valores aleatórios pequenos.

Passo 1: Execução da propagação e retro-propagação até que a condição de parada seja satisfeita.

Passo 2: Para cada padrão de treinamento são executados os passos de 3 a 8

Passo 3: Cada padrão de treinamento X tem seus valores propagados da camada de entrada para a camada intermediária.

Passo 4: Na propagação de um padrão de treinamento, há uma influência dos pesos sinápticos calculando zin_j da seguinte maneira:

$$zin_j^q = \sum_{i=1}^n [x_i^q \cdot v_{ij}^q] + v0_j^q = [zin_1^q \quad zin_2^q \quad \dots \quad zin_j^q] \quad (3.8)$$

Onde:

- x_i^q - São os valores de entrada para o q -ésimo padrão.
- v_{ij}^q - São os pesos sinápticos dos neurônios que ligam a i -ésima entrada ao j -ésimo neurônio da camada intermediária.
- $v0_j^q$ - São os pesos sinápticos dos *bias* da camada intermediária.

Esses valores são submetidos a uma função de ativação para então serem enviados da camada intermediária para a camada de saída.

$$z_j^q = f(zin_j^q) = [f(zin_1^q) \quad f(zin_2^q) \quad \dots \quad f(zin_k^q)] = [z_1^q \quad z_2^q \quad \dots \quad z_k^q] \quad (3.9)$$

Passo 5: Cada unidade de saída é obtida a partir da propagação dos valores de z_j^q da camada intermediária para a camada de saída, onde são aplicados os pesos sinápticos. Matematicamente, esse passo é representado por:

$$yin_k^q = \sum_{j=1}^P [z_j^q \cdot w_{jk}^q] + w0_k^q = [yin_1^q \quad yin_2^q \quad \dots \quad yin_k^q] \quad (3.10)$$

Onde:

- w_{jk}^q - São os pesos sinápticos que ligam o j -ésimo neurônio da camada intermediária à k -ésima saída.
- $w0_k^q$ - São os pesos sinápticos dos *bias* correspondentes as saídas da rede.

Os valores de yin são submetidos à função de ativação gerando os valores y que representam a saída da rede de acordo com a equação 3.11.

$$y_k^q = f(yin_k^q) = [f(yin_1^q) \quad f(yin_2^q) \quad \dots \quad f(yin_k^q)] = [y_1^q \quad y_2^q \quad \dots \quad y_k^q] \quad (3.11)$$

3.3.2 Retropropagação do erro

Passo 6: Cada padrão de entrada tem uma unidade de saída correspondente que é representada pelo vetor-alvo, visto que trata-se de um treinamento supervisionado. O erro proveniente da diferença entre esses valores é usado para a atualização dos pesos. Na equação 3.12 temos o cálculo do erro quadrático e nas equações 3.13, 3.14 e 3.15 esse erro é utilizado nas matrizes de atualização de pesos entre a camada intermediária e a camada de saída.

$$E = \frac{1}{2} \cdot \sum_k \left(t_k^q - y_k \right)^2 \quad (3.12)$$

Onde:

- t_k^q - É a saída esperada para o q -ésimo padrão.

$$\delta_k^q = \left(t_k^q - y_k \right) \cdot f' \left(y_{in_k}^q \right) \quad (3.13)$$

$$\Delta w_{jk}^q = \alpha \cdot \delta_k^q \cdot z_j^q \quad (3.14)$$

$$\Delta w_{0k}^q = \alpha \cdot \delta_k^q \quad (3.15)$$

Onde:

- α - É a taxa de aprendizagem da rede.

Passo 7: Cálculo das matrizes de atualização de pesos entre a camada de entrada e a camada intermediária.

$$\delta_{in_j}^q = \sum_{k=1}^m \left[\delta_k^q \cdot w_{jk}^q \right] \quad (3.16)$$

$$\delta_j^q = \delta_{in_j}^q \cdot f' \left(z_{in_j}^q \right) \quad (3.17)$$

$$\Delta v_{ij}^q = \alpha \cdot \delta_j^q \cdot x_i^q \quad (3.18)$$

$$\Delta v0_j^q = \alpha \cdot \delta_j^q \quad (3.19)$$

Passo 8: Atualização dos pesos e bias.

$$w_{jk}^{q+1} = w_{jk}^q + \Delta w_{jk}^q \quad (3.20)$$

$$v_{ij}^{q+1} = v_{ij}^q + \Delta v_{ij}^q \quad (3.21)$$

$$w0_k^{q+1} = w0_k^q + \Delta w0_k^q \quad (3.22)$$

$$v0_j^{q+1} = v0_j^q + \Delta v0_j^q \quad (3.23)$$

Passo 9: Verificação da condição de parada.

A propagação e retro-propagação de todos os padrões completam um ciclo ou uma época. O algoritmo é executado até que a condição de parada seja satisfeita.

3.4 Fundamentos de vetores-alvo

3.4.1 Vetores-alvo na aprendizagem de uma MLP

A literatura indica a utilização de dois tipos de vetores-alvo no treinamento de redes do tipo MLP: vetores binários e vetores bipolares. Os vetores binários (VBN)s (MANZAN et al., 2011) tem seus elementos iguais a “1” nas posições em que “ $i = j$ ” e iguais a “0” nas posições em que “ $i \neq j$ ”.

$$\vec{V}_{ij} = \begin{cases} 1 & \text{para } i = j \\ 0 & \text{para } i \neq j \end{cases} \quad (3.24)$$

Os vetores binários são ortogonais entre si, mas a distância euclidiana entre eles é sempre igual a $\sqrt{2}$.

No caso dos vetores bipolares (VBC)s definidos pela equação 3.25, que aqui incluiremos o adjetivo “convencionais” para diferenciá-los dos vetores-alvo da nova proposta, tem seus elementos iguais a “1” nas posições em que “ $i = j$ ” e iguais a “-1” nas posições em que “ $i \neq j$ ”.

$$\vec{V}_{ij} = \begin{cases} 1 & \text{para } i = j \\ -1 & \text{para } i \neq j \end{cases} \quad (3.25)$$

Já os vetores bipolares ortogonais (VBO)s, que constituem a nova proposta, têm por propriedade o produto interno nulo, ou seja, para dois vetores \vec{V} e \vec{W} pertencentes ao espaço R^n , o produto interno tem seu resultado igual a zero. Os vetores \vec{V} e \vec{W} dados pelas equações 3.26 e 3.27 são exemplos de vetores bipolares e ortogonais de dimensão 8. Pode-se perceber pela equação 3.28 que eles são ortogonais em razão do produto interno nulo. A obtenção de VBOs pode ser efetuada com o uso de um algoritmo específico mostrado na seção seguinte.

$$\vec{V} \stackrel{def}{=} (1, 1, 1, 1, -1, -1, -1, -1) \quad (3.26)$$

$$\vec{W} \stackrel{def}{=} (1, 1, -1, -1, 1, 1, -1, -1) \quad (3.27)$$

$$\vec{V} \bullet \vec{W} = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot (-1) + 1 \cdot (-1) + (-1) \cdot 1 + (-1) \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1) \quad (3.28)$$

Existem ainda os vetores bipolares não-ortogonais (VNO)s que têm a mesma dimensão dos VBOs, mas possuem produto interno não-nulo. O produto interno dá a medida do grau

de similaridade entre os vetores. Devido ao fato de que estes vetores terão apenas um dígito igual a “1” e os demais iguais a “-1”, o produto interno aumenta à medida que a dimensão dos vetores é ampliada. Nesse sentido, a taxa de reconhecimento da rede diminui à medida que o grau de similaridade dos vetores-alvo aumenta.

3.4.2 Algoritmo de geração de vetores bipolares ortogonais

O método de geração de vetores bipolares ortogonais é proposto por (FAUSETT, 1994). Nele é utilizado um vetor de sementes na geração de VBOs. Chamamos de vetor de sementes pelo fato de que através deles será gerada uma 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. A fórmula dada pela equação 3.29 leva a esse número de componentes de cada vetor.

$$n = 2^k m \quad (3.29)$$

Na equação 3.29, 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 mesmos, seleciona somente a quantidade de vetores que necessitar.

Assim sendo, um conjunto com 2^k VBOs é construído com $2^k m$ componentes. O algoritmo é executado a partir dos seguintes passos:

Passo 1 - Iniciação de m e k - Os valores de m e k devem ser determinados de acordo com a necessidade da aplicação. O valor de m pode ser “1”, e a medida que o valor de m é aumentado, a quantidade de componentes dos VBOs cresce progressivamente.

Passo 2 - Iniciação do vetor de sementes - O vetor de sementes é obtido através de 3.30. Vale ressaltar que o vetor de sementes pode ter um único componente.

$$V_m^0 = \overbrace{(1, 1, \dots, 1)}^m \quad (3.30)$$

Passo 3 - Cálculo do número de componentes em um VBO - O cálculo do número de componentes do VBO é feito através da equação 3.29.

Passo 4 - Construção de vetores - Usa-se a função de concatenação para a construção dos vetores.

Sejam os vetores $U = (u_1, u_2, \dots, u_n)$ e $W = (w_1, w_2, \dots, w_n)$. A função de concatenação $fcc(U, W)$ é definida pela equação:

$$fcc(U, W) = (u_1, u_2, \dots, u_n, w_1, w_2, \dots, w_n) \quad (3.31)$$

A partir do vetor de sementes obtido no passo 2, são construídos os vetores $V_{2m}^1 = fcc(V_m^0, V_m^0)$ e $V_{2m}^2 = fcc(V_m^0, -V_m^0)$, onde $fcc(V_1, V_2)$. Pode se perceber que o vetor de sementes é utilizado duas vezes, sendo que na primeira há a concatenação com o segundo argumento com sinal mantido e na outra o sinal do segundo argumento é trocado.

Passo 5 - Construção de vetores - A partir dos dois vetores obtidos no passo 4, serão determinados quatro novos vetores com a utilização da função de concatenação. Os vetores construídos são $V_{4m}^1 = fcc(V_{2m}^1, V_{2m}^1)$, $V_{4m}^2 = fcc(V_{2m}^1, -V_{2m}^1)$, $V_{4m}^3 = fcc(V_{2m}^2, V_{2m}^2)$ e $V_{4m}^4 = fcc(V_{2m}^2, -V_{2m}^2)$. Os vetores determinados pelo passo anterior são usados duas vezes na função de concatenação, sendo que na primeira vez o sinal do segundo argumento é mantido e na segunda é trocado.

Passo 6 - Sequência da Concatenação - São realizadas concatenações até que se consiga 2^k vetores ortogonais com n componentes: $V_n^1, \dots, V_n^{2^k}$.

Consideremos o exemplo em que se deseja construir 8 vetores ortogonais bipolares. Suponha que seja decidido usar um vetor de sementes com um componente. Assim, teremos de acordo com a equação 3.32, 8 componentes em cada vetor.

$$n = 2^k m = 8 \cdot 1 = 8 \quad (3.32)$$

Nosso vetor de sementes é dado então por $V_m^0 = (1)$. Passando ao passo 4, obtemos os vetores dados pelas equações 3.33 e 3.34.

$$V_{2m}^1 = fcc(V_m^0, V_m^0) = (1, 1) \quad (3.33)$$

$$V_{2m}^1 = fcc(V_m^0, -V_m^0) = (1, -1) \quad (3.34)$$

No passo 5, obtemos um novo conjunto de vetores dados pelas equações 3.35, 3.36, 3.37 e 3.38.

$$V_{4m}^1 = fcc(V_{2m}^1, V_{2m}^1) = (1, 1, 1, 1) \quad (3.35)$$

$$V_{4m}^2 = fcc(V_{2m}^1, -V_{2m}^1) = (1, 1, -1, -1) \quad (3.36)$$

$$V_{4m}^3 = fcc(V_{2m}^2, V_{2m}^2) = (1, -1, 1, -1) \quad (3.37)$$

$$V_{4m}^4 = fcc(V_{2m}^2, -V_{2m}^2) = (1, -1, -1, 1) \quad (3.38)$$

Seguindo o algoritmo num possível passo 6, obtemos os vetores dados pelas equações 3.39, 3.40, 3.41, 3.42, 3.43, 3.44, 3.45 e 3.46.

$$V_{8m}^1 = fcc(V_{4m}^1, V_{4m}^1) = (1, 1, 1, 1, 1, 1, 1, 1) \quad (3.39)$$

$$V_{8m}^2 = fcc(V_{4m}^1, -V_{4m}^1) = (1, 1, 1, 1, -1, -1, -1, -1) \quad (3.40)$$

$$V_{8m}^3 = fcc(V_{4m}^2, V_{4m}^2) = (1, 1, -1, -1, 1, 1, -1, -1) \quad (3.41)$$

$$V_{8m}^4 = fcc(V_{4m}^2, -V_{4m}^2) = (1, 1, -1, -1, -1, -1, 1, 1) \quad (3.42)$$

$$V_{8m}^5 = fcc(V_{4m}^3, V_{4m}^3) = (1, -1, 1, -1, 1, -1, 1, -1) \quad (3.43)$$

$$V_{8m}^6 = fcc(V_{4m}^3, -V_{4m}^3) = (1, -1, 1, -1, -1, 1, -1, 1) \quad (3.44)$$

$$V_{8m}^7 = fcc(V_{4m}^4, V_{4m}^4) = (1, -1, -1, 1, 1, -1, -1, 1) \quad (3.45)$$

$$V_{8m}^8 = fcc(V_{4m}^4, -V_{4m}^4) = (1, -1, -1, 1, -1, 1, 1, -1) \quad (3.46)$$

Se realizarmos o produto interno tomando cada par de vetores, verificaremos que serão iguais a zero. Dessa forma, conseguimos gerar os oito VBOs e cada um possui oito componentes.

Uma característica importante dos VBOs é que a distância euclidiana entre eles aumenta a medida que suas dimensões são ampliadas. Pode-se mostrar isso determinando essa distância para um conjunto hipotético de VBOs. Considere a fórmula da distância mostrada na equação 3.49 a partir de dois vetores \vec{U} e \vec{W} mostrados nas equações 3.47 e 3.48.

$$\vec{U} = (u_1, u_2, u_3, \dots, u_n) \quad (3.47)$$

$$\vec{W} = (w_1, w_2, w_3, \dots, w_n) \quad (3.48)$$

$$d_{U,W} = \sqrt{(u_1 - w_1)^2 + (u_2 - w_2)^2 + (u_3 - w_3)^2 + \dots + (u_n - w_n)^2} \quad (3.49)$$

Pela característica dos VBOs, tem-se que para vetores de dimensão n , metade das n

diferenças da fórmula da distância se anulará e a outra metade será igual a $2^2 = 4$. Por essa razão, a fórmula da equação 3.49 pode ser reescrita na equação 3.50.

$$d_{U,W} = \sqrt{\frac{n}{2} \cdot 4} = \sqrt{2n} \quad (3.50)$$

O gráfico da Figura 3.19 mostra a evolução da distância euclidiana de acordo com o aumento da dimensão entre os vetores-alvo. Com VBC e VNO a distância permanece inalterada valendo sempre $\sqrt{2}$. Para VBOs essa distância é sempre crescente.

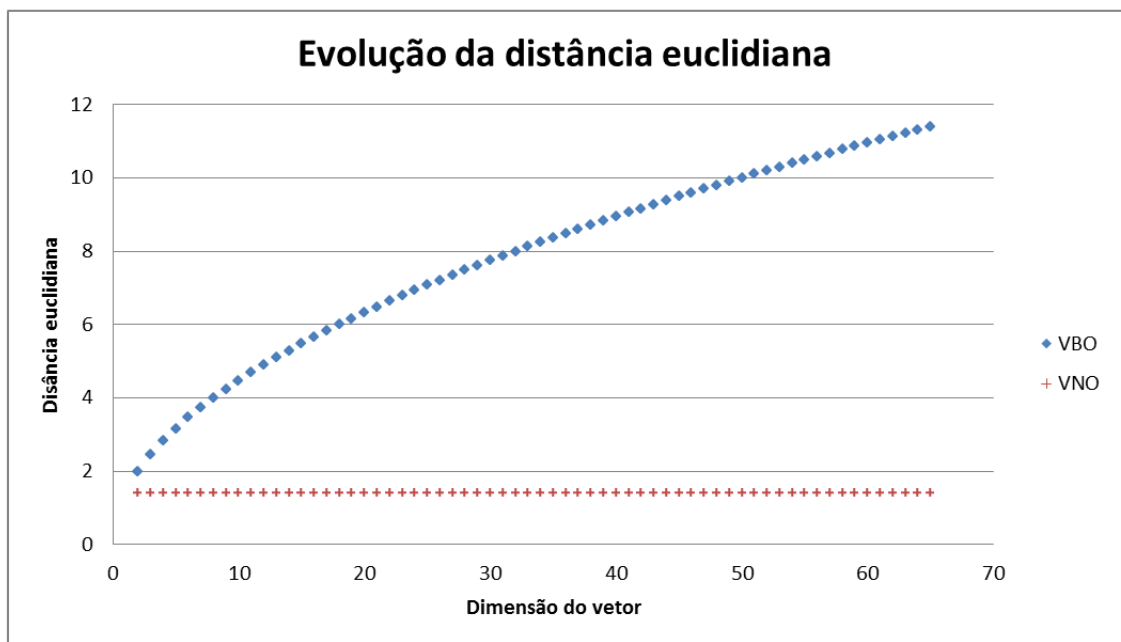


Figura 3.19: Distância euclidiana de vetores-alvo

Capítulo 4

Método Proposto

4.1 Hipótese de comparação de vetores-alvo - Intersecção entre regiões de convergência

À medida que os vetores-alvo convencionais (VBC)s ou os vetores-alvo não ortogonais (VNO)s aumentam de dimensão, seu produto interno também aumenta. Quanto maior o valor do produto interno, menor será o ângulo formado entre esses vetores, o que indica que quão maiores forem esses vetores, maior será a similaridade deles, ou seja, ficarão mais próximos de um paralelismo. Por outro lado, vetores ortogonais sempre têm produto interno nulo, fazendo com que os mesmos tenham um ângulo de 90° entre si. Essa propriedade matemática possibilita que os vetores tenham a maior distância possível entre si. A maior distância implica em uma menor similaridade. Logicamente essa menor similaridade facilita a capacidade da rede MLP em reconhecer padrões.

Observe a interferência que os vetores-alvo comuns acarretam entre suas áreas de convergência. A Figura 4.1 ilustra o treinamento de uma RNA com o uso de vetores comuns. A Figura 4.2 mostra o treinamento de uma RNA usando VBOs. É perceptível que a interferência é muito menor se comparada ao uso dos vetores convencionais. É nessa lógica que o uso de VBOs possibilita uma aprendizagem que permite que a Rede possua maior capacidade de

generalização.

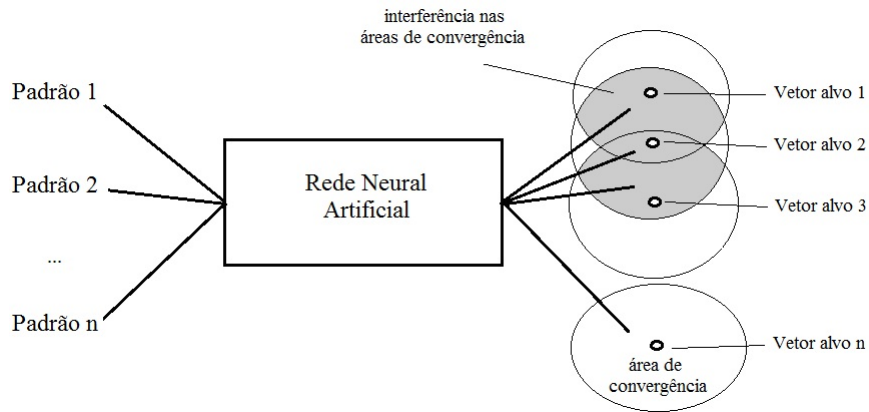


Figura 4.1: Treinamento de rede MLP com o uso de vetores-alvo comuns

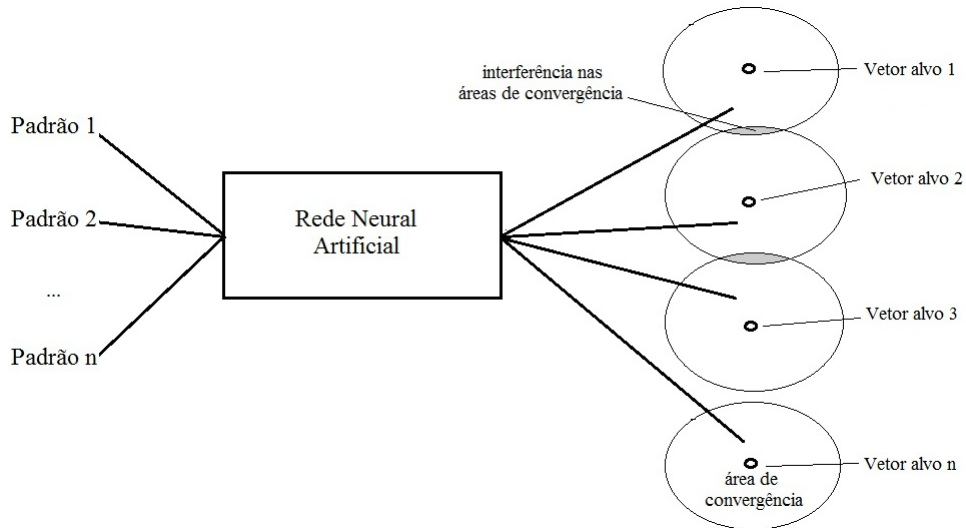


Figura 4.2: Treinamento de rede MLP com o uso de vetores-alvo bipolares e ortogonais

Outro ponto interessante de ser analisado é a questão da distância euclidiana. As equações 4.1 e 4.2 representam dois possíveis vetores-alvo, a equação 4.3 ilustra o produto interno entre eles e a equação 4.4, sua distância euclidiana.

$$\vec{V}_i = (v_1, v_2, \dots, v_n) \quad (4.1)$$

$$\vec{W}_i = (w_1, w_2, \dots, w_n) \quad (4.2)$$

$$\vec{V}_i \bullet \vec{W}_i^T = v_1 \cdot w_1 + v_2 \cdot w_2 + v_3 \cdot w_3 + \dots + v_n \cdot w_n \quad (4.3)$$

$$d_{v,w} = \sqrt{(w_1 - v_1)^2 + (w_2 - v_2)^2 + (w_3 - v_3)^2 + \dots + (w_n - v_n)^2} \quad (4.4)$$

Pode-se ver o caso específico em que \vec{v} e \vec{w} são ortogonais bipolares com tamanho n . Seu produto interno terá $\frac{n}{2}$ parcelas cujo produto é positivo e $\frac{n}{2}$ parcelas cujo produto é negativo. Parcelas cujo produto é positivo correspondem àquelas em que os termos têm mesmo sinal. Estes termos têm diferença nula, não interferindo na soma compreendida pela raiz quadrada da fórmula da distância euclidiana representada pela equação 4.4. Termos que têm sinais diferentes terão diferença cujo módulo é igual a “2”. O quadrado dessas diferenças contribui para a soma compreendida pela raiz quadrada da fórmula da distância euclidiana. Logo, quanto maior for o valor de n , maior será a distância euclidiana entre os vetores.

No caso dos VNOs ou de VBCs, um dos termos na posição “ i ” é igual a “1” e os demais são iguais a “-1”. Dessa forma, se \vec{v} e \vec{w} forem convencionais, as diferenças entre os termos correspondentes serão sempre iguais a 0, exceto nos dois termos, um de cada vetor, que são iguais a “1”. Conclui-se que a distância euclidiana de vetores convencionais é sempre menor que a de VBOs.

Deve-se analisar também os vetores binários (VBN)s que têm um elemento igual a “1” na posição “ i ” e os demais elementos iguais a “0”. A distância entre VBNs quaisquer é sempre igual a $\sqrt{2}$ independentemente da sua dimensão.

Esse aspecto dos VBOs proporciona a obtenção de um número maior de neurônios ativos. Além disso, essa distância ampliada entre os vetores-alvo pode ser comparada à diferença entre padrões, o que nos mostra que a MLP melhora seu desempenho graças à similaridade nula entre esses vetores. Isso não ocorre com os vetores convencionais.

4.1.1 Posições relativas entre circunferências

Para tornar clara a discussão aqui apresentada, é importante fazer menção às posições relativas entre circunferências. Esses conceitos são importantes para se obter uma melhor visualização das regiões de convergência. Tomamos o conceito de circunferência e de raios de convergência no espaço bidimensional para fins ilustrativos. Entretanto, nos problemas práticos lidamos com regiões do espaço n -dimensional cuja visualização é complexa. Duas circunferências podem ser concêntricas, secantes, tangentes internas, tangentes externas ou simplesmente externas como ilustram as Figuras 4.3, 4.4, 4.5, 4.6 e 4.7.

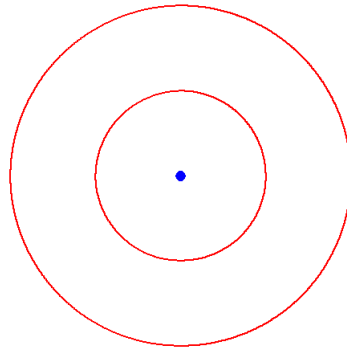


Figura 4.3: Circunferências concêntricas

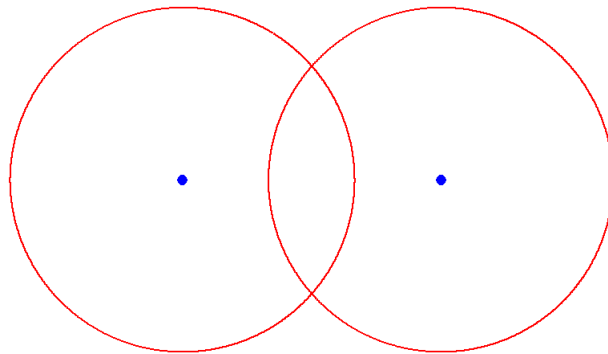


Figura 4.4: Circunferências secantes

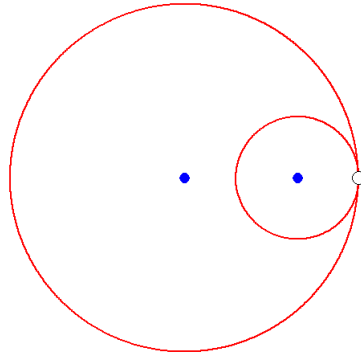


Figura 4.5: Circunferências tangentes internas

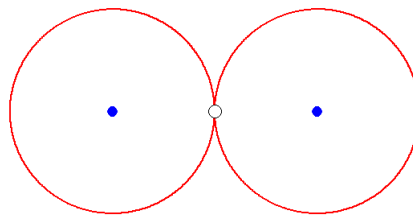


Figura 4.6: Circunferências tangentes externas

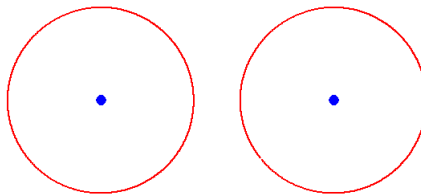


Figura 4.7: Circunferências externas

4.1.2 Procedimento experimental sobre regiões de convergência

A fim de comprovar a hipótese apresentada nas Figuras 4.1 e 4.2 foram realizados experimentos para a classificação de dígitos manuscritos com redes neurais do tipo MLP. Os dígitos manuscritos (*Semeion Handwritten Digit*) foram obtidos do repositório internacional conhecido como *UCI Machine Learning Repository* (REPOSITORY, 1994). Tratam-se de padrões obtidos de um grupo de cerca de 80 pessoas que foram solicitadas a escrever duas vezes os algarismos de 0 a 9. Na primeira solicitação, as pessoas escreveram os dígitos de forma tranquila, primando pela perfeição dos manuscritos. Na segunda solicitação, as pessoas foram orientadas a escrever os algarismos de forma rápida, sem preocupação com a legibilidade dos mesmos.

Cada algarismo foi digitalizado em uma imagem contendo 256 pixels, no formato de 16 linhas por 16 colunas. Cada imagem foi processada numa escala de resolução de 256 níveis de cinza para então ser binarizada. Para cada pixel correspondente ao fundo da imagem, foi atribuído o valor “0” e para os pixels do algarismo, foi atribuído o valor “1”. No treinamento desse trabalho, os pixels com valor “0” passaram a receber o valor “-1”, para que fosse possível realizar o treinamento com vetores bipolares.

Esses experimentos tiveram por objetivo a avaliação das distâncias euclidianas entre os vetores-alvo e as saídas obtidas durante o treinamento da rede. A comparação ocorreu entre vetores-alvo bipolares convencionais e vetores-alvo bipolares e ortogonais. No caso dos vetores-alvo bipolares convencionais foram avaliadas as dimensões 10, 16 e 32. Para os vetores-alvo bipolares e ortogonais foram utilizadas as dimensões 16 e 32. No total, foram comparados 5 tipos de vetores-alvo.

Dessa maneira, cinco tipos de redes neurais foram utilizadas nos experimentos, correspondendo cada um a um tipo de vetor-alvo. A camada de entrada é formada por 256 neurônios. Todas as redes foram treinadas com 50 neurônios na camada intermediária. A quantidade de neurônios da camada de saída depende do tipo de vetor-alvo utilizado. Os pesos sinápticos iniciais foram mantidos fixos a fim de garantir a igualdade de condições nos treinamentos. A taxa de aprendizagem foi fixada em 0,2.

Ao todo foram utilizados 10 conjuntos de dados, ou 10 amostras. Para cada rede neural, foram realizadas 10 simulações, ou seja, a mesma rede neural foi treinada 10 vezes. Em cada uma dessas simulações foi utilizada uma amostra de treinamento diferente. Cada tipo de rede foi treinada com cada uma das amostras disponíveis para treinamento.

Em cada rede, foram calculadas as distâncias euclidianas entre a saída obtida pela rede e os vetores-alvo em tempo de treinamento. Cada treinamento foi programado para ser encerrado com 10000 ciclos.

Após a execução de todas as simulações, foi selecionada a maior de todas as distâncias euclidianas correspondente a cada padrão. Se cada rede treina 10 dígitos, cada treinamento conta com 10 valores de máximas distâncias entre as saídas e seus respectivos alvos. Como são 10 redes para cada tipo de vetor-alvo, temos um total de 100 máximos valores de distância euclidiana citados anteriormente.

Essas distâncias máximas são associadas aos raios de convergência do treinamento de cada padrão em torno da saída esperada, ou melhor dizendo, em torno de seu vetor-alvo correspondente. Deve ser lembrado que o conceito de raio não se restringe a circunferências que são próprias do espaço bidimensional e esferas que é um dos tipos de superfícies do espaço tridimensional. Toda e qualquer região do espaço R^n que seja composta por pontos equidistantes a um centro traz consigo o conceito de raio. Nos experimentos abordados, estamos trabalhando em espaços com dimensões de tamanhos 10, 16 e 32, e iremos visualizar essas distâncias como raios de convergência.

Essa avaliação é importante no sentido de estudar a nuvem de pontos do espaço n -dimensional que é produzida em torno de cada vetor-alvo. De maneira mais ampla, as respostas para a hipótese levantada nas seções 4.1 e 4.2 estão na nuvem de pontos que é formada em torno de cada vetor-alvo usado no treinamento da rede MLP. Cada vetor terá sua respectiva nuvem. A intersecção entre essas nuvens vem de encontro à discussão levantada e nos mostram de que maneira os VBOs melhoram a capacidade de generalização de redes do tipo MLP.

Relacionando essa discussão com as posições relativas de circunferência mostradas na subseção 4.1.1 é possível chegar às seguintes conclusões. Não é possível que as regiões de

convergência estejam dispostas como circunferências concêntricas ou tangentes internamente conforme mostrado nas Figuras 4.3 e 4.5. Isso é explicado pelo fato de que entendendo cada vetor-alvo como o centro da região de convergência, eles nunca estarão na mesma posição. A associação das distâncias máximas com os raios de convergência também explica a impossibilidade de haver regiões dispostas como citado anteriormente.

Assim, nessa associação de posições relativas de circunferências com regiões de convergência, nos restam regiões de circunferências secantes, tangentes externamente ou externas conforme mostrado nas Figuras 4.4, 4.6 e 4.7. O ideal seria a disposição de regiões de convergência externas ou tangentes externamente dando assim a condição de interseção vazia ou de interseção com ponto único. Contudo, os resultados experimentais que serão mostrados a seguir, indicam a ocorrência apenas de regiões secantes. Assim, sempre existe interseção entre as regiões de convergência. Dessa maneira verificam-se que, regiões de convergência com menor grau de interseção possibilitam uma melhor aprendizagem para a rede neural.

4.1.3 Resultados experimentais sobre áreas de convergência

Os resultados estatísticos são apresentados na Tabela 4.1 e no gráfico da Figura 4.8. A partir das simulações foram determinados os máximos das distâncias entre o vetor-alvo e as saídas obtidas pela rede em cada uma das 10 simulações. Considerando que cada rede foi treinada para reconhecer 10 padrões (dígitos manuscritos de 0 a 9), então, cada simulação gerou 10 máximos. Dessa maneira, cada tipo de rede neural, treinada com um tipo de vetor-alvo, tem 100 valores para máximas distâncias. A seguir, apontamos as medidas estatísticas que são apresentadas na Tabela 4.1 e os dados sobre os vetores-alvo.

1. *Máximos globais* - para se obter o valor de um máximo global, de um conjunto de 100 máximos, escolheu-se o maior de todos;
2. *Média estimada dos máximos* - é a média aritmética dos 100 máximos obtidos nas 10 simulações;
3. *Intervalo de confiança para a média* - considerando um nível de significância de

- 5% foi determinado um intervalo de confiança para a média dos máximos;
4. *Desvio-padrão* - foi calculado o desvio-padrão dos 100 máximos obtidos no experimento de maneira a determinar o grau de variabilidade das distâncias euclidianas;
 5. *Coefficiente de variação* - a partir do desvio-padrão e da média dos máximos, foi determinada a variação relativa de cada tipo de vetor-alvo;
 6. *Distância entre vetores-alvo (dva)* - trata-se da distância euclidiana entre vetores-alvo que é única para cada tipo de vetor-alvo, uma vez que eles são equidistantes;
 7. *Metade da dva - $\frac{1}{2}dva$* - metade da distância entre vetores-alvo que é o limite do raio ideal de regiões de convergência, visto que nessa condição as regiões estariam dispostas como em circunferências tangentes externamente, ou seja, o raio é a metade da distância entre os centros das regiões;
 8. *Diferença entre máximos e $\frac{1}{2}dva$* - é a diferença entre os máximos globais, que representam os raios das regiões de convergência obtidos nos experimentos e o raio de regiões de convergência ideais que são a metade da distância euclidiana entre vetores-alvo;
 9. *Percentual da diferença em relação a $\frac{1}{2}dva$* - é o percentual da diferença citada no item 8 em relação ao raio ideal para regiões de convergência citado na linha 7.

Tabela 4.1: Análise estatística da comparação das áreas de convergência de vetores-alvo

Parâmetros estatísticos	VBC(10)	VNO(16)	VNO(32)	VBO(16)	VBO(32)
1 - Máximos globais	5,00	6,58	9,75	6,26	8,57
2 - Média estimada dos máximos*	2,65	3,03	3,70	5,18	7,33
3 - Intervalo de confiança da média*	2,52 a 2,77	2,82 a 3,24	3,32 a 4,09	5,09 a 5,27	7,23 a 7,44
4 - Desvio-padrão estimado*	0,63	1,06	1,95	0,47	0,53
5 - Coeficiente de variação*	23,84%	34,93%	52,85%	9,16%	7,23%
6 - Distância entre vetores-alvo (dva)	1,41	1,41	1,41	5,66	8,00
7 - Metade da dva - $\frac{1}{2}dva$	0,71	0,71	0,71	2,83	4,00
8 - Diferença entre máximos e $\frac{1}{2}dva$	4,29	5,87	9,04	3,43	4,57
9 - Percentual da diferença em relação a $\frac{1}{2}dva$	607,11%	830,55%	1278,86%	121,32%	114,25%

* As estimações foram feitas ao nível de 5% de significância

Pode-se verificar que os menores coeficientes de variação são obtidos com a utilização de VBOs. Isso mostra que os raios de convergência de seus respectivos vetores-alvo são mais homogêneos ao passo que os vetores-alvo bipolares e não ortogonais possuem raios de convergência bastante heterogêneos, principalmente quando suas dimensões são ampliadas. Os

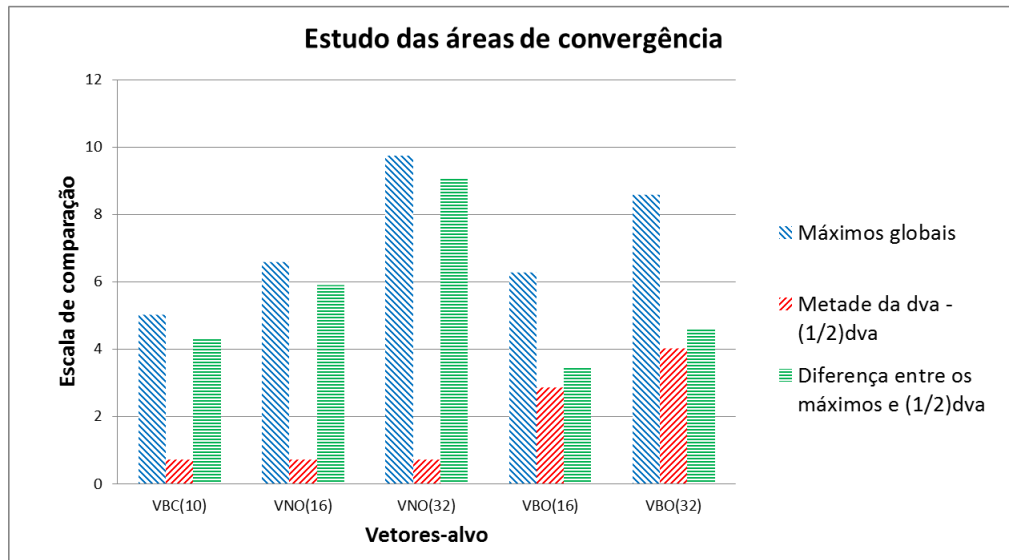


Figura 4.8: Comparação das invasões entre áreas de convergência

coeficientes de variação são menores do que 10% para VBOs e são superiores a 20% para os VNOs de maneira que para a dimensão 32 seu valor chega a 52,85%.

Contudo a melhor forma de avaliar a intersecção entre as áreas de convergência é a comparação do quanto excede o raio da área de convergência em relação à metade da distância entre vetores-alvo. Metade da distância entre os vetores-alvo seria o limite do raio ideal para as áreas de convergência, visto que nessa situação teríamos circunferências tangentes externamente. A diferença entre o raio de convergência obtido, ou seja, o máximo global e a metade da distância entre os vetores-alvo representa o quanto o raio de convergência avança além do limite do ideal. Com os VNOs o avanço vai de 607,11% para (10) a 1278,86% para VNO (32). No caso dos VBOs esse avanço é da ordem de 121,32% para VBO (16) e 114,25% para VBO (32).

O gráfico da Figura 4.8 também nos mostra essa diferença. Verifica-se que as diferenças entre os máximos globais e $\frac{1}{2}dva$ excede a metade das distâncias entre vetores-alvo com pouca intensidade quando os mesmos são VBOs. No caso de vetores-alvo do tipo VBCs ou VNOs essas mesmas diferenças excedem de maneira bastante intensa a metade das distâncias entre vetores-alvo.

Tomemos uma maneira adicional de avaliação das regiões de convergência. Consideremos uma representação bidimensional de circunferências com raios equivalentes aos raios de convergência obtidos nos experimentos. As Figuras 4.9, 4.10, 4.11, 4.12 e 4.13 representam as circunferências ilustrativas das regiões de convergência que têm seus centros distantes entre si de acordo com as distâncias entre os vetores-alvo. Tais distâncias são ilustradas em cada uma das figuras. Todas as figuras estão representadas na mesma escala o que permite uma comparação justa. Essas figuras mostram a invasão que cada região de convergência tem nas regiões de convergência de outro vetor-alvo.

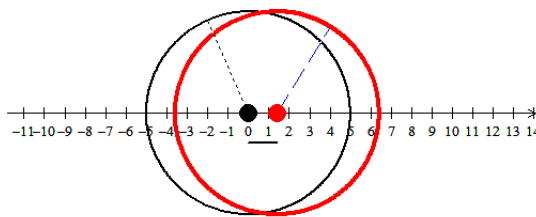


Figura 4.9: Regiões de convergência para VBC (10)

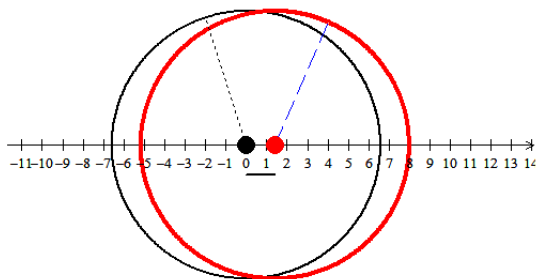


Figura 4.10: Regiões de convergência para VNO (16)

A Figura 4.9 mostra a região de convergência da VBCs e as Figuras 4.10 e 4.11 mostram as regiões de convergência para VNOs. Pode-se verificar que as regiões de convergência avançam de forma bastante significativa umas nas outras. O que vemos é praticamente uma sobreposição de uma região sobre a outra. Essa característica dos vetores-alvo do tipo VBC e VNO vem de encontro com a hipótese formulada na Figura 4.1 onde o alto grau de similaridade dos pontos do espaço de saída dificulta a aprendizagem da RNA.

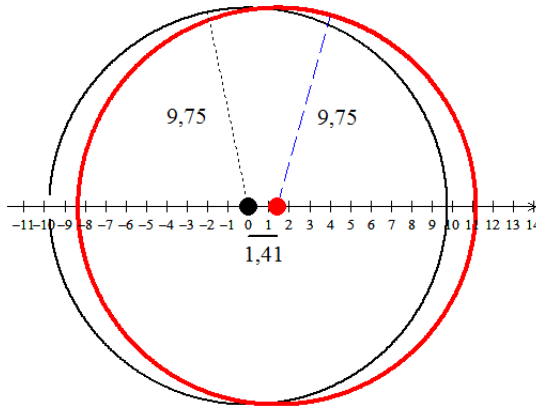


Figura 4.11: Regiões de convergência para VNO (32)

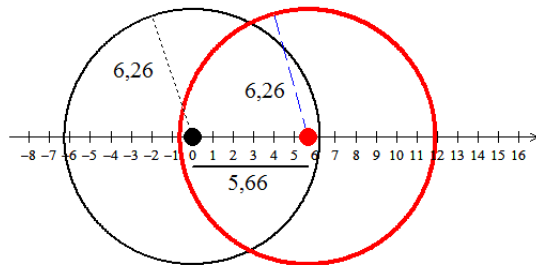


Figura 4.12: Regiões de convergência para VBO (16)

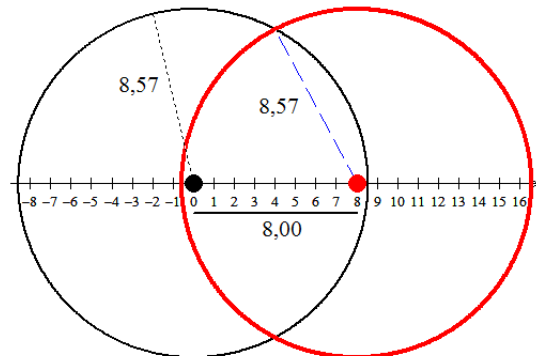


Figura 4.13: Regiões de convergência para VBO (32)

No caso das Figuras 4.12 e 4.13 existe o avanço de uma região sobre outra, porém com intensidade bem menor. Nesse sentido, os resultados experimentais estão de acordo com a hipótese formulada. Pela ortogonalidade entre vetores-alvo do tipo VBO a similaridade entre eles é nula. Logo, a interseção entre as regiões de convergência deveria ser bem menor como

pode ser concluída pela análise dos resultados experimentais.

Capítulo 5

Análise matemática

5.1 Melhoria do desempenho de MLP com o uso de VBOs como vetores-alvo da rede

Resultados experimentais demonstrando a melhora do desempenho de redes do tipo MLP com o uso de VBOs como vetores-alvo, já foram mostrados nos trabalhos de Nomura et al. (2005), Nomura et al. (2004), Manzan et al. (2011) e Nomura et al. (2010). Entretanto, as razões para esse desempenho merecem um estudo mais cuidadoso e detalhado a fim de esclarecer quais elementos garantem essa melhora na capacidade de reconhecer padrões degradados.

No intuito de responder essa questão, esse estudo teve por estratégia, a análise genérica da propagação de dois padrões consecutivos juntamente com as retro-propagações de seus erros. Após a apresentação de um padrão para a rede, sua saída é comparada com o vetor-alvo correspondente. A diferença dessa comparação é utilizada na atualização dos pesos sinápticos. Logo faz-se a comparação das interferências de vetores-alvo consecutivos, que podem ser ortogonais ou não, na atualização dos pesos.

Após a propagação do primeiro padrão os pesos sinápticos iniciais são atualizados e nessa atualização, ocorre uma comparação entre a saída encontrada e a saída desejada, ou seja, uma comparação com o vetor-alvo. Quando o segundo padrão é propagado pela rede ainda

dentro do mesmo ciclo, os pesos sinápticos são atualizados na propagação do primeiro padrão, que contém a diferença envolvendo seu vetor-alvo. Nessa nova atualização dos pesos, temos presentes a comparação do segundo vetor de saída com o segundo vetor-alvo. Mas esse sinal propagado possui valores oriundos da diferença com o alvo do primeiro padrão.

A partir dessa evidência, é que se realiza o estudo dessa interação entre dois vetores-alvo de padrões diferentes e consecutivos. Em que essa interação interfere na aprendizagem da rede? Quais são as diferenças quando são usados VBOs ou vetores convencionais como vetores-alvo da rede?

Inicialmente é feita uma análise sobre o que acontece na atualização dos pesos sinápticos dos neurônios que ligam a camada intermediária à camada de saída. Em seguida é analisado o que ocorre na atualização dos pesos sinápticos dos neurônios que ligam a camada de entrada à camada intermediária.

Nas expressões matemáticas mostradas neste trabalho, iremos considerar que o símbolo “ \cdot ” representa a multiplicação usual entre matrizes ou o produto entre dois números reais ou ainda a multiplicação de um escalar por uma matriz. O símbolo “ \bullet ” será atribuído para o produto interno entre vetores do espaço R^n . O símbolo “ $*$ ” é usado para uma multiplicação não usual entre duas matrizes com mesma dimensão. Nessa operação, cada elemento da primeira matriz, correspondente a i -ésima linha e j -ésima coluna, é multiplicado pelo elemento da segunda matriz, correspondente a i -ésima linha e j -ésima coluna. O resultado da operação é uma matriz com a mesma dimensão das matrizes envolvidas pela multiplicação não usual.

Esse estudo foi submetido em forma de artigo ao “ICAI’12 - *The 2012 International Conference on Artificial Intelligence*” e foi aceito para publicação nos anais do evento (Manzan et al., 2012).

5.2 Atualização dos pesos entre a camada intermediária e a camada de saída

Consideremos que um padrão de ordem q seja propagado pela rede, e consideremos a retro propagação do erro, mais especificamente o cálculo de δ_k^q na equação 5.1.

$$\delta_k^q = (t_k^q - y_k) \cdot f'(y_{in_k}^q) \quad (5.1)$$

Onde:

t_k^q - Representa o vetor-alvo correspondente ao q -ésimo padrão que é propagado pela rede.

y_k - Representa a saída da rede para o q -ésimo padrão que é propagado pela rede.

$f'(y_{in_k}^q)$ - Denota a derivada da função de ativação da saída líquida da rede $y_{in_k}^q$ no q -ésimo padrão.

Representemos a expressão 5.1 de forma vetorial na expressão 5.2.

$$\delta_k^q = \left(\begin{array}{c} \left[\begin{array}{c} t_1 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ t_k \end{array} \right] \\ - \left[\begin{array}{cccc} y_1 & y_2 & \dots & y_k \end{array} \right]^T \end{array} \right) * \left[\begin{array}{cccc} y'_1 & y'_2 & \dots & y'_k \end{array} \right]^T = \left[\begin{array}{c} \delta_1 \\ \delta_2 \\ \cdot \\ \cdot \\ \delta_k \end{array} \right] \quad (5.2)$$

Onde:

$(t_i - y_i)$ é a diferença entre o i -ésimo valor do vetor-alvo t_i e o i -ésimo valor da saída y_i .

Na equação 5.2, podemos identificar algumas diferenças entre o uso de VBCs, VBNs, VNOs e VBOs. Se forem utilizados VBNs como alvo da rede, os valores de y_i estarão sempre compreendidos entre “0” e “1”. Nesse sentido quando o valor do k -ésimo elemento do vetor-

alvo for “0”, a diferença $(t_i - y_i)$ estará compreendida entre “-1” e “0”. Por outro lado, quando o valor do k -ésimo elemento do vetor-alvo for “1”, a diferença $(t_i - y_i)$ estará compreendida entre “0” e “1”. Essas diferenças são sempre multiplicadas pela derivada da função ativação em y_i . Sendo a função de ativação sempre crescente, essas derivadas serão sempre positivas. Como nesse tipo de vetor-alvo, todos os elementos são nulos pela exceção de um elemento na posição “ i ” que será igual a “1”, os valores de δ_k serão todos negativos estando entre “-1” e “0” exceto pelo elemento da posição “ i ”.

No caso dos VNOs e VBCs os valores de y_i estão sempre compreendidos entre “-1” e “1”. Se o valor de um elemento t_i do vetor de saída for “1”, a diferença $(t_i - y_i)$ estará compreendida entre “0” e “2”. Por outro lado quando o valor do elemento de t_i do vetor de saída for -1, a diferença estará compreendida entre “-2” e “0”. Essas diferenças são sempre multiplicadas pela derivada da função ativação em y_i . Sendo a função de ativação sempre crescente, essas derivadas serão sempre positivas. VNOs e VBCs têm todos os elementos iguais a “-1” exceto por um elemento na posição “ i ” que é igual a “1”. Logo os valores de δ_k serão sempre negativos exceto na posição “ i ”.

No caso de VBOs, podemos ter um vetor com todos os valores iguais a “1” e os outros vetores com metade dos elementos iguais a “1” e a outra metade dos elementos iguais a “-1”. Assim sendo, para o primeiro vetor, todos os valores de δ_k serão positivos. Para os demais vetores ortogonais, metade dos valores de δ_k será positiva e a outra metade será negativa.

O vetor δ_k é usado no cálculo de Δw_{jk} 5.3 e Δw_{0k} 5.4.

$$\Delta w_{jk}^q = \alpha \cdot \delta_k^q \cdot z_j^q \quad (5.3)$$

$$\Delta w_{0k}^q = \alpha \cdot \delta_k^q \quad (5.4)$$

No caso de Δw_{jk} há uma multiplicação pela matriz z_j^q que tem valores compreendidos entre “-1” e “1”. Quando são utilizados VBCs, VNOs ou VBNs, cada linha da matriz de Δw_{jk} terá todos os seus elementos com sinal oposto ao j -ésimo valor de z_j^q exceto em uma linha na

posição “ i ” correspondente ao valor positivo de δ_k . Constatamos essa informação com o caso da propagação do 1º padrão de treinamento em uma rede que usa vetores convencionais de dimensão “ k ” como vetores-alvo. Os valores de $\overset{q}{z}_j$ podem ter sinais arbitrários. Para exemplificar consideremos que $\overset{q}{z}_j$ tenha os sinais de seus valores de acordo com a equação 5.5.

$$\overset{q}{z}_j = \left[+ \quad + \quad \dots \quad - \quad + \quad - \right] \quad (5.5)$$

Com esse exemplo o cálculo de Δw_{jk} é dado pela equação 5.6.

$$\Delta w_{jk}^q = \alpha \cdot \delta_k^q \cdot \overset{q}{z}_j = \left(\alpha \cdot \begin{bmatrix} \overset{q}{+} \\ - \\ \cdot \\ \cdot \\ - \end{bmatrix} \cdot \begin{bmatrix} + & + & \dots & - & + & - \end{bmatrix} \right)^T = \left[\begin{array}{cccccc} + & + & \dots & - & + & - \\ - & - & \dots & + & - & + \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ - & - & \dots & + & - & + \end{array} \right] \left. \vphantom{\begin{bmatrix} \overset{q}{+} \\ - \\ \cdot \\ \cdot \\ - \end{bmatrix}} \right\} n-1 \quad (5.6)$$

Para VBOs, quando o vetor tiver todos os seus valores iguais a “1”, todas as linhas da matriz de sinais Δw_{jk} terão o mesmo sinal dos valores de $\overset{q}{z}_j$. Para os demais vetores ortogonais metade das linhas da matriz de sinais Δw_{jk} terá sinal oposto ao sinal do j -ésimo valor de $\overset{q}{z}_j$ enquanto a outra metade terá o mesmo sinal dos valores de $\overset{q}{z}_j$. Para exemplificar, vamos considerar que $\overset{q}{z}_j$ tenha os mesmos sinais exibidos na equação 5.5. Nesse caso, os valores da equação 5.7 serão:

$$\Delta w_{jk}^q = \alpha \cdot \delta_k^q \cdot z_j^q = \left(\alpha \cdot \begin{bmatrix} + \\ + \\ \cdot \\ \cdot \\ - \\ - \end{bmatrix} \cdot \begin{bmatrix} + & + & \dots & - & + & - \end{bmatrix} \right)^T = \left[\begin{array}{cccccc} + & + & \dots & - & + & - \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ + & + & \dots & - & + & - \\ - & - & \dots & + & - & + \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ - & - & \dots & + & - & + \end{array} \right] \left. \begin{array}{l} \left. \vphantom{\begin{matrix} + \\ + \\ \cdot \\ \cdot \\ - \\ - \end{matrix}} \right\} \frac{n}{2} \\ \left. \vphantom{\begin{matrix} + \\ + \\ \cdot \\ \cdot \\ - \\ - \end{matrix}} \right\} \frac{n}{2} \end{array} \right) \quad (5.7)$$

Já para $\Delta w0_k$, a interferência do tipo de vetor-alvo é mais acentuada. Os valores dessa matriz de sinais são obtidos pela multiplicação do escalar α pelos valores de δ_k . Com VBCs, VBNs ou VNOs, todos os valores de $\Delta w0_k$ serão negativos exceto por um valor na posição “i”. Veja, por exemplo, a propagação do 1º padrão de treinamento com o uso de qualquer um desses vetores-alvo em 5.8. No caso dos VBOs, todos os elementos serão positivos para o vetor que tem todas as entradas iguais a “1”. Para os outros vetores bipolares e ortogonais, metade desses valores será positiva e a outra metade negativa como mostra 5.9.

$$\Delta w0_k^q = \alpha \cdot \delta_k^q = \alpha \cdot \left[+ \underbrace{- \dots -}_{n-1} \right]^T = \left[+ \underbrace{- \dots -}_{n-1} \right]^T \quad (5.8)$$

$$\Delta w0_k^q = \alpha \cdot \left[\underbrace{+ \dots +}_{\frac{n}{2}} \underbrace{- \dots -}_{\frac{n}{2}} \right]^T = \left[\underbrace{+ \dots +}_{\frac{n}{2}} \underbrace{- \dots -}_{\frac{n}{2}} \right]^T \quad (5.9)$$

O uso dos VBOs tem capacidade maior de preservar as características do sinal do padrão em propagação. Isso possibilita um mapeamento mais eficiente, pois o reconhecimento de cada padrão em particular, é feito de forma mais independente e com uma convergência mais direcionada.

5.3 Atualização dos pesos entre a camada de entrada e a camada intermediária

Consideremos a propagação de dois padrões de treinamento consecutivos, sendo um padrão de ordem q e o outro padrão de ordem $q + 1$. Para o padrão de ordem q , tomemos pontualmente os cálculos dos vetores de atualização expressados nas equações 5.10, 5.11 e 5.12. Para o padrão de ordem $q + 1$, vamos analisar a determinação de $\delta^{q+1}_{in_j}$ que é mostrado na equação 5.13.

$$\delta_k^q = \left(t_k^q - y_k^q \right) \cdot f' \left(y_{in_k}^q \right) \quad (5.10)$$

$$\Delta w_{jk}^q = \alpha \cdot \delta_k^q \cdot z_j^q \quad (5.11)$$

$$w_{jk}^{q+1} = w_{jk}^q + \Delta w_{jk}^q \quad (5.12)$$

$$\delta_{in_j}^{q+1} = \sum_{k=1}^m \left[\delta_k^{q+1} \cdot w_{jk}^{q+1} \right] \quad (5.13)$$

Substituímos as equações 5.10 e 5.12 na equação 5.13 obtendo-se a equação 5.14.

$$\delta_{in_j}^{q+1} = \sum_{k=1}^m \left\{ \left[\left(t_k^{q+1} - y_k^{q+1} \right) \cdot f' \left(y_{in_k}^{q+1} \right) \right] \cdot \left[w_{jk}^q + \Delta w_{jk}^q \right] \right\} \quad (5.14)$$

Substituindo a expressão 5.11 na expressão 5.14 passamos a ter:

$$\delta \dot{in}_j^{q+1} = \sum_{k=1}^m \left\{ \left[\begin{array}{c} t_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \\ - y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \end{array} \right] \cdot \left[w_{jk}^q + \alpha \cdot \delta_k^q \cdot z_j^q \right] \right\} \quad (5.15)$$

Realiza-se substituição da expressão 5.10 na expressão 5.15, obtendo-se a expressão 5.16.

$$\delta \dot{in}_j^{q+1} = \sum_{k=1}^m \left\{ \left[\begin{array}{c} t_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \\ - y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \end{array} \right] \cdot \left[w_{jk}^q + \alpha \cdot \left(t_k^q - y_k^q \right) \cdot f' \left(yin_k^q \right) \cdot z_j^q \right] \right\} \quad (5.16)$$

Se trabalhada a expressão 5.16 resultará em:

$$\delta \dot{in}_j^{q+1} = \sum_{k=1}^m \left\{ \left[\begin{array}{c} t_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \\ - y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \end{array} \right] \cdot \left[\begin{array}{c} w_{jk}^q + \alpha \cdot t_k^q \cdot f' \left(yin_k^q \right) \cdot z_j^q \\ - \alpha \cdot y_k^q \cdot f' \left(yin_k^q \right) \cdot z_j^q \end{array} \right] \right\} \quad (5.17)$$

Pela propriedade distributiva da adição pela multiplicação de matrizes, obtemos a equação 5.18.

$$\delta \dot{in}_j^{q+1} = \sum_{k=1}^m \left[\begin{array}{c} \left(t_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \right) \cdot \left(w_{jk}^q \right) + \left(t_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \right) \cdot \left(\alpha \cdot t_k^q \cdot f' \left(yin_k^q \right) \cdot z_j^q \right) \\ - \left(y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \right) \cdot \left(\alpha \cdot y_k^q \cdot f' \left(yin_k^q \right) \cdot z_j^q \right) - \left(y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \right) \cdot \left(w_{jk}^q \right) \\ - \left(y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \right) \cdot \left(\alpha \cdot t_k^q \cdot f' \left(yin_k^q \right) \cdot z_j^q \right) \\ + \left(y_k^{q+1} \cdot f' \left(yin_k^{q+1} \right) \right) \cdot \left(\alpha \cdot y_k^q \cdot f' \left(yin_k^q \right) \cdot z_j^q \right) \end{array} \right] \quad (5.18)$$

A expressão obtida na equação 5.18, é mostrada em forma vetorial na expressão dada na equação 5.19.

$$\begin{aligned}
\delta^{q+1} \mathbf{in}_j &= \begin{bmatrix} t_1 & t_2 & \dots & t_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \begin{bmatrix} w_{11} & w_{21} & \dots & w_{j1} \\ w_{21} & w_{22} & \dots & w_{2k} \\ \cdot & \cdot & \dots & \cdot \\ w_{j1} & w_{j2} & \dots & w_{jk} \end{bmatrix}^q + \\
&\begin{bmatrix} t_1 & t_2 & \dots & t_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \alpha \cdot \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ t_k \end{bmatrix}^q * \begin{bmatrix} y'_1 \\ y'_2 \\ \cdot \\ y'_k \end{bmatrix}^q \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q - \\
&\begin{bmatrix} t_1 & t_2 & \dots & t_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \alpha \cdot \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ y_k \end{bmatrix}^q * \begin{bmatrix} y'_1 \\ y'_2 \\ \cdot \\ y'_k \end{bmatrix}^q \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q - \\
&\begin{bmatrix} y_1 & y_2 & \dots & y_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \begin{bmatrix} w_{11} & w_{21} & \dots & w_{j1} \\ w_{21} & w_{22} & \dots & w_{2k} \\ \cdot & \cdot & \dots & \cdot \\ w_{j1} & w_{j2} & \dots & w_{jk} \end{bmatrix}^q - \\
&\begin{bmatrix} y_1 & y_2 & \dots & y_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \alpha \cdot \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ t_k \end{bmatrix}^q * \begin{bmatrix} y'_1 \\ y'_2 \\ \cdot \\ y'_k \end{bmatrix}^q \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q + \\
&\begin{bmatrix} y_1 & y_2 & \dots & y_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \alpha \cdot \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ y_k \end{bmatrix}^q * \begin{bmatrix} y'_1 \\ y'_2 \\ \cdot \\ y'_k \end{bmatrix}^q \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q
\end{aligned} \tag{5.19}$$

Da expressão 5.19 extraímos o segundo termo que é representado pela equação 5.20.

$$\begin{bmatrix} t_1 & t_2 & \dots & t_k \end{bmatrix}^{q+1} * \begin{bmatrix} y'_1 & y'_2 & \dots & y'_k \end{bmatrix}^{q+1} \cdot \alpha \cdot \begin{bmatrix} t_1 \\ t_2 \\ \cdot \\ t_k \end{bmatrix}^q * \begin{bmatrix} y'_1 \\ y'_2 \\ \cdot \\ y'_k \end{bmatrix}^q \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \quad (5.20)$$

A expressão dada pela equação 5.20 possui uma particularidade. Ela compreende vetores-alvo diferentes. A interação desses vetores influencia no resultado da expressão quando são usados VBNs, VBCs, VNOs ou VBOs. Para visualizar a diferença existente, relembramos que α é um escalar positivo (taxa de aprendizagem) e que todos os outros elementos da expressão são vetores. Consideremos esses vetores com seus valores genéricos a seguir:

$$t_k^{q+1} = \begin{bmatrix} a_1 & a_2 & \dots & a_k \end{bmatrix} \quad (5.21)$$

$$f' \left(y_{in_k}^{q+1} \right) = \begin{bmatrix} m_1 & m_2 & \dots & m_k \end{bmatrix} \quad (5.22)$$

$$t_k^q = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_k \end{bmatrix} \quad (5.23)$$

$$f'(y^n_k) = \begin{bmatrix} n_1 \\ n_2 \\ \cdot \\ \cdot \\ n_k \end{bmatrix} \quad (5.24)$$

$$z_j^q = [z_1 \ z_2 \ \dots \ z_j] \quad (5.25)$$

Dessa forma o produto da expressão 5.20 pode ser reescrito na expressão 5.26 da seguinte forma:

$$\left(\left[a_1 \ a_2 \ \dots \ a_k \right] * \left[m_1 \ m_2 \ \dots \ m_k \right] \right) \cdot \alpha \cdot \left(\begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_k \end{bmatrix} * \begin{bmatrix} n_1 \\ n_2 \\ \cdot \\ \cdot \\ n_k \end{bmatrix} \right) \cdot [z_1 \ z_2 \ \dots \ z_j] \quad (5.26)$$

Resolvendo apenas os produtos especiais de multiplicação de elemento por elemento, ficamos com:

$$\left(\left[a_1 m_1 \ a_2 m_2 \ \dots \ a_k m_k \right] \cdot \alpha \cdot \begin{bmatrix} b_1 n_1 \\ b_2 n_2 \\ \cdot \\ \cdot \\ b_k n_k \end{bmatrix} \right) \cdot [z_1 \ z_2 \ \dots \ z_j] \quad (5.27)$$

$$\left(\alpha \cdot \begin{bmatrix} a_1 m_1 & a_2 m_2 & \dots & a_k m_k \end{bmatrix} \cdot \begin{bmatrix} b_1 n_1 \\ b_2 n_2 \\ \cdot \\ \cdot \\ b_k n_k \end{bmatrix}^T \right) \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix} \quad (5.28)$$

$$(\alpha \cdot [a_1 m_1 b_1 n_1 + a_2 m_2 b_2 n_2 + \dots + a_k m_k b_k n_k]) \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix} \quad (5.29)$$

Da expressão 5.29 consideremos apenas o termo dado pela expressão 5.30.

$$(a_1 b_1 m_1 n_1 + a_2 b_2 m_2 n_2 + \dots + a_k b_k m_k n_k) \quad (5.30)$$

Os produtos $m_i n_i$ para $1 \leq i \leq k$ serão todos positivos, pois são resultados da derivada da função de ativação que é sempre crescente, tendo consequentemente valores positivos pela ascendência da função. Os valores de a_i e b_i para $1 \leq i \leq k$, dependem do tipo de vetor-alvo utilizado. Eles assumem valores iguais a “0”, “1” ou “-1”.

Vetores-alvo convencionais têm todos os elementos iguais a “-1” exceto por um elemento na posição “ i ” para $1 \leq i \leq k$ que será igual a “1”. Dois vetores-alvo convencionais tem o elemento “1” em posições distintas. Logo o resultado de dois dos produtos $a_i b_i$ para $1 \leq i \leq k$ será “-1”. Os demais produtos serão todos iguais a “1”.

Vetores-alvo binários têm todos os elementos iguais a “0”, exceto por um elemento na posição “ i ” que é igual a “1”. O resultado de todos os produtos $a_i b_i$ para $1 \leq i \leq k$ será sempre “0”, pois dois vetores-alvo diferentes tem o elemento “1” em posições distintas. Assim, o termo destacado na expressão 5.30 será nulo, tornando nulo o vetor correspondente ao item da expressão 5.35.

Por outro lado, quando os vetores são ortogonais bipolares, metade dos produtos $a_i b_i$ serão iguais a “1” e a outra metade será igual a “-1”.

Considerando vetores-alvo com dimensões maiores do que quatro, podemos afirmar

que a expressão 5.30 terá módulo sempre maior quando os vetores-alvo forem do tipo VBC ou VNO.

Para exemplificar, consideremos que os vetores-alvo sejam VBCs de tamanho 16. Tomemos também, por exemplo, que os valores de $f'(y_{in_k}^{q+1})$ e $f'(y_{in_k}^q)$ sejam dados pelas expressões 5.31 e 5.32. Consideremos ainda que a taxa de aprendizagem “ α ” seja igual a 0,01.

$$f'(y_{in_k}^{q+1}) = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}}_{16} \quad (5.31)$$

$$\left[f'(y_{in_k}^{q+1}) \right]^T = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}}_{16} \quad (5.32)$$

$$t_k^{q+1} = \underbrace{\begin{bmatrix} 1 & -1 & -1 & -1 & \dots & -1 \end{bmatrix}}_{16} \quad (5.33)$$

$$\left[t_k^q \right]^T = \underbrace{\begin{bmatrix} -1 & 1 & -1 & -1 & \dots & -1 \end{bmatrix}}_{16} \quad (5.34)$$

Assim sendo, a expressão 5.26 será dada pelas expressões 5.35, 5.36, 5.37 e 5.38:

$$\underbrace{\begin{bmatrix} 1 & -1 & \dots & -1 & -1 \end{bmatrix}}_{1 \times 16} * \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \end{bmatrix}}_{1 \times 16} \cdot 0,01 \bullet \underbrace{\begin{bmatrix} -1 \\ 1 \\ -1 \\ \cdot \\ \cdot \\ -1 \end{bmatrix}}_{16 \times 1} * \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix}}_{16 \times 1} \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix} \quad (5.35)$$

$$\underbrace{\begin{bmatrix} 1 \cdot 1 & (-1) \cdot 1 & (-1) \cdot 1 & \dots & (-1) \cdot 1 \end{bmatrix}}_{1 \times 16} \cdot 0,01 \bullet \underbrace{\begin{bmatrix} (-1) \cdot 1 \\ 1 \cdot 1 \\ (-1) \cdot 1 \\ \cdot \\ \cdot \\ (-1) \cdot 1 \end{bmatrix}}_{16 \times 1} \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \quad (5.36)$$

$$\underbrace{\begin{bmatrix} 1 & -1 & -1 & \dots & -1 \end{bmatrix}}_{1 \times 16} \cdot 0,01 \bullet \underbrace{\begin{bmatrix} -1 \\ 1 \\ -1 \\ \cdot \\ \cdot \\ -1 \end{bmatrix}}_{16 \times 1} \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \quad (5.37)$$

$$\begin{aligned} &= \underbrace{\left[1 \cdot (-1) + (-1) \cdot 1 + (-1) \cdot (-1) + \dots + (-1) \cdot (-1) \right]}_{16} \cdot 0,01 \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \\ &= \underbrace{\left[(-1) + (-1) + 1 + 1 + \dots + 1 \right]}_{16} \cdot 0,01 \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \\ &= 14 \cdot 0,01 \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \end{aligned} \quad (5.38)$$

Considere agora que os vetores-alvo sejam ortogonais de tamanho 16, representados pelas expressões 5.39 e 5.40 e que $f' \left(\overset{q+1}{yin}_k \right)$ e $f' \left(\overset{q}{yin}_k \right)$ tenham seus valores preservados.

$$\overset{q+1}{t}_k = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}}_{16} \quad (5.39)$$

$$\begin{bmatrix} q \\ t_k \end{bmatrix}^T = \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & -1 & \dots & -1 & -1 \end{bmatrix}}_{16} \quad (5.40)$$

Nesse caso a expressão 5.38 pode ser substituída pela expressão 5.41.

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \end{bmatrix}}_{1 \times 16} * \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \end{bmatrix}}_{1 \times 16} \cdot 0,01 \bullet \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ 1 \\ -1 \\ \cdot \\ \cdot \\ -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \quad (5.41)$$

$\underbrace{\hspace{1.5cm}}_{16 \times 1} \quad \underbrace{\hspace{1.5cm}}_{16 \times 1}$

$$\underbrace{\begin{bmatrix} 1 \cdot 1 & 1 \cdot 1 & \dots & 1 \cdot 1 & 1 \cdot 1 \end{bmatrix}}_{1 \times 16} \cdot 0,01 \bullet \begin{bmatrix} 1 \cdot 1 \\ \cdot \\ \cdot \\ 1 \cdot 1 \\ (-1) \cdot 1 \\ \cdot \\ \cdot \\ (-1) \cdot 1 \end{bmatrix} \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \quad (5.42)$$

$\underbrace{\hspace{1.5cm}}_{16 \times 1}$

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \end{bmatrix}}_{1 \times 16} \cdot 0,01 \bullet \underbrace{\begin{bmatrix} 1 \\ \cdot \\ \cdot \\ 1 \\ -1 \\ \cdot \\ \cdot \\ -1 \end{bmatrix}}_{16 \times 1} \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \quad (5.43)$$

$$\begin{aligned} &= \underbrace{[1 \cdot 1 + \dots + 1 \cdot 1 + 1 \cdot (-1) + \dots + 1 \cdot (-1)]}_{16} \cdot 0,01 \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \\ &= \underbrace{[1 + \dots + 1 + (-1) + \dots + (-1)]}_{16} \cdot 0,01 \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \\ &= 0 \cdot 0,01 \cdot \begin{bmatrix} z_1 & z_2 & \dots & z_j \end{bmatrix}^q \end{aligned} \quad (5.44)$$

5.4 Confirmação da hipótese formulada

Como foi admitido que os valores de $f' \left(\overset{q+1}{yin_k} \right)$ e $f' \left(\overset{q}{yin_k} \right)$ eram todos iguais a 1, a expressão 5.44 tornou-se um vetor nulo. Mesmo quando esses valores são diferentes entre si, é possível visualizar que o termo da expressão 5.30 será sempre menor quando são utilizados VBOs. A expressão 5.30 age como um fator potencializador do sinal correspondente à expressão 5.20. Os valores que forem negativos na matriz $\overset{q}{z_j}$ serão menores quando utilizados vetores-alvo não ortogonais. Por outro lado os valores positivos serão maiores quando utilizados os mesmos vetores.

Como a expressão 5.20 é um termo do cálculo de δin_k , podemos visualizar a influência dos vetores-alvo no aprendizado da rede. A interação entre dois vetores-alvo numa expressão matemática de produto matricial pode ser entendida como um ruído de uma saída sobre a outra.

O uso de VBOs minimiza essa propagação de ruído. Ocorre então um refinamento na aprendizagem da RNA.

Considerando que δin_k é usado no cálculo de δ_k dado pela expressão 5.45 e que este por sua vez é usado nos cálculos de Δv_{ij} da expressão 5.46 e de $\Delta v0_j$ da expressão 5.48 é notório o efeito deste vetor sobre a atualização dos pesos da rede. O ruído propagado interfere no treinamento da MLP. Os VBOs são mais distantes entre si do que os VNOs. Essa maior distância euclidiana facilita a aprendizagem da rede.

$$\delta_j^{q+1} = \delta in_j^{q+1} \cdot f' \left(zin_j^{q+1} \right) \quad (5.45)$$

$$\Delta v_{ij}^{q+1} = \alpha \cdot \delta_j^{q+1} \cdot x_i^{q+1} = \alpha \cdot \left[\delta in_j^{q+1} \cdot f' \left(zin_j^{q+1} \right) \right] \cdot x_i^{q+1} \quad (5.46)$$

$$\Delta v_{ij}^{q+1} = \left(\alpha \cdot \left(\left[\delta in_1^{q+1} \quad \delta in_2^{q+1} \quad \dots \quad \delta in_j^{q+1} \right] * \left[f' \left(zin_1^{q+1} \right) \quad f' \left(zin_2^{q+1} \right) \quad \dots \quad f' \left(zin_j^{q+1} \right) \right] \right) \cdot \left[x_1^{q+1} \quad x_2^{q+1} \quad \dots \quad x_i^{q+1} \right] \right)^T \quad (5.47)$$

$$\Delta v0_j^{q+1} = \alpha \cdot \delta_j^{q+1} = \alpha \cdot \left(\left[\delta in_1^{q+1} \quad \delta in_2^{q+1} \quad \dots \quad \delta in_j^{q+1} \right] * \left[f' \left(zin_1^{q+1} \right) \quad f' \left(zin_2^{q+1} \right) \quad \dots \quad f' \left(zin_j^{q+1} \right) \right] \right) \quad (5.48)$$

Na prática essa minimização do ruído mostrada por meio da análise matemática, possibilita um distanciamento dos pontos de saída no treinamento entre padrões diferentes em conformidade com a hipótese formulada.

Capítulo 6

Análise experimental - Classificação de dígitos manuscritos

6.1 Introdução

Neste capítulo apresenta-se resultados experimentais que mostram a superioridade dos vetores-alvo bipolares e ortogonais sobre os vetores normalmente utilizados como alvos de redes do tipo MLP no reconhecimento de padrões. Os experimentos foram realizados para a classificação dos dígitos manuscritos utilizados no estudo das regiões de convergência da seção 4.1.

Os dígitos manuscritos utilizados nos experimentos possuem um alto grau de degradação e irregularidade. Como citado na subseção 4.1.2 são dígitos obtidos de 80 pessoas. É fato que cada pessoa tem um tipo de caligrafia o que representa um obstáculo à padronização dos dados. Também deve ser lembrado que cada pessoa foi solicitada a escrever o mesmo dígito duas vezes, sendo que na primeira deveria escrever com calma e primando pela perfeição e na segunda vez, escrever com rapidez ignorando a perfeição. Dessa maneira, a classificação desse conjunto de padrões representa uma tarefa complexa para a área de RPs ou mais especificamente para RNAs.

É importante salientar que VBOs são aplicáveis a problemas como este, em que os

padrões são demasiadamente degradados, irregulares ou com grande quantidade de ruídos. Padrões que contenham pouca irregularidade, ruído ou que sejam pouco degradados podem ser classificados com pouco esforço através das técnicas convencionais.

6.2 Dificuldades

Encontrar a combinação de parâmetros de treinamento de uma RNA não é uma tarefa simples. Embora existam indicações para a escolha do número de neurônios da camada intermediária de uma MLP (FAUSETT, 1994) e para a escolha da taxa de aprendizagem, as simulações realizadas não obtiveram as melhores taxas de reconhecimento com elas. Na prática, as melhores taxas encontradas e mostradas neste trabalho foram conseguidas através de muita investigação e persistência ante as inúmeras possibilidades de combinações.

Outro fator que representou grande dificuldade foi a lentidão da execução das simulações. O objetivo era a comparação fiel entre os desempenhos de diferentes vetores-alvo. Para isso, os pesos sinápticos foram mantidos fixos para todas as simulações ocorrendo apenas a inclusão de novos pesos quando a rede tinha a sua arquitetura ampliada em função dos diferentes tamanhos de vetores-alvo ou pelo número de neurônios da camada intermediária que foi de 100 ou 200. O simulador foi criado em linguagem C++ que por ser compilada apresenta uma velocidade maior de execução. Mesmo assim algumas simulações necessitaram de uma grande quantidade de épocas para atingirem o erro tolerado, especificado como parâmetro.

Algumas simulações levaram mais de 50000 ciclos, quando o experimento ainda se encontrava na fase de determinações dos melhores parâmetros. Computadores comuns precisaram de mais de 2 horas para concluir a simulação. Já um computador especial contendo memória RAM de 16 GB precisou de aproximadamente 1 hora e 20 minutos para executar simulações com um número aproximado de 50000 ciclos. Mesmo encontrando a combinação de parâmetros com as melhores taxas de reconhecimento, algumas simulações levaram cerca de 17000 ciclos para o encerramento do treinamento. Houve uma redução significativa de tempo. Mesmo assim foi necessário um longo tempo até a conclusão dos experimentos.

6.3 Motivação

A motivação principal do trabalho era realmente a avaliação do desempenho dos diferentes tipos de vetores-alvo no reconhecimento de um tipo de padrão degradado. Contudo, a melhoria do desempenho das RNAs pode ajudar na solução de inúmeros problemas da engenharia e das ciências como um todo. O potencial dessa importante ferramenta já foi discutido anteriormente.

No caso de dígitos manuscritos, alguns estudos, também já comentados anteriormente, têm utilizado redes neurais para o controle automático de tráfego de veículos. Redes mais robustas, capazes de lidar com problemas de aquisição de imagens como luminosidade, poeira e umidade poderão ser essenciais num futuro próximo para a criação de melhoria de dispositivos para esse fim.

O reconhecimento de dígitos ou letras pode inclusive vir a auxiliar na criação de sistemas que sejam capazes de reconhecer textos ou números em uma folha de papel, convertendo a informação em som para deficientes visuais. Existem inúmeras possibilidades de aplicações para RNA e a ampliação de seu potencial e eficiência é de extrema importância para a ciência.

6.4 Procedimento experimental

A RNA utilizada neste experimento é uma MLP com 256 neurônios na camada de entrada. Para a camada intermediária, foram realizadas simulações com a utilização de 100 e 200 neurônios escondidos. Na camada de saída, foram feitos testes com 10 e 64 neurônios, de acordo com o tipo de vetor-alvo escolhido para treinamento. Numa primeira etapa, foram utilizados VBOs com 64 elementos. Num segundo momento, foram utilizados VNOs também com 64 elementos. Finalmente, a rede foi simulada com VBCs contendo 10 elementos. Nos experimentos, foram utilizadas taxas de aprendizagem iguais a 0,01 e a 0,005, com critério de parada efetuado através do alcance do erro tolerado ou pelo limite máximo de 80000 ciclos. Para todos os tipos de simulação, foram mantidos os pesos sinápticos que foram gerados de

forma aleatória.

Os dados de treinamento e teste são os dígitos manuscritos descritos na subseção 4.1.2. O total de dígitos presentes nessa biblioteca é de 1593. Nesse procedimento experimental escolhemos 200 amostras, sendo 20 de cada dígito para o treinamento da rede. Os outros 1393 foram utilizados na fase de teste para avaliar o desempenho da rede. Uma amostra de 100 dígitos é ilustrada na Figura 6.1.

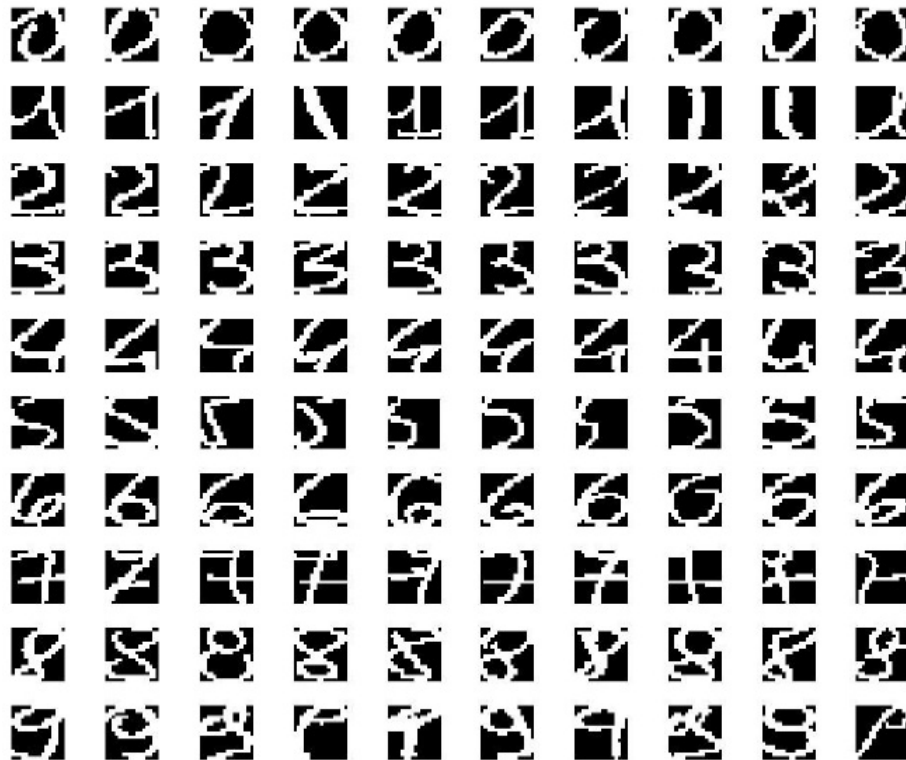


Figura 6.1: Amostra de dígitos manuscritos submetidos ao treinamento
Fonte: (REPOSITORY, 1994)

O treinamento de uma Rede MLP pode ser finalizado de duas formas: parada por erro e parada por limite de ciclos. No caso de parada por erro, consideramos os seguintes cálculos:

- $E = \frac{1}{2} \sum_{j=1}^{N_s} (d_j - y_j)^2$ onde E é o erro quadrático para um padrão; N_s é o número de neurônios de saída; d_j é a saída desejada para o neurônio j ; y_j é o valor de saída do neurônio j ;
- $E_m = \frac{1}{N} \sum_{p=1}^N E_p$ onde E_m é o erro quadrático médio de todos os padrões para cada ciclo de treinamento; N é o número de padrões; E_p é o erro quadrático para o padrão

p calculado pela equação anterior;

Dessa forma, o treinamento é finalizado quando a condição $E_m < \varepsilon$ for satisfeita. O valor de ε corresponde ao valor de *Tolerância* fornecido na simulação. Se a condição do erro não for satisfeita e o número de ciclos for igual a 80000, o treinamento será finalizado.

Após a determinação das combinações de parâmetros mais eficientes para o experimento, as simulações finais foram feitas em computadores com a mesma capacidade de processamento. Foram três computadores com processadores do tipo INTEL(R) CORE(TM) 2 DUO CPU E7500 - 2,93GHz e memória (RAM) de 2,00 GB.

6.5 Resultados experimentais

Embora a MLP tenha sido projetada com 200 e com 100 neurônios na camada intermediária, as melhores taxas de reconhecimento ocorreram usando 200 neurônios. Várias combinações de parâmetros foram realizadas, sempre no intuito de obter as melhores taxas de reconhecimento para cada tipo de vetor-alvo utilizado. Assim, há a garantia de que as comparações sejam as mais justas possíveis.

Os resultados são apresentados nas tabelas 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 6.11 e 6.12. Em cada uma delas, a primeira coluna identifica o número da simulação realizada pelo programa. A segunda coluna, denominada Tolerância, representa o erro máximo tolerado no treinamento da rede. A terceira coluna mostra o número de ciclos necessários para encerrar o treinamento da rede. Finalmente, a quarta coluna representa o percentual de padrões reconhecidos pela rede treinada.

As tabelas de 6.2 a 6.12 apresentam os resultados da rede treinada com vetores-alvo dos tipos VBCs, VNOs e VBOs. Nessas simulações, foram usadas as taxas de aprendizagem de 0,01 e 0,005. Também houve alternância do número de neurônios da camada escondida com valores iguais a 100 e 200.

Os valores sublinhados e em negrito simbolizam as melhores taxas de reconhecimento que foram obtidas. Os resultados mostram que em todos os casos, o uso dos VBO propicia

Tabela 6.1: Análise do desempenho da MLP, usando VBCs, alfa igual a 0,01 e 200 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
701	$1 \cdot 10^{-2}$	354	82,63%
702	$8 \cdot 10^{-3}$	1325	83,99%
703	$6 \cdot 10^{-3}$	6393	83,27%
704	$4 \cdot 10^{-3}$	10328	82,77%

Tabela 6.2: Análise do desempenho da MLP, usando VNOs, alfa igual a 0,01 e 200 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
705	$8 \cdot 10^{-3}$	412	78,47%
706	$6 \cdot 10^{-3}$	1423	81,12%
707	$4 \cdot 10^{-3}$	6754	79,61%
708	$2 \cdot 10^{-3}$	12321	77,67%

Tabela 6.3: Análise do desempenho da MLP, usando VBOs, alfa igual a 0,01 e 200 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
709	$8 \cdot 10^{-3}$	501	86,79%
710	$6 \cdot 10^{-3}$	1723	87,22%
711	$4 \cdot 10^{-3}$	7213	87,72%
712	$2 \cdot 10^{-3}$	14723	87,07%

Tabela 6.4: Análise do desempenho da MLP, usando VBCs, alfa igual a 0,01 e 100 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
713	$6 \cdot 10^{-3}$	378	82,27%
714	$4 \cdot 10^{-3}$	1272	83,78%
715	$2 \cdot 10^{-3}$	6318	85,71%
716	$1 \cdot 10^{-3}$	10721	84,92%

Tabela 6.5: Análise do desempenho da MLP, usando VNOs, alfa igual a 0,01 e 100 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
717	$6 \cdot 10^{-3}$	391	77,96%
718	$4 \cdot 10^{-3}$	1305	81,55%
719	$2 \cdot 10^{-3}$	6991	82,77%
723	$1 \cdot 10^{-3}$	11255	82,20%

Tabela 6.6: Análise do desempenho da MLP, usando VBOs, alfa igual a 0,01 e 100 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
724	$8 \cdot 10^{-3}$	407	86,50%
725	$6 \cdot 10^{-3}$	1522	86,86%
728	$4 \cdot 10^{-3}$	7544	87,65%
729	$2 \cdot 10^{-3}$	13357	87,08%

Tabela 6.7: Análise do desempenho da MLP, usando VBCs, alfa igual a 0,005 e 200 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
730	$8 \cdot 10^{-3}$	341	81,77%
731	$6 \cdot 10^{-3}$	1211	83,42%
732	$4 \cdot 10^{-3}$	6278	85,71%
733	$2 \cdot 10^{-3}$	10157	84,21%

Tabela 6.8: Análise do desempenho da MLP, usando VNOs, alfa igual a 0,005 e 200 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
731	$1 \cdot 10^{-2}$	403	81,69%
732	$8 \cdot 10^{-3}$	1308	83,42%
739	$6 \cdot 10^{-3}$	6791	80,62%
740	$4 \cdot 10^{-3}$	11001	80,26%

Tabela 6.9: Análise do desempenho da MLP, usando VBOs, alfa igual a 0,005 e 200 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
741	$1 \cdot 10^{-2}$	605	86,14%
742	$8 \cdot 10^{-3}$	1521	87,87%
743	$6 \cdot 10^{-3}$	8056	86,57%
744	$4 \cdot 10^{-3}$	15124	85,93%

Tabela 6.10: Análise do desempenho da MLP, usando VBCs, alfa igual a 0,005 e 100 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
745	$1 \cdot 10^{-2}$	318	82,99%
746	$8 \cdot 10^{-3}$	1197	83,35%
747	$6 \cdot 10^{-3}$	6005	83,70%
748	$4 \cdot 10^{-3}$	9137	83,42%

Tabela 6.11: Análise do desempenho da MLP, usando VNOs, alfa igual a 0,005 e 100 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
749	$6 \cdot 10^{-3}$	389	78,03%
750	$4 \cdot 10^{-3}$	1276	80,90%
751	$2 \cdot 10^{-3}$	6624	83,99%
752	$1 \cdot 10^{-3}$	10954	82,70%

Tabela 6.12: Análise do desempenho da MLP, usando VBOs, alfa igual a 0,005 e 100 neurônios escondidos

Nº da Simulação	Tolerância	Nº de Ciclos	Taxa de Reconhecimento
753	$8 \cdot 10^{-3}$	575	86,86%
754	$6 \cdot 10^{-3}$	1501	86,79%
755	$4 \cdot 10^{-3}$	8002	87,43%
756	$2 \cdot 10^{-3}$	15012	86,36%

um melhor desempenho no reconhecimento de padrões. Em todos os experimentos, a taxa de aprendizagem foi mantida constante. O treinamento da rede é encerrado se o erro quadrático total atingir ao valor tolerado escolhido como parâmetro ou se for atingido o limite de 80000 ciclos.

Um resumo de todos os resultados apresentados nas tabelas anteriores é feito no gráfico da Figura 6.2.

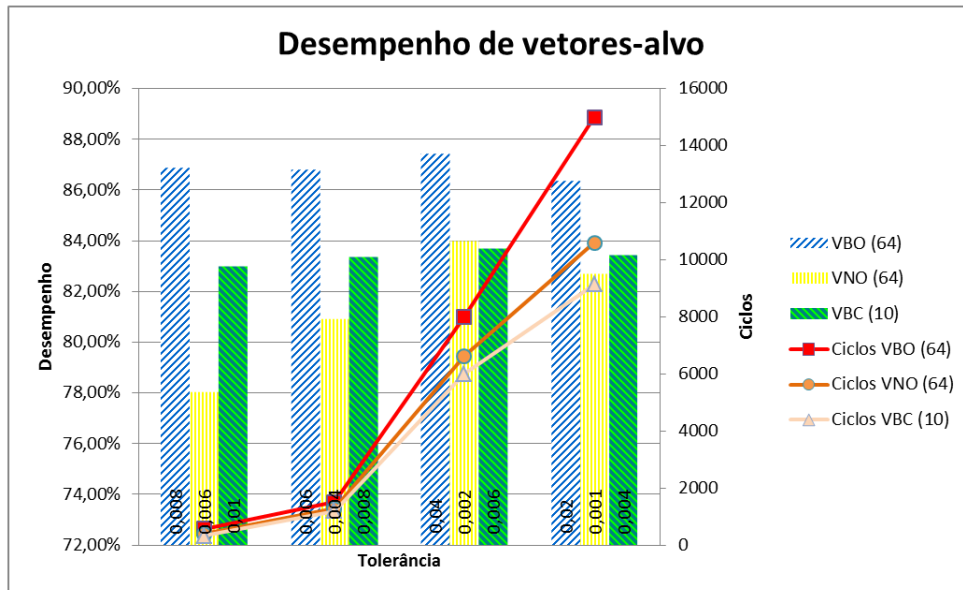


Figura 6.2: Resultados do reconhecimento de dígitos

6.6 Estatística de comparação

Do ponto de vista estatístico, foram utilizados diferentes tipos de vetores-alvo sobre uma mesma amostra. Assim sendo, foi realizado um teste estatístico para a proporção de dígitos classificados corretamente, comparando-se os melhores desempenhos obtidos com o uso de vetores-alvo de abordagem convencional, com os melhores desempenhos obtidos com o uso de VBOs. O melhor desempenho para o uso de vetores-alvo de abordagem convencional, deu-se com a rede treinada com VBCs de tamanho 10, 100 neurônios na camada intermediária e taxa de aprendizagem $\alpha = 0,005$, onde a taxa de reconhecimento foi de 85,71%. O melhor desempenho para o uso de VBOs, deu-se com a rede treinada com 200 neurônios na camada intermediária e taxa de aprendizagem $\alpha = 0,005$, onde a taxa de reconhecimento foi de 87,87%.

Para uma proporção p_0 e uma proporção \hat{p} em uma amostra de tamanho n , a estatística de comparação é dada pela equação 6.1.

$$z_{cal} = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \quad (6.1)$$

Se módulo do valor calculado de z_{cal} por meio da equação 6.1, for maior ou igual que

o valor z_{tab} da tabela normal padronizada, considera-se que \hat{p} é diferente de p_0 . Caso contrário, considera-se que as referidas proporções não são diferentes (DEVORE, 2006). Para o caso em questão, considera-se que: $\hat{p} = 0,8787$, $p_0 = 0,8571$ e que $n = 1393$. Dessa maneira, a estatística de teste é dada pela equação 6.2.:

$$z_{cal} = \frac{0,8787 - 0,8571}{\sqrt{\frac{0,8571(1-0,8571)}{1393}}} \approx 2,3035 \quad (6.2)$$

Considerando-se um nível de 95% de confiança, o valor correspondente na tabela normal padronizada é $z_{tab} = 1,96$. Como $|z_{cal}| \geq z_{tab}$, pode-se afirmar que as proporções são diferentes. Portanto, ao nível de 5% de significância, verifica-se que a proporção de dígitos classificados corretamente é maior quando são utilizados VBOs como vetores-alvo da MLP.

6.7 Discussão dos resultados

A melhor dentre todas as taxas de reconhecimento se deu com a utilização de 200 neurônios na camada intermediária, taxa de aprendizagem igual a 0,005 e vetores-alvo bipolares e ortogonais de tamanho 64. Ela foi da ordem de 87,72%. Tomando por base as mesmas combinações de parâmetros nos casos considerados, redes treinadas com VBOs tiveram taxas de classificação em média 4% superiores aos demais vetores-alvo utilizados.

De todas as simulações, o melhor resultado de classificação para redes treinadas com VBCs foi de 85,71% quando as redes tinham 200 neurônios intermediários e taxa de aprendizagem de 0,005 e com 100 neurônios intermediários e taxa de aprendizagem de 0,01. Pode-se perceber em todas as tabelas da seção 6.5 a ocorrência de over-training, onde, com a redução do erro tolerado, ocorre a redução nas taxas de classificação.

Pelo gráfico da Figura 6.2 é possível perceber que as redes treinadas com vetores-alvo do tipo VBOs necessitam de um número de ciclos ligeiramente maior do que as redes treinadas com os vetores-alvo do tipo VNOs e VBCs. Em suma a nova proposta do uso de vetores-alvo VBOs mostrou resultados melhores do que os vetores-alvo convencionais. Isso significa que o

uso desses novos vetores-alvo melhora a capacidade de generalização da RNA na classificação de padrões degradados.

Capítulo 7

Análise experimental - Classificação de imagens de íris humana

7.1 Introdução

A partir da hipótese sobre vetores-alvo do tipo VBOs mostrada no capítulo ?? e dos resultados experimentais obtidos no capítulo 6 surgiu a preocupação em saber se os resultados eram aplicáveis apenas a padrões como dígitos manuscritos ou a imagens de placas de veículos. Dessa maneira optou-se por estudar a influência de vetores-alvo no reconhecimento de padrões biométricos. Realizou-se então o reconhecimento de íris humana usando VBOs de dimensão 128, VBCs de dimensão 71 e VNOs de dimensão 128.

Estudos envolvendo o reconhecimento de íris têm mostrado ótimos resultados com técnicas de RPs baseadas em abordagem estatística ou em RNA. As pesquisas têm mostrado que esse tipo de padrão tem alto grau de regularidade e confiabilidade. Os únicos fatores que podem introduzir ruídos são fatores externos de aquisição da imagem, como rotação, translação ou distância entre o olho e a câmera.

7.2 Dificuldades

Ao contrário dos dígitos manuscritos, as bibliotecas existentes na internet que disponibilizam imagens de íris, trazem consigo pequenas quantidades de imagens. No caso dos dígitos, houve a disponibilização de 1593 amostras de 10 dígitos. Algumas bibliotecas de íris disponibilizam 3 imagens, 5 imagens e quando muito 7 imagens. No treinamento de redes neurais, é interessante que seja utilizada uma quantidade significativa de amostras para cada padrão, de maneira a fornecer à rede a capacidade de conhecer as mais variadas diferenças. Isso garante à rede a capacidade de generalização.

Nos experimentos realizados nesse trabalho utilizou-se um banco de imagens de 108 indivíduos, onde alguns contém 7 imagens e os outros menos do que 7 imagens. Para uniformizar o treinamento e o teste, selecionamos apenas os indivíduos com 7 imagens cada um. Das 7 imagens foram utilizadas 4 para treinamento e 3 para teste.

A extração de características fez com que cada imagem resultasse em 8640 pixels. Esse número de dados dos padrões de entrada ocasiona o problema de que redes com arquiteturas muito grandes têm treinamentos demasiadamente longos. Uma das ações utilizadas para a minimização desse problema foi a escolha de apenas parte da imagem da íris. A imagem foi processada de forma circular, gerando 18 circunferências com 480 pixels cada uma. Foram escolhidas apenas as 5 circunferências mais internas pelo fato dessa região sofrer menos interferência das pálpebras e dos cílios. Com essa metodologia, cada padrão de treinamento e classificação conteve 2400 pixels que foram organizados em um vetor linha.

Outra alternativa para minimizar o tempo de simulação foi a utilização da *toolbox trainingdx* do software Matlab (HANSELMAN & LITTLEFIELD, 2003). Essa toolbox utiliza recursos como o momentum e taxa de aprendizagem adaptativa. Com essa escolha surge outro problema. A cada simulação o software Matlab utiliza pesos sinápticos aleatórios. Para que as comparações entre os diferentes vetores-alvo fossem justas, capturamos as matrizes de pesos iniciais da primeira execução e as utilizamos nas demais execuções. Por questões de arquitetura, as redes treinadas com VBOs e VNOs ambos com dimensão igual a 128 usaram os mesmos

pesos sinápticos iniciais, enquanto que a rede treinada com VBCs utilizaram pesos sinápticos diferentes.

7.3 Motivação

Íris são consideradas um tipo de padrão pouco suscetível a sugestões e outras interferências externas. Além disso, a probabilidade de que dois seres humanos tenham íris similares é praticamente nula (CARNEIRO & VEIGA, 2005). Esses fatores tem feito com que órgãos de segurança utilizem esse padrão para identificar indivíduos. Além disso, empresas tem utilizado a íris como chave de acesso a locais, a sistemas ou mesmo para o registro de frequência de funcionários.

Dessa maneira, técnicas que venham a contribuir para melhorias na classificação de íris são sempre bem vindas. Além do grande potencial que as íris oferecem a sistemas de segurança ou mesmo para controles de acesso, esses experimentos são fundamentais para mostrar que os vetores-alvo do tipo VBO propiciam melhores taxas de reconhecimento em relação aos vetores convencionais.

7.4 Procedimento experimental

7.4.1 Topologias adotadas

A determinação das topologias a serem utilizadas nesses experimentos foi dada através do método de Kolmogorov indicado por Silva et al. (2010). De acordo com essa técnica, o número n_1 de neurônios da camada intermediária é dado em função do número n de neurônios da camada de entrada como mostra a equação 7.1.

$$n_1 = 2 \cdot n + 1 \quad (7.1)$$

Em nosso problema, temos 2400 entradas o que de acordo com o método indicado,

indica a utilização de 4801 neurônios na camada escondida. Na camada de saída, foram feitos testes com 71 e 128 neurônios, de acordo com o tipo de vetor-alvo escolhido para treinamento. A escolha de vetores-alvo com 71 elementos foi feita em função do número de íris de indivíduos. Como são 71 indivíduos, a abordagem convencional necessita de 71 vetores-alvo diferentes. Logo, os vetores-alvo convencionais com 71 elementos são suficientes para o treinamento de 71 padrões. Os VBOs têm tamanhos correspondentes a uma potência de 2. Os VBOs gerados serão de tamanhos 2, 4, 8, 16, 32, 64, 128, 256 e assim por diante. A quantidade de vetores disponíveis para cada tamanho equivale ao tamanho do vetor. Portanto, para termos VBOs suficientes para esse experimento, foi necessária a utilização de vetores-alvo com 128 componentes. Também foram utilizados VNOs com 128 componentes, a fim de se ter uma comparação justa.

Dessa maneira, esse estudo experimental contou com as seguintes arquiteturas: 2400 x 4801 x 71 e 2400 x 4801 x 128.

7.4.2 Caracterização dos dados experimentais

Os dados de treinamento são íris de indivíduos do banco de dados da *Chinese Academy of Sciences - Institute of Automation - CASIA* (ACADEMY, 2003). Tratam-se de 108 indivíduos, sendo que 71 deles possuem 7 imagens e os demais possuem menos do que 7 imagens. Para uniformizar o treinamento da rede, foram utilizados somente os padrões dos indivíduos com 7 imagens, onde 4 delas formaram o conjunto de treinamento e 3 o conjunto de teste. Estas imagens foram obtidas especialmente para serem usadas em pesquisas de reconhecimento de íris, usando uma luz próxima da infravermelha. Portanto, as características da região da íris são facilmente visíveis e existe um bom contraste entre a pupila, a íris e a parte branca do olho.

7.4.3 Extração de características

O processamento da imagem da íris se dá em algumas etapas. A primeira delas é a localização da região da íris na imagem que é realizada utilizando a Transformada de Hough Circular. Em seguida a região da íris, que possui uma forma anelar, é normalizada para que seja

representada por uma matriz retangular. Finalmente é realizada a extração das características da íris. Neste trabalho, as características da íris foram extraídas convoluindo a imagem normalizada com o chamado filtro Log-Gabor. A filtragem dá origem a coeficientes complexos cujas fases são quantizadas para um dos quatro quadrantes do plano complexo. Cada quadrante é referenciado por dois bits e, assim, um template binário é criado (CARNEIRO & VEIGA, 2005). Para cada imagem são 8640 pixels organizados em 18 circunferências concêntricas contendo 480 pixels cada uma.

Nesse trabalho foram extraídas apenas as 5 circunferências mais internas da íris, o que elimina a interferência das pálpebras, dos cílios e reduz o esforço de treinamento da MLP. Assim, cada padrão de treinamento corresponde a um conjunto de $5 \times 480 = 2400$ pixels. Os pixels de cor branca receberam o valor “-1” e os pixels de cor preta receberam o valor “1”, de maneira que da circunferência mais interna para a mais externa, os vetores de treinamento foram montados em uma única linha contendo os 2400 pixels.

7.4.4 Processo de treinamento das redes

Nos experimentos, foi adotada a taxa de aprendizagem inicial $\alpha = 0,3$. De acordo com Silva et al. (2010) a taxa de aprendizagem deve ser escolhida de tal maneira que não seja pequena a ponto de tornar o treinamento muito longo e nem muito grande a ponto de causar instabilidades na convergência. Outra preocupação no treinamento de redes neurais artificiais é o ponto ideal de parada desse treinamento. À medida que o erro médio quadrático reduz, obtém-se uma maior precisão na classificação dos padrões. Contudo, se a rede for treinada para um erro demasiadamente pequeno, é possível que a rede se torne tão rigorosa a ponto de deixar de reconhecer corretamente um padrão que contenha pouca quantidade de ruído ou não muito diferente do padrão de treinamento. Esse fenômeno é chamado de *overtraining* que na língua Portuguesa pode ser interpretado como *excesso de treinamento*.

Uma das alternativas para se evitar esse problema é a técnica da parada prematura, mais conhecida como *early stopping*. A partir da amostra, são separados três sub-conjuntos.

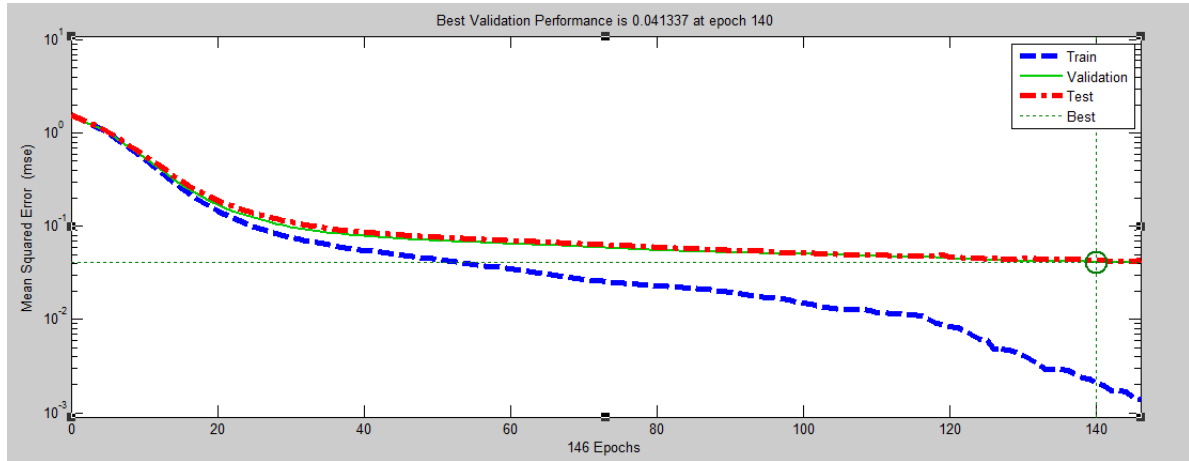


Figura 7.1: *Early stopping* para redes com 71 saídas e VBCs

Um deles é denominado conjunto de treinamento, que é utilizado na atualização dos pesos sinápticos. O outro conjunto é denominado conjunto de validação, pois à medida que a rede é treinada, realiza-se constantemente uma checagem da classificação dos dados de validação. Enquanto a checagem mostrar que o erro no reconhecimento é reduzido, o treinamento prossegue. Ao sinal de que o erro no reconhecimento começa a aumentar o treinamento é interrompido a fim de se evitar o fenômeno de *overtraining*.

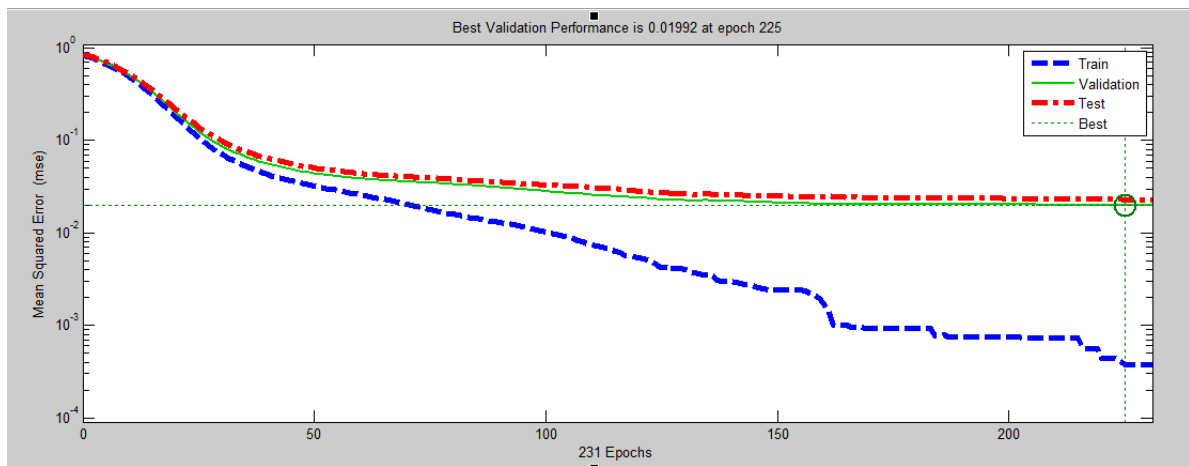


Figura 7.2: *Early stopping* para redes com 128 saídas e VNOs

Em nosso trabalho utilizamos a ferramenta *NNtool* do Matlab para a estimação do ponto de parada. Essa ferramenta separa a matriz de entrada em três subconjuntos. Um subconjunto de treinamento composto por 60% da amostra, um sub-conjunto de validação composto por 20% da amostra e um sub-conjunto de teste composto também por 20% da amostra.

A Figura 7.1 mostra a simulação para determinação do ponto de parada de treinamento para a rede treinada com VBCs. Enquanto que a Figura 7.2 corresponde à determinação do ponto de parada para redes treinadas com VNOs. Finalmente, a determinação do ponto de encerramento de treinamento para a rede treinada com VBOs é mostrada por meio da Figura 7.3.

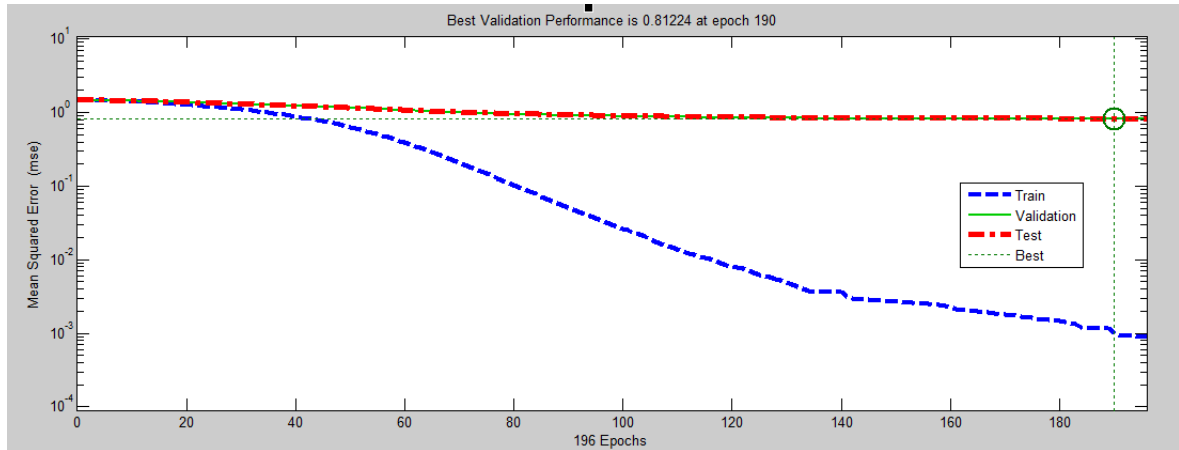


Figura 7.3: *Early stopping* para redes com 128 saídas e VBOs

Com base na determinação dos pontos de *early stopping* partiu-se então para as simulações, checando as taxas de reconhecimento para tolerâncias maiores, ou seja, para menores níveis de rigor no treinamento. Essas comparações são importantes para a análise do comportamento durante o treinamento dessas redes. Para cada classe ou indivíduo foram utilizadas 4 amostras para treinamento e 3 para teste.

A *toolbox* do Matlab pode ter seu treinamento encerrado em três circunstâncias: alcance do erro quadrático médio tolerado, alcance do número máximo de épocas ou alcance do gradiente descendente mínimo. Para que a rede tivesse seu treinamento encerrado exclusivamente pelo alcance do erro quadrático médio tolerado, atribuímos um valor bastante pequeno para o gradiente máximo tolerado e um valor bastante grande para o número máximo de épocas de treinamento. Para cada topologia os erros foram sendo reduzidos até o alcance do ponto de parada indicado pelo critério de *early stopping*. As simulações foram feitas em um computador com processador do tipo INTEL(R) CORE i5-2410TM de 2.30GHz e memória (RAM) de 4,00 GB.

7.5 Resultados experimentais

A tabela 7.1 indica os resultados das simulações da rede treinada com vetor-alvo do tipo VBC. Já a tabela 7.2 mostra os resultados provenientes das simulações com as redes treinadas com vetores-alvo do tipo VNO. Finalmente os resultados provenientes das simulações com redes treinadas com vetores-alvo do tipo VBO são mostrados na tabela 7.3. A tolerância indica o erro quadrático médio máximo estabelecido como critério de parada para o treinamento. A convergência obtida, indica o erro quadrático médio alcançado pela rede durante o seu treinamento. Os ciclos indicam a quantidade de iterações necessárias para se atingir a tolerância estabelecida no treinamento. A taxa de reconhecimento indica o percentual de acertos na fase de teste da rede. Os melhores resultados são sublinhados e destacados em negrito.

Tabela 7.1: Desempenho da rede treinada com VBCs de 71 componentes

Tolerância	Convergência obtida	Ciclos	Taxa de reconhecimento
1,00E 00	6,59E-01	1	0,94%
7,50E-01	3,66E-01	2	0,94%
5,00E-01	2,72E-01	3	1,41%
2,50E-01	2,18E-01	4	1,41%
1,00E-01	8,47E-02	5	2,35%
1,00E-02	9,90E-03	34	63,85%
1,00E-03	8,46E-04	67	83,57%
1,00E-04	1,49E-05	103	85,92%
1,00E-05	6,16E-06	131	86,85%
1,00E-06	9,63E-07	154	91,55%
1,00E-07	9,83E-08	191	<u>90,61%</u>

Além das tabelas, a comparação também é feita por meio do gráfico da Figura 7.4. Para a mesma faixa de tolerâncias, faz-se as comparações entre os desempenhos das três redes.

7.6 Discussão dos resultados experimentais

Os resultados das tabelas 7.1, 7.2 e 7.3 mostram que em todos os casos, o uso de VBOs como vetores-alvo propicia um melhor desempenho no reconhecimento de padrões. A partir da tolerância de 1,00E-01 nota-se que as taxas de reconhecimento das redes treinadas com VBOs

Tabela 7.2: Desempenho da rede treinada com VNOs de 128 componentes

Tolerância	Convergência obtida	Ciclos	Taxa de reconhecimento
1,00E 00	9,71E-01	1	1,41%
7,50E-01	5,96E-01	2	1,41%
5,00E-01	3,58E-01	3	1,88%
2,50E-01	2,23E-01	4	1,88%
1,00E-01	8,23E-02	7	2,82%
1,00E-02	9,91E-03	61	40,85%
1,00E-03	9,00E-04	131	75,59%
1,00E-04	2,96E-05	148	79,34%
1,00E-05	9,94E-06	180	85,92%
1,00E-06	9,71E-05	213	89,67%
1,00E-07	9,79E-05	239	93,43%
1,00E-08	9,56E-05	279	<u>91,55%</u>

Tabela 7.3: Desempenho da rede treinada com VBOs de 128 componentes

Tolerância	Convergência obtida	Ciclos	Taxa de reconhecimento
1,00E 00	9,79E-01	10	21,60%
7,50E-01	7,40E-01	16	60,09%
5,00E-01	4,75E-01	23	83,10%
2,50E-01	2,43E-01	32	91,08%
1,00E-01	9,80E-02	43	93,43%
1,00E-02	9,58E-03	97	95,77%
1,00E-03	9,75E-04	173	<u>95,77%</u>
1,00E-04	9,68E-05	234	<u>95,77%</u>
1,00E-05	9,70E-06	281	<u>95,77%</u>
1,00E-06	9,68E-07	327	<u>95,77%</u>
1,00E-07	9,65E-08	373	<u>95,77%</u>
1,00E-08	9,66E-09	419	<u>95,77%</u>
1,00E-09	9,72E-10	465	<u>95,77%</u>
1,00E-10	9,81E-11	511	<u>95,77%</u>

de dimensão 128 ficam acima de 90%. A partir da tolerância de 1,00E-02 é atingido o maior valor de taxa de reconhecimento da ordem de 95,77% que permanece inalterado até a tolerância de 1,00E-10. Para as redes treinadas com o uso de VBCs e VNOs as taxas são bem inferiores para as mesmas faixas de tolerâncias. Veja na Tabela 7.1 e na Tabela 7.2 que as maiores taxas alcançadas são respectivamente iguais a 91,55% e 93,43%.

Apesar de serem utilizadas apenas 4 amostras de cada indivíduo no treinamento, as

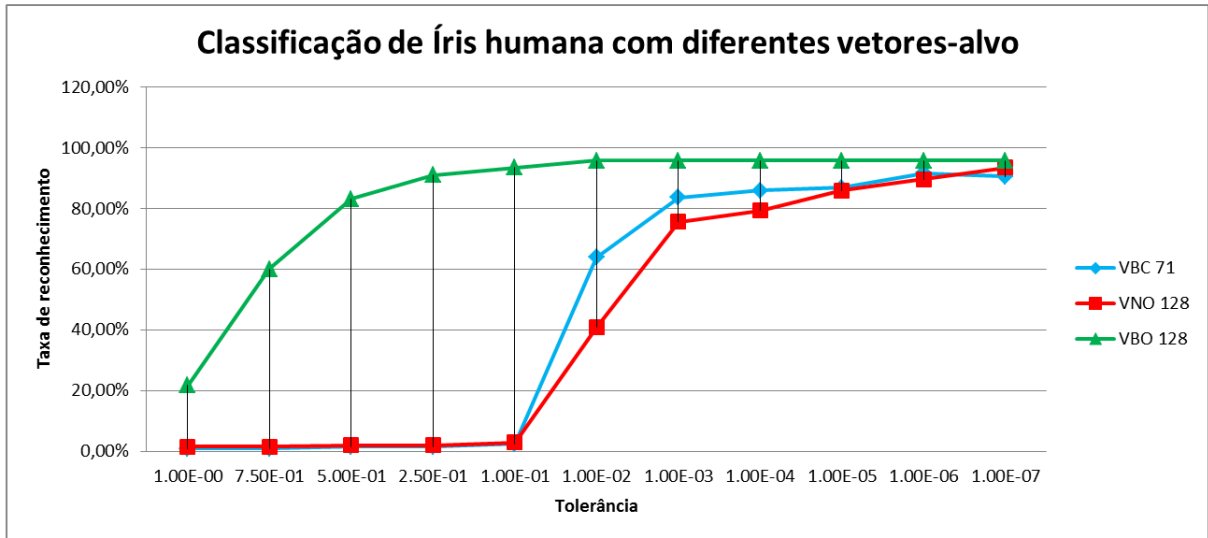


Figura 7.4: Comparação do desempenho dos vetores-alvo VBCs, VNOs e VBOs

taxas de reconhecimento foram muito superiores às taxas de reconhecimento obtidas nos experimentos com dígitos manuscritos. Isso mostra que as íris realmente constituem-se em padrões com pouca interferência de ruídos, ou melhor dizendo, são padrões bem comportados.

O mais interessante nesses experimentos é que o uso dos VBOs como vetores-alvo de redes do tipo MLP no reconhecimento de imagens de íris humana oferece a vantagem de altas taxas de reconhecimento com uma leve carga de treinamento. Verifica-se que os vetores do tipo VBCs e VNOs fazem com que o número de ciclos de treinamento seja consideravelmente elevado para que se tenha um desempenho aceitável no reconhecimento. Isso exige uma grande carga de processamento.

No caso dos VBOs, obtêm-se taxas superiores a 93% já com a tolerância de 1,00E-01. Nesse mesmo nível de tolerância, os outros tipos de vetores-alvo ainda apresentam taxas que vão de 2,35% a 2,82%. Assim, os VBOs adotados como vetores-alvo nesse trabalho, mostraram-se mais eficientes na melhoria do desempenho de MLPs, exigindo poucos ciclos no treinamento com a conseqüente redução no esforço computacional.

Capítulo 8

Conclusão e sugestões para trabalhos futuros

8.1 Conclusão

A proposta do trabalho foi analisar uma nova abordagem no treinamento de RNAs do tipo MLP com o uso de novos vetores-alvo, os VBOs. Os estudos matemáticos mostraram diferenças significativas no treinamento das redes MLPs principalmente no que diz respeito à redução de propagação de ruídos provenientes do treinamento de um padrão para o treinamento de outro padrão quando são usados esses novos vetores-alvo. Também foi verificado que o uso dos VBOs preserva características do padrão em treinamento, permitindo uma melhor separação de suas características.

Os resultados experimentais também mostram diferenças no desempenho de redes treinadas com VBOs das redes treinadas com vetores bipolares convencionais. No caso dos dígitos manuscritos, que são padrões com alto grau de degradação, verificou-se por meio do teste estatístico para proporções que houve diferença nas taxas de reconhecimento, de maneira que numericamente pode-se observar melhorias em cerca de 4% com o uso de VBOs. Nesses experimentos, foi possível observar que enquanto as redes treinadas com VBCs e VNOs atingem o ponto de over-training, as outras redes treinadas com VBOs conseguiram melhorar suas taxas

à medida que o erro tolerado é reduzido. Esses resultados foram publicados em Manzan et al. (2011).

No caso do reconhecimento de imagens de íris de seres humanos, verificaram-se altas taxas de reconhecimento para um baixo rigor no treinamento. Em alguns casos houve diferenças superiores a 40% nas taxas de reconhecimento entre redes treinadas com VBOs e VBCs. Trata-se de um resultado interessante que exige treinamentos menos extensos, que em alguns casos torna-se um entrave na utilização de RNAs fazendo com que essa ferramenta seja alvo de críticas.

Dessa maneira, o trabalho aqui apresentado, acrescenta aos anteriores já apresentados em Nomura et al. (2005), Nomura et al. (2004) e Nomura et al. (2010), novos resultados experimentais relativos a utilização de vetores-alvo do tipo VBO. Além disso, o trabalho demonstra matematicamente, evidências de que o uso desses novos vetores-alvo propicia melhorias no desempenho de RNA do tipo MLP. A utilização destes vetores não-convencionais VBOs propicia benefícios para o processo de treinamento da rede, devido ao aumento da distância euclidiana, aumento de neurônios ativos na camada de saída da MLP e redução da similaridade entre os vetores. Como a obtenção de VBOs é simples de ser realizada e não ocorrem outras alterações da técnica de redes MLP, podemos dizer que a nova abordagem é de extrema importância para a área de RNAs aplicadas a reconhecimento de padrões.

8.2 Agradecimentos

Os autores agradecem à Pró-Reitoria de Pesquisa e Pós-Graduação da Universidade Federal de Uberlândia pelo apoio na execução do projeto 72/2010 do qual o presente trabalho fez parte. O agradecimento estende-se à CAPES pelo financiamento do programa MINTER (Mestrado Interinstitucional) entre a Universidade Federal de Uberlândia e o Instituto Federal do Triângulo Mineiro.

8.3 Trabalhos futuros

As evidências matemáticas da melhora do desempenho de redes MLP são um importante passo para a teoria de RNA. Nesse trabalho, elas foram feitas de forma mais específica, com foco na atualização dos pesos sinápticos. Entretanto, muitas questões ainda precisam ser respondidas.

Um estudo matemático detalhado acerca das áreas de convergência, merece atenção. Os primeiros resultados experimentais mostram indícios bastante elucidativos do maior distanciamento entre vetores-alvo. Análises matemáticas que venham a confirmar a menor invasão entre áreas de convergência por parte de VBOs poderiam consolidar a hipótese de melhoria no desempenho de RNAs do tipo MLP.

Capítulo 9

Lista de publicações

Nesta parte apresentamos trabalhos relacionados com essa pesquisa que foram publicados ou aceitos em eventos e periódicos.

A Melhoria no desempenho de MLP com o uso de novos vetores alvo

Autores: José Ricardo G. Manzan, Keiji Yamanaka e Shigueo Nomura

Evento: X CBIC'11 (Congresso Brasileiro de Inteligência Computacional) - 2011

Local Fortaleza - CE

Mathematical evidence for target vector type influence on MLP learning improvement

Autores: José Ricardo G. Manzan, Shigueo Nomura e Keiji Yamanaka

Evento: ICAI'12 (The 2012 International Conference on Artificial Intelligence) - 2012

Local: Las Vegas, Nevada, EUA

Improving Iris Recognition through New Target Vectors in MLP Artificial Neural Networks

Autores: José Ricardo G. Manzan, Shigueo Nomura, Keiji Yamanaka, Milena Bueno Pereira Carneiro e Antônio C. Paschoarelli Veiga

Evento: ANNPR (5th Workshop on Artificial Neural Networks in Pattern Recognition) - 2012

Extensão para Springer LNAI

Local: Trento, Itália.

Referências Bibliográficas

- ABREU, N. C. L. (2003). *Extração de sinais de voz em ambientes ruidosos por decomposição em funções bases estatisticamente independentes*. Unpublished master's thesis, Universidade Federal do Maranhão.
- ACADEMY, C. (2003). www.cbsr.ia.ac.cn/irisDatabase.htm. (Acesso em fev. de 2012)
- ALVES, W. A. L., & ARAÚJO, S. A. (2005). Uma avaliação da utilização de dois diferentes modelos de redes neurais na classificação de imagens baseada em características texturais. In X. B. S. on Computer Graphics & I. Processing (Eds.), . Natal, RN.
- CARNEIRO, M. B. P., & VEIGA, A. C. P. (2005). Application of genetic algorithms to improve the reliability of an íris recognition system. *IEEE Workshop on Machine Learning for Signal Processing, Mystic*, 159-164.
- CHEN, L., & PUNG, H. K. (2008). Convergence analysis of convex incremental neural networks. *Springer Science - Ann Math Intell*, 52, 67-80.
- COOPER, L. N. (1973). A possible organization of animal memory and learning. In N. S. on Collective Propertiers of Physical Systems (Ed.), . Lundquist and S. Lundquist.
- COSTA, M. A., BRAGA, A. P., & MENEZES, B. R. (2002/2003). Improving neural networks generalization with new constructive and pruning methods. *Jornal of Intelligent & Fuzzy Systems*, 13, 75-83.
- DEVORE, J. L. (2006). *Probabilidade e estatística para engenharia e ciências* (P. T. Learning, Ed.).
- DUDA, R. O., HART, P. E., & STORK, D. G. (2001). *Pattern classification* (2nd ed.; Wiley, Ed.). New York: Wiley.

- FAUSETT, L. (1994). *Fundamentals os neural networks: Architecture, algorithms, and applications* (Prentice-Hall, Ed.). Prentice-Hall.
- FERNEDA, E. (2006). Redes neurais e sua aplicação em sistemas de recuperação de informação. *Ciência da Informação*, 35, 25-30.
- FU, K. S. (1974). *Syntactic methods in pattern recognition* (A. Press, Ed.). Academic Press.
- GALO, M. L. B. T. (2000). *Aplicação de redes neurais artificiais e sensoriamento remoto na caracterização ambiental do parque estadual morro do diabo*. Unpublished doctoral dissertation, Universidade de São Paulo.
- GONZALEZ, R. C., & WOODS, R. E. (1992). *Digital image processing* (A.-W. P. Company, Ed.). Addison-Wesley Publishing Company.
- GUARNIERI, R. A. (2006). *Emprego de redes neurais artificiais e regressão linear múltipla no refinamento das previsões de radiação solar do modelo eta*. Unpublished master's thesis, Instituto Nacional de Pesquisas Espaciais.
- GUINGO, B. C., RODRIGUES, R. J., & THOMÉ, A. C. G. (2002). Reconhecimento automático de placas de veículos automotores através de redes neurais artificiais. In . C. B. de Computação (Ed.), . Itajaí-SC.
- HANSELMAN, D., & LITTLEFIELD, B. (2003). *Matlab 6 - curso completo* (Prenti, Ed.).
- HEBB, D. O. (1949). *The organization of behavior* (J. W. . Sons, Ed.).
- KIM, D. (2005). Improving prediction performance of neural networks in pattern classification. *International Journal of Computer Mathematics*, 82, 391-399.
- KINOSHITA, S., MARQUES, P. A., & FRÈRE, A. F. (2000). Utilização da análise de componentes principais e redes neurais artificiais para a classificação de nódulos em mamogramas digitalizados. In C. B. de Engenharia Biomédica 2000 (Ed.), .
- KOHONEN, T. (1977). *Associative memory: A system-theoretical approach* (Springer-Verlag, Ed.). Springer-Verlag.
- LEE, C. M., YANG, S. S., & HO, C. L. (2006). Modified back-propagation algorithm applied to decision-feedback equalisation. *IEE Proceedings - Vision, Image, and Signal Processing*, 153, 805-809.

- 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)* (Vol. Único). Fortaleza.
- MCCULLOCH, W. S., & PITTS, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.
- NOMURA, S., YAMANAKA, K., KATAI, O., KAWAKAMI, H., & SHIOSE, T. (2004). A new mathematical approach for performance improvement on artificial neural networks. In B. B. S. on Artificial Neural Networks (Ed.), *Biannual brazilian symposium on artificial neural networks*. São Luís.
- NOMURA, S., YAMANAKA, K., KATAI, O., KAWAKAMI, H., & SHIOSE, T. (2005). Improved mlp learning via orthogonal bipolar target vectors. *Jornal of Advanced Computational Intelligence and Intelligent Informatics*, 9, 580-589.
- NOMURA, S., YAMANAKA, K., & MANZAN, J. R. G. (2010). An experimentation with improved target vectors for mlp in classifying degraded patterns. *Learning and Nonlinear Models*, 8, 240-252.
- PASSOS, M. G. (2006). *Modelos de dispositivos de microondas e ópticos através de redes neurais artificiais de alimentação direta*. Unpublished master's thesis, Centro de Tecnologia da Universidade Federal do Rio Grande do Norte.
- REPOSITORY, U. M. L. (1994). *Semeion handwritten digit do machine learning repository*. <http://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit>. Retrieved from <http://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit>
- ROBERTS, J. E. (1989). Influence of target-vector code selection on the performance of neural-network word recognizer. In I. Xplore (Ed.), *Neural networks. ijcnn., international joint conference*. MITRE Corp., Bedford, MA, USA.
- SANTOS, A. M., SEIXAS, J. M., PEREIRA, B. B., & MEDRONHO, R. A. (2005). Usando redes neurais artificiais e regressão logística na predição da hepatite a. *Revista Brasileira e Epidemiologia*, 8(2), 117-126.
- SILVA, I. N., FLAUZINO, D. H., & ANDRADE, R. (2010). *Redes neurais artificiais para*

engenharia e ciências aplicadas (Artliber, Ed.). Artliber.

WANG, X., CHANG, C.-C., & DU, F. (2004). Achieving a more robust neural network model for control of a mr damper by signal sensitivity analysis. In *Machine learning: ECML 2004, 15th european conference on machine learning, pisa, italy, september 20-24, 2004, proceedings* (pp. 39–50). Springer.