



UNIVERSIDADE ESTADUAL DO MARANHÃO

**Programa de Pós-Graduação em Engenharia de
Computação e Sistemas**

Aline Porfiro Teixeira

Múltiplas Redes Neurais Utilizando Redes MLP e RBF

São Luís

2017

Universidade Estadual do Maranhão
Programa de Pós-Graduação em Engenharia de Computação e Sistemas
Mestrado em Engenharia de Computação e Sistemas

ALINE PORFIRO TEIXEIRA

MÚLTIPLAS REDES NEURAIIS UTILIZANDO REDES MLP E RBF

São Luís - MA
Janeiro de 2017

ALINE PORFIRO TEIXEIRA

MÚLTIPLAS REDES NEURAIS UTILIZANDO REDES MLP E RBF

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Computação e Sistemas da UEMA, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação e Sistemas.

Aprovada em / /

Composição da Banca Examinadora

Prof. Dr. Areolino de Almeida Neto
Orientador (UFMA)

Prof. Dr. Lúcio Flávio de Albuquerque Campos
Membro (UEMA)

Prof. Dr. Alexandre César Muniz de Oliveira
Membro (UFMA)

São Luís - MA
Janeiro de 2017

Teixeira, Aline Porfiro.

Múltiplas Redes Neurais Utilizando Redes MLP e RBF / Aline Porfiro Teixeira. – São Luís, 2017.

70 f.

Impresso por computador (fotocópia).

Orientador: Areolino de Almeida Neto.

Dissertação (Mestrado) – Universidade Estadual do Maranhão, Programa de Pós-Graduação em Engenharia de Computação e Sistemas, 2017.

1. Redes neurais artificiais. 2. Múltiplas Redes Neurais. 3. Reconhecimento de Padrões. I. Título.

AGRADECIMENTOS

Ao meu orientador, Prof. Areolino de Almeida Neto, por todo apoio e ensinamentos passados para realização deste trabalho.

Ao Instituto Federal do Maranhão, instituição a qual faço parte, pela liberação e apoio financeiro.

À Universidade Estadual do Maranhão e a todos os professores do Programa de Pós-Graduação em Engenharia da Computação, pela oportunidade de aprendizado.

À Universidade Federal do Maranhão, por conceder o espaço em que este trabalho foi realizado.

À Fundação de Amparo à Pesquisa do Maranhão, pelo auxílio financeiro concedido na publicação de artigo realizado a partir deste trabalho.

Aos meus familiares, em especial ao meu filho Benjamin Porfiro Ribeiro, minha grande motivação.

“O autoconhecimento é o começo da sabedoria,
em cuja tranquilidade e silêncio encontra o imensurável.”

Jiddu Krishnamurti

RESUMO

Este trabalho descreve uma metodologia que promove a colaboração e autocoordenação entre redes do tipo *Multilayer Perceptron* e *Radial Basis Function* através da técnica de Múltiplas Redes Neurais, que dispensa o uso de coordenador. Múltiplas Redes Neurais são utilizadas com o objetivo de facilitar o aprendizado e a solução de problemas muito complexos. Normalmente, para que o aprendizado de uma rede não prejudique o aprendizado das demais, utiliza-se um coordenador, que tem por função evitar os conflitos entre as redes. No entanto, foi desenvolvida uma técnica que dispensa o uso do elemento coordenador, onde as próprias redes autocoordenam. Visando ampliar a abrangência desta técnica, este trabalho propõe a colaboração entre redes de tipos diferentes, fazendo com que suas vantagens colaborem de modo a solucionar eficientemente problemas diversos. Esta metodologia é comparada a outras a partir de testes da solução de detecção de câncer de mama e de distúrbios na coluna vertebral.

Palavras-Chave: Redes Neurais Artificiais, Múltiplas Redes Neurais, Reconhecimento de Padrões.

ABSTRACT

This work describes a methodology that promotes a collaboration and selfcoordination between Multilayer Perceptron and Radial Base Function networks through the Multiple Neural Networks technique, which doesn't require the use of a coordinator. Multiple Neural Networks are used for the purpose of facilitating learning and solving very complex problems. Normally, in order to learn an unimpaired network or learn from others, use a coordinator, whose function is to avoid conflicts between networks. However, a technique has been developed that dispenses with the use of the element coordinator, where as self-coordinating networks. In order to extend the scope of this technique, this work proposes a collaboration between networks of different types, making their advantages collaborate in order to efficiently solve diverse problems. This methodology is compared to other solutions for detection of breast cancer and disorders of the spine.

Keywords: Artificial Neural Networks, Multiple Neural Networks, Pattern Recognition

SUMÁRIO

AGRADECIMENTOS.....	III
EPÍGRAFE.....	IV
RESUMO.....	V
ABSTRACT.....	VI
LISTA DE FIGURAS.....	IX
LISTA DE TABELAS.....	X
LISTA DE ABREVIATURAS E SIGLAS.....	XI
CAPÍTULO 1 - INTRODUÇÃO	14
1.1. OBJETIVOS.....	14
1.2. MOTIVAÇÃO	15
1.3. TRABALHOS RELACIONADOS.....	16
1.4. ORGANIZAÇÃO DA DISSERTAÇÃO	20
CAPÍTULO 2 - REDES NEURAS ARTIFICIAIS	21
2.1. DEFINIÇÕES.....	21
2.2. ESTRUTURA	22
2.3. APRENDIZADO	25
2.4. TIPOS DE RNA	27
2.4.1. Redes MLP	28
2.4.2. Redes RBF.....	30
2.5. MÚLTIPLAS REDES NEURAS ARTIFICIAIS.....	32
CAPÍTULO 3 - COLABORAÇÃO ENTRE REDES MLP E RBF.....	37
3.1. DESCRIÇÃO DA METODOLOGIA DE TREINAMENTO.....	37
3.2. METODOLOGIA DE AVALIAÇÃO DE RESULTADOS	38
3.3. RESULTADOS	39
CAPÍTULO 4 - CONCLUSÃO	47
4.1. CONSIDERAÇÕES FINAIS.....	47

4.2. CONTRIBUIÇÕES DESTE TRABALHO.....	48
4.3. TRABALHOS FUTUROS.....	48
REFERÊNCIAS BIBLIOGRÁFICAS	50
APÊNDICE A – ARTIGO VERSÃO ORIGINAL.....	53
APÊNDICE B – ARTIGO TRADUZIDO	62

LISTA DE FIGURAS

Figura 2.1: Modelo de neurônio MCP.....	22
Figura 2.2: Estrutura da RNA do tipo MLP	23
Figura 2.3: Topologia <i>feedforward</i>	24
Figura 2.4: Topologia Estritamente <i>feedforward</i>	24
Figura 2.5: Topologia <i>feedback</i>	24
Figura 2.6: RNA com uso de <i>bias</i>	25
Figura 2.7: Aprendizado supervisionado.....	26
Figura 2.8: Aprendizado não supervisionado.....	26
Figura 2.9: Estrutura da RNA do tipo RBF.....	31
Figura 2.10: Classificação com redes MLP(esquerda) e redes RBF(direita)	32
Figura 2.11: Combinação do tipo ensemble.....	33
Figura 2.12: Combinação do tipo modular.....	33
Figura 2.13: Combinação do tipo modular.....	34
Figura 2.14: Arranjo de MRNA com duas redes.....	35
Figura 3.1: Gráfico do Erro Quadrático Médio.....	40
Figura 3.2: Curva ROC (Melhor Caso) – Base de dados 1	43
Figura 3.3: Curva ROC (Pior Caso)– Base de dados 1	44
Figura 3.4: Curva ROC (Intermediário)– Base de dados 1	44
Figura 3.5: Curva ROC – Base de dados 2.....	46

LISTA DE TABELAS

Tabela 3.1: Taxas de acerto – Base de dados 1	40
Tabela 3.2: Matriz de Confusão – Base de dados 1	42
Tabela 3.3: Tabela de dados para curva ROC – Base de dados 1	42
Tabela 3.4: Taxas de acerto – Base de dados 2	45
Tabela 3.5: Tabela de dados para curva ROC - Base de dados 2.....	46

LISTA DE ABREVIATURAS E SIGLAS

BP	<i>Backpropagation</i>
IA	<i>Inteligência Artificial</i>
MLP	<i>Multilayer Perceptron</i>
MRN	<i>Múltiplas Redes Neurais</i>
RBF	<i>Radial Basis Function</i>
RNA	<i>Redes Neurais Artificiais</i>
TP	<i>True Positive</i>
TN	<i>True Negative</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>
FNN	<i>Fuzzy Neural Network</i>
SOM	<i>Self Organizing Maps</i>

CAPÍTULO 1 - INTRODUÇÃO

Sendo um mecanismo para classificação, o Reconhecimento de Padrões tem sido cada vez mais útil e necessário em sistemas computacionais das mais diversas áreas. Objetivando a classificação de objetos dentro de um número de categorias ou classes, este mecanismo computacional possibilita a criação de uma enorme gama de aplicações para uso científico e tecnológico, como reconhecimento de fala, classificação de documentos, reconhecimento de escrita e reconhecimento de faces, entre outros.

Entre os mais variados métodos utilizados para realizar o Reconhecimento de Padrões, têm-se as Redes Neurais Artificiais (RNA). A partir de um processo de treinamento, uma RNA é capaz de acumular conhecimento e generalizá-lo para outros casos ainda não vistos pela rede durante o treinamento. Com isso, aprendendo a identificar características dos padrões apresentados no treinamento, a RNA poderá reconhecer outros objetos com as mesmas características.

Porém existem situações em que encontrar a solução utilizando apenas uma rede neural torna-se uma tarefa muito difícil. Funcionando como um aproximador de funções, nestes casos, seriam necessários muitos ajustes na topologia, nos parâmetros e no treinamento da RNA utilizada para que esta consiga aproximar-se da função da solução durante o aprendizado.

Em situações como esta, alternativamente é possível utilizar mais de uma rede neural para facilitar o aprendizado, que trabalhando em conjunto passam a ser chamadas de Múltiplas Redes Neurais.

Desta forma, este trabalho pretende apresentar um modelo inovador de múltiplas redes neurais combinando redes de tipos diferentes: *Multilayer Perceptron* (MLP) e *Radial Basis Function* (RBF). Além disso, outro ponto importante deste trabalho é que a combinação dá-se sem o uso de um coordenador, mas mesmo assim evitando conflitos entre as redes durante o aprendizado.

1.1. OBJETIVOS

Este trabalho tem como objetivo realizar uma melhoria no reconhecimento de padrões utilizando Múltiplas Redes Neurais Artificiais de tipos diferentes (MLP e RBF) de forma autocoordenada.

Mais detalhadamente, é proposto um modelo de Múltiplas Redes Neurais, combinando redes de tipos diferentes, de forma que haja colaboração entre elas e evitando conflitos.

O objetivo principal depende da concretização dos seguintes objetivos específicos:

- Realizar a união de redes do tipo MLP e RBF;
- Possibilitar o aprendizado de modo autocoordenado, evitando conflitos;
- Garantir a melhoria quanto à generalização através desta metodologia.

1.2. MOTIVAÇÃO

No contexto de Redes Neurais Artificiais, existem muitos casos em que realizar o aprendizado, utilizando uma única rede, torna-se uma tarefa muito difícil. Isto ocorre porque os parâmetros da rede, assim como sua topologia, devem ser definidos pelo projetista, não havendo, porém, uma metodologia que sirva para configurar a rede de forma adequada. Desta forma, é comum que sejam necessárias várias tentativas de treinamento, visando ajustes na rede para que se possa chegar a uma configuração de rede que resolva o problema.

Como alternativa para facilitar o processo de aprendizagem nestes casos, tem-se a metodologia de Múltiplas Redes Neurais. Esta metodologia propicia o aprendizado de soluções em sistemas complexos, onde haveria grande dificuldade de encontrar a RNA que conseguiria aprender a solução, uma vez que o trabalho do aprendizado seria compartilhado entre vários elementos, diminuindo, assim, o esforço de cada rede.

Contudo, nesta metodologia, normalmente se utiliza um coordenador com a função de evitar que uma rede prejudique o aprendizado de outra. Deste modo, o trabalho do projetista torna-se sobrecarregado com o projeto, implementação e teste do elemento coordenador, que além de não garantir, pode até mesmo prejudicar a harmonia entre as redes usadas.

Tendo essa dificuldade como premissa, foi desenvolvida por de Almeida Neto [3] uma técnica que dispensa o uso do coordenador, fazendo com que redes do tipo MLP coordenem-se. Porém, no trabalho desenvolvido por este pesquisador, somente redes do tipo MLP foram usadas e testadas, impedindo assim o aproveitamento de vantagens de redes de outros tipos. Desta forma, visando aumentar a abrangência desta técnica, este trabalho combina redes do tipo *Multilayer Perceptron* (MLP) e *Radial Basis Function* (RBF)

utilizando a técnica citada acima, que dispensa o uso de coordenador e verifica sua capacidade de aprendizado quando se trata de redes de diferentes tipos.

1.3. TRABALHOS RELACIONADOS

Nesta seção, são descritos alguns trabalhos relacionados, que são trabalhos que envolvem Múltiplas Redes Neurais para solução de problemas e também trabalhos em que houve o desenvolvimento de metodologias que promoveram algum tipo de colaboração entre diferentes técnicas computacionais de Inteligência Artificial (IA).

Dentre os trabalhos que visam à colaboração entre redes neurais, destaca-se a técnica de autocoordenação de Múltiplas Redes Neurais Artificiais, desenvolvida por de Almeida Neto [3], a qual é utilizada quando uma RNA estagnou seu aprendizado ou quando apresenta baixa redução no valor do erro, possibilitando o seu aprendizado não ser descartado.

Nesse trabalho, o aprendizado da rede que ainda não obteve o valor de erro esperado é utilizado no treinamento de uma nova rede neural. Desta forma, obtém-se a vantagem de não desperdiçar o aprendizado já conseguido e também faz com que as duas ou mais redes colaborem. Esta colaboração dá-se sem o uso de um coordenador, diferentemente de outras formas de colaboração entre redes, onde há a preocupação de implementação do elemento coordenador. Pode-se então afirmar que a técnica desenvolvida por de Almeida Neto [3] traz um grande ganho ao projetista, uma vez que possibilita a colaboração sem a necessidade de implementação de um elemento a mais.

No trabalho de Ahmad *et al.*[28], é proposto um método para predição em tempo real do índice de qualidade da água. Para tanto, utilizou-se inicialmente uma única rede neural *feedforward* e os resultados foram comparados com a utilização de Múltiplas Redes Neurais do tipo MLP e com algoritmo de treinamento Levenberg-Marquardt. Foram utilizados métodos de combinação de seleção direta e eliminação reversa para combinar as saídas das RNAs, que pode ser visto como um coordenador. O que se verificou foi um melhor desempenho ao se utilizar Múltiplas Redes Neurais ao se comparar com os resultados quando se utiliza uma única rede neural.

No estudo realizado por D'Andrea *et al.*[29], buscou-se prever a composição de ligas de bronze desconhecidas através da técnica de Múltiplas Redes Neurais. O conjunto de redes nesse trabalho foi composto por duas redes do tipo RBF. Como forma de combinar as saídas das redes, utilizou-se a média das saídas das mesmas. Através da comparação dos resultados

do conjunto de redes com uma única rede do tipo RBF, pôde-se concluir que a utilização da técnica de Múltiplas Redes Neurais possibilitou melhores resultados.

Outro trabalho que aborda sobre a técnica de Múltiplas Redes Neurais é o trabalho de Li e Liu [30]. A metodologia proposta é utilizada para conduzir a previsão de séries temporais. O conjunto de redes neurais foi composto por redes do tipo MLP que utilizaram o algoritmo *Backpropagation* para treinamento e outra rede neural, tendo papel de um coordenador, foi utilizada para dar a saída final, tendo como entrada os valores gerados no conjunto de RNAs. Os resultados foram comparados a métodos tradicionais e foi verificado que o método proposto melhorou a precisão da previsão de séries temporais.

No trabalho de Wang *et al.* [31], é proposta uma metodologia de treinamento adaptativo baseado na técnica de Múltiplas Redes Neurais para a estimativa *online* do valor do pH de minérios em um processo industrial. Inicialmente, uma rede neural do tipo RBF foi utilizada para classificar os dados em 3 classes possíveis e para cada classe, uma nova rede RBF é construída. O valor de pH estimado final é a saída a qual a amostra pertence. Foi proposta também uma abordagem adaptativa para que a rede seja atualizada às mudanças no ambiente a qual está inserida. A metodologia proposta apresentou bons resultados quando comparado ao aprendizado de uma única rede RBF.

Kung *et al.* [32] propõem uma metodologia baseada na técnica de Múltiplas Redes Neurais do tipo *ensemble* para gerar previsões de rendimento agrícola. No modelo proposto vários modelos de redes são criados, onde cada uma possui uma configuração quanto ao número de camadas ocultas e quantidade de neurônios. As redes são do tipo MLP e utilizam o algoritmo *Backpropagation* para treinamento. O modelo que não atingir a taxa de precisão é descartado. Para dar uma saída final do conjunto, foi utilizada a média ponderada de todos os modelos de redes restantes. Obteve-se uma menor taxa de erro e uma maior precisão ao se comparar essa metodologia ao BP tradicional.

Xie *et al.* [33] desenvolvem uma metodologia baseada em redes *ensemble* para a classificação de tumores melanocíticos como benignos ou malignos. No modelo proposto, redes do tipo MLP são combinadas com redes neurais *fuzzy*, que é definido como um sistema de inferência que combina lógica *fuzzy* com uma RNA. Para se combinar as saídas dos modelos, são utilizados dois métodos, o de votação e a média. Ao se comparar a metodologia proposta com diversos métodos tradicionais, concluiu-se que o método proposto traz ganhos quanto à classificação de tumores melanocíticos.

Outro trabalho que promove colaboração de técnicas com as redes neurais pertence à Zanchettin *et al* [25], que propõe o desenvolvimento de um sistema híbrido denominado

GaTSa (*Genetic Algorithm, Tabu Search e Simulating Annealing*), que melhora uma RNA construtiva em estrutura e desempenho, integrando o algoritmo Backpropagation, o Algoritmo Genético para a recomendação de adição de novos neurônios, fuga de mínimo local através da utilização da técnica de *Simulating Annealing* e Busca Tabu para obter uma convergência acelerada a partir da validação de um conjunto de soluções para cada iteração.

Nascimento e Coelho [23] investigaram a eficácia da fusão de algoritmos simples e complexos, instáveis e estáveis. Foram selecionados então dez algoritmos diferentes de aprendizagem: o algoritmo *Naïve Bayes* simples, baseado em estatística bayesiana; redes neurais RBF e máquinas de vetores-suporte treinadas via algoritmo SMO, que se baseiam em funções numéricas não-lineares; algoritmos J48 e REP Tree, baseados em árvores de decisão; algoritmo IBk, baseado nos conceitos de vizinhança e aprendizado local; e algoritmos *Decision Stump*, OneR, PART (9) e *Decision Table*, baseados em regras. Além disto, para a questão da diversidade de resultados, utilizou-se o método *bagging* para a combinação de entradas via reamostragem. Para seleção automática dos diferentes tipos de algoritmos, de acordo com as nuances do problema, foi utilizado um algoritmo genético customizado.

No trabalho de Woźniak *et al.* [15], faz-se um estudo aprofundado sobre Sistemas Inteligentes Híbridos, com um apanhado histórico sobre sua evolução. Dentro deste conjunto o foco do trabalho está nos Sistemas Multi-Classificadores (MCS), destacando a melhoria de resultados quando comparados a um único classificador.

No trabalho de Adawy *et al.* [1], a técnica *Single Offspring* (SOFT), que consiste na utilização de algoritmo genético no treinamento de redes neurais, é mesclada com o algoritmo *Backpropagation*. Nesta metodologia híbrida, utiliza-se algoritmo genético para determinar pesos iniciais adequados para uma determinada rede e então estes são refinados com o algoritmo *Backpropagation*. Com esta concatenação de algoritmos, consegue-se unir a capacidade de um bom início de pesos com algoritmo genético e um refinamento dos pesos com o *Backpropagation*.

Khotanzad e Sadek [16] utilizam redes MLP (*Multilayer Perceptron*) e FNN (*Fuzzy Neural Network*) como integrantes do sistema de múltiplas redes. Investiga-se vários métodos de combinação dessas duas saídas. Tendo foco principal a determinação que as redes em conjunto produzem melhor resultado quando comparado com o resultado de uma única rede e também o melhor método de combinação dessas saídas, sendo verificado que esquemas não-lineares de combinação produzem melhores resultados.

No trabalho de Dhlamini e Marwala [17] são analisadas características como velocidade, a estabilidade e a precisão de Múltiplas Redes com redes individuais. Quatro tipos

diferentes de rede neural artificial *feedforward* foram testados individualmente e coletivamente, redes MLP, RBF (Função de Base Radial), SVM (Suporte Máquinas de Vetores) e BNN (Rede Bayesiana). As redes com melhores desempenhos foram selecionadas para compor o conjunto de redes, que tiveram suas saídas combinadas pela aplicação da média. O modelo proposto apresentou uma taxa de acerto de 78% através da metodologia proposta.

No trabalho de Abdella e Marwala [18], combinam-se redes do tipo MLP e RBF para aproximar dados ausentes em um banco de dados. Para combinar as saídas e determinar o término do aprendizado, foi utilizado um algoritmo genético. Utiliza-se o erro padrão e o coeficiente de correlação para avaliar a qualidade dos resultados obtidos. Foi verificada a melhora dos resultados quando comparado o método proposto com redes individuais do tipo MLP e RBF.

Woods *et al.* [19], utilizaram como componentes redes neurais, classificadores do tipo *k-nearest neighbour*, árvores de decisão e classificadores quadráticos bayesianos e, para cada componente, utilizaram a estimativa de sua precisão local no espaço de atributos para escolher qual desses componentes seria responsável pela resposta para uma nova entrada. Desta forma, foi proposto um algoritmo para seleção dos componentes do sistema inteligente, onde foi destacado a dificuldade de otimizar cada classificador individual com a seleção realizada.

Wang *et al.* [20], utilizaram como componentes redes neurais e árvores de decisão, chegando à conclusão de que os melhores desempenhos são obtidos com um número de redes neurais maior que o de árvores de decisão, mas com pelo menos uma árvore no ensemble. Langdon *et al.* [21] utilizaram redes neurais e árvores de decisão, mas aplicaram Programação Genética para evoluir uma regra de combinação dos indivíduos.

Soares *et al.* [22] utilizaram como componentes de um sistema inteligente redes neurais do tipo MLP, redes neurais do tipo *radial basis function* (RBF), classificadores Naïve Bayes, máquinas de vetores suporte (SVM) e classificadores baseados em aprendizado de regras proposicionais, e propuseram duas técnicas de seleção de componentes (baseadas em algoritmo de agrupamento e *knearest neighbours*) que buscam não apenas reduzir o erro do *ensemble*, mas também aumentar a diversidade de seus componentes.

Convém ressaltar que os trabalhos relacionados citados, quando realizaram algum tipo de colaboração entre redes de tipos diferentes, foi necessária a utilização de algum mecanismo de coordenação entre as saídas, de modo que nenhum trabalho investigou a autocoordenação entre técnicas diferentes de aprendizado.

1.4. ORGANIZAÇÃO DA DISSERTAÇÃO

Em sequência a este capítulo, têm-se no Capítulo 2 os principais fundamentos teóricos para o desenvolvimento deste trabalho, a respeito das redes neurais, tais como as principais definições, a rede do tipo MLP e RBF e sobre Múltiplas Redes Neurais.

No Capítulo 3, são apresentadas as etapas do desenvolvimento da metodologia proposta, onde se combinam redes do tipo MLP e RBF através da técnica de Múltiplas Redes Neurais de modo autocoordenado. Vale ressaltar que os resultados obtidos com esse trabalho foram publicados no periódico JCIT – *Journal of Convergence Information Technology*, volume 11, página 1 a 9 (Qualis B1), cuja versão original em inglês, assim como uma versão em português em tradução livre, está em apêndice.

O Capítulo 4 mostra quais as conclusões relevantes a partir da análise dos resultados obtidos, tal como as contribuições científicas. Além disso, são exibidas as vertentes que podem ser exploradas a partir deste estudo, como sugestões de trabalhos futuros.

CAPÍTULO 2 - REDES NEURAIS ARTIFICIAIS

Quando se fala de construção de soluções computacionais, pode-se afirmar que existem duas principais abordagens possíveis, a abordagem algorítmica e a abordagem baseada em inteligência artificial (IA). A primeira abordagem se baseia nos paradigmas tradicionais de programação e são utilizadas para problemas com níveis de abrangência representáveis.

No entanto, existem problemas que possuem modelo lógico-matemático de desconhecido ou de difícil representação. Criar soluções para estes problemas através da abordagem algorítmica se torna uma tarefa complexa. Neste contexto, surgem metodologias baseadas em inteligência artificial, que apresentam a capacidade de aprendizado e deste modo possibilitam a construção de soluções computacionais para problemas em que a abordagem algorítmica não é adequada. Dentre as metodologias baseadas em IA, podem-se citar as Redes Neurais Artificiais (RNA).

2.1. DEFINIÇÕES

Uma rede neural artificial pode ser definida como um sistema processador intenso e paralelamente distribuído, cujas unidades de processamento são bem simples, que têm a propensão natural para armazenar conhecimento e torná-lo disponível para o uso [1].

Desta forma, uma rede neural assemelha-se ao cérebro humano ao adquirir conhecimento através de um processo de aprendizagem e por possuir forças de conexões entre neurônios, conhecidas como pesos sinápticos, utilizadas para armazenar o conhecimento adquirido.

Uma RNA é formada então, basicamente, por neurônios artificiais e conexões entre os mesmos. Sendo a unidade básica de processamento de uma rede, um neurônio artificial tem seu funcionamento inspirado no neurônio humano, sendo um elemento com baixo processamento e alta conectividade.

Proposto por McCulloch e Pitts, o modelo do neurônio artificial foi chamado de MCP, por representar as iniciais de seus criadores. Neste modelo, podem-se identificar quatro elementos básicos de um neurônio artificial: conjunto de sinapses ou elos de conexão, combinador dos sinais de entrada, uma função de ativação e um valor de saída. A Figura 2.1 representa este modelo de neurônio, que forma a base para o projeto de redes neurais artificiais.

Figura 2.1: Modelo de neurônio MCP



Fonte: de Almeida Neto [3].

Inicialmente, o neurônio recebe um conjunto de entradas, análogo às conexões sinápticas do neurônio biológico. No neurônio artificial, essas entradas são caracterizadas por pesos, de modo que cada entrada seja valor numérico, podendo ser este valor negativo ou positivo.

Os sinais de entrada são então combinados, através de um somatório, de modo a gerar um valor denominado *net*. Após se combinar as entradas, é aplicada a chamada função de ativação [1], com o objetivo de restringir a amplitude de saída de um neurônio. Podem-se encontrar referências à função de ativação como função restritiva, já que restringe (limita) o intervalo de saída a um valor finito.

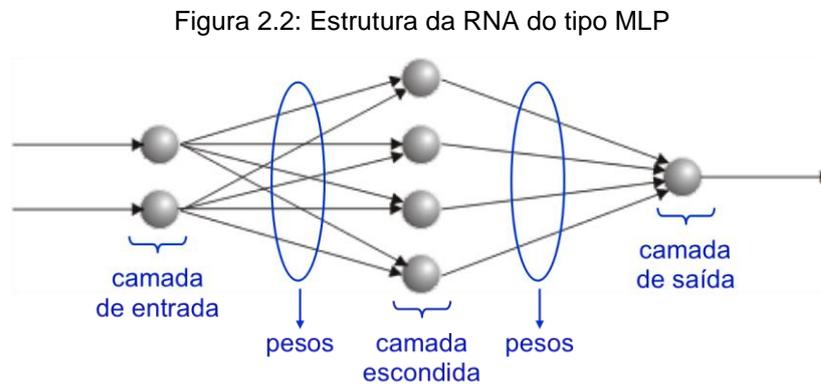
Quanto aos tipos de função de ativação, alguns exemplos mais utilizados são citados a seguir:

- Função Linear: $out = k.net$, onde k é uma constante;
- Função Sigmoide: $out = (1 + e^{-net})^{-1}$;
- Função Tangente Hiperbólica: $out = (1 - e^{-net})(1 + e^{-net})^{-1}$;
- Função Gaussiana: $out = e^{-\frac{net^2}{2\sigma^2}}$.

2.2. ESTRUTURA

Quanto à estrutura de uma RNA, pode-se agrupar os neurônios de uma rede para a formação de camadas, sendo que sempre deverá haver pelo menos uma camada de entrada e uma camada de saída, podendo também haver uma ou mais camadas intermediária, denominadas também de camada escondida ou oculta. A maneira que o projetista escolhe para organizar os neurônios de uma rede neural está intimamente ligada com o algoritmo de

aprendizagem usado para treinar a rede. A Fig. 2.2 representa a estrutura básica de uma RNA com uma camada oculta.



Fonte: de Almeida Neto [3].

Na figura acima, tem-se a camada de entrada com dois neurônios, representados pelo círculo. Cada uma das setas entre os neurônios representa os pesos que, como já mencionado, correspondem às conexões sinápticas do neurônio biológico. O valor numérico dos pesos é alterado gradativamente no processo de treinamento até que seja garantido o aprendizado da RNA.

Existem variações quanto à topologia da rede, determinando a forma como se dá a conexão entre os neurônios e seu sentido. A Figura 2.3 representa a topologia *feedforward*, onde as ligações entre os neurônios são feitas de modo em que o fluxo de dados seja realizado somente em um sentido.

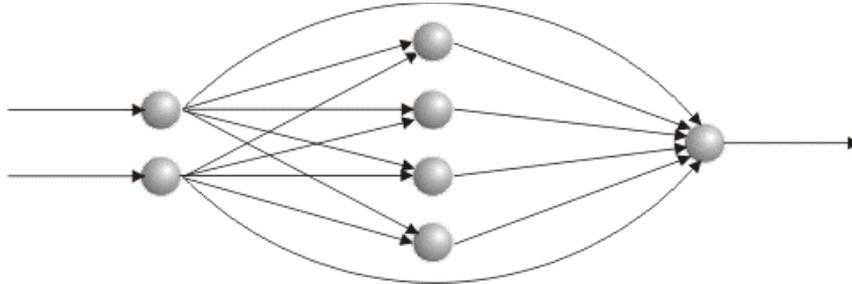
Existe uma variação da topologia acima mencionada, onde as ligações são realizadas também em apenas um sentido e, além disso, estritamente em camadas vizinhas. Este tipo de topologia é denominado estritamente *feedforward*, como pode ser visualizado na Figura 2.4.

Para redes em que o fluxo de dados ocorre não apenas em um único sentido, dá-se o nome de topologia *feedback*. A representação desta topologia pode ser visualizada na Figura 2.5.

Um elemento chamado *bias* pode ser adicionado à topologia de uma rede neural. O *bias* tem o efeito de aumentar ou diminuir a entrada líquida de uma função de ativação, dependendo se ele é positivo ou negativo, respectivamente [1]. Deste modo, este elemento funciona como um parâmetro externo que se liga a um ou a um conjunto de neurônios. Possuindo sempre o valor absoluto de um como saída, a representação desse elemento se assemelha a de um neurônio artificial, como pode ser visualizado na Figura 2.6.

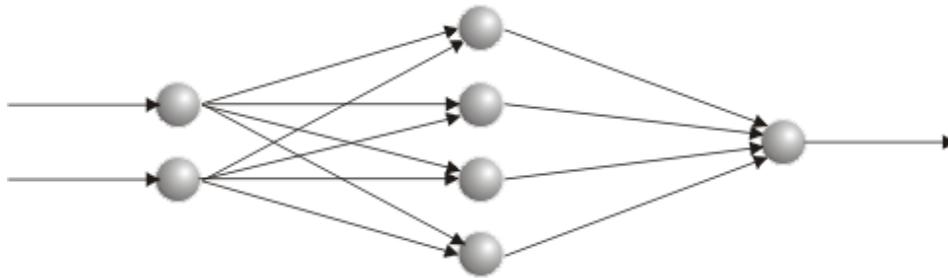
Como já mencionado, a escolha feita pelo projetista de como organizar os neurônios em uma rede está ligada ao algoritmo de aprendizagem utilizado e este por sua vez está ligado ao problema que se deseja resolver.

Figura 2.3: Topologia *feedforward*



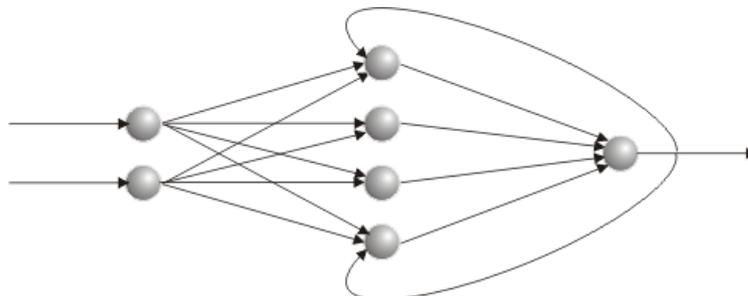
Fonte: de Almeida Neto [3].

Figura 2.4: Topologia Estritamente *feedforward*

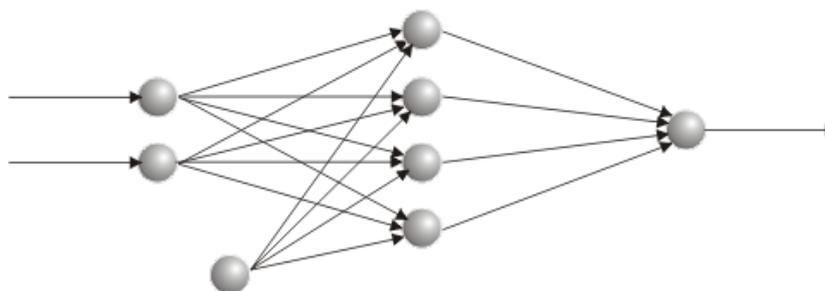


Fonte: de Almeida Neto [3].

Figura 2.5: Topologia *feedback*



Fonte: de Almeida Neto [3].

Figura 2.6: RNA com uso de *bias*

Fonte: de Almeida Neto [3].

2.3. APRENDIZADO

Redes Neurais Artificiais possuem a capacidade de aprender por exemplos e fazer interpolações e extrapolações do que aprendem [2]. Para que isto seja possível, um conjunto de procedimentos é executado para que os valores de conexões entre os neurônios sejam adaptados durante o processo denominado treinamento da rede neural.

Este processo, portanto, define o próprio aprendizado da rede. Conforme também definido por Haykin [1], aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida.

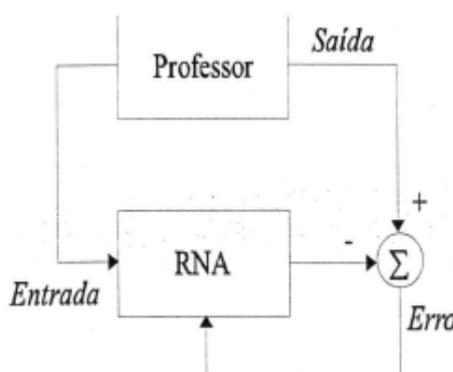
A maneira como este processo é realizado define os tipos possíveis de aprendizado, de acordo com a disponibilidade dos dados. Desta forma, os diversos métodos de treinamento podem ser agrupados em três paradigmas de tipos de aprendizado: supervisionado, não supervisionado e semisupervisionado.

Sendo o método mais utilizado em treinamento de RNAs, o aprendizado supervisionado ocorre quando dados de entrada e saída desejadas são fornecidas para a rede por um supervisor (professor) externo. Deste modo, a rede é ajustada durante o processo de treinamento de modo a encontrar uma ligação entre os pares de entrada e saída fornecidos.

A Figura 2.7 ilustra o mecanismo de aprendizado supervisionado. Neste mecanismo o professor indica explicitamente um comportamento bom ou ruim para a rede, com objetivo de direcionar o processo de treinamento. A rede tem então sua saída corrente comparada com a saída desejada, recebendo informações do supervisor sobre o erro da resposta atual. Este cálculo é feito para cada padrão de entrada submetido à rede, ajustando-se então os pesos das conexões para a minimização do erro. Assim, a minimização do erro é incremental, à medida

que pequenos ajustes são feitos nos pesos a cada etapa de treinamento, de tal forma que caminhem para uma solução.

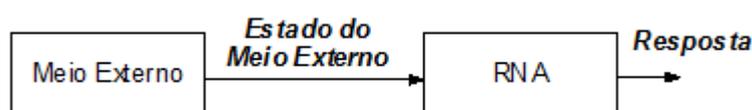
Figura 2.7: Aprendizado supervisionado



Fonte: Braga *et al* [2].

No aprendizado não supervisionado, como o próprio nome sugere, ocorre quando não é utilizado um professor para acompanhar o processo de treinamento.

Figura 2.8: Aprendizado não supervisionado



Fonte: Braga *et al* [2].

A Figura 2.8 ilustra o funcionamento do aprendizado não supervisionado. Para este tipo de aprendizado, somente os padrões de entrada estão disponíveis para a rede, ao contrário do aprendizado supervisionado, onde há pares de entrada e saída durante o treinamento. Desta forma, o conjunto de dados de entrada apresentado à rede neural será organizado de modo a formar relações e será desenvolvida a habilidade de formar representações internas para codificar características de entrada e criar novas classes ou grupos automaticamente.

Existe também um terceiro tipo de aprendizado, menos comum, denominado semisupervisionado. Trata-se de uma abordagem recente onde, durante o treinamento, há um conjunto de dados com pares de entrada e saída como no treinamento supervisionado e também um segundo conjunto de dados em que há apenas a entrada disponível. Uma das metas do aprendizado semisupervisionado é o treinamento de classificadores quando uma

grande quantidade de exemplos não rotulados está disponível juntamente com um pequeno conjunto de exemplos rotulados [13].

Portanto, o projetista da rede deverá fazer uso de um desses três tipos de aprendizado na implementação de uma rede neural. Esta implementação deverá utilizar algum tipo de algoritmo, dentre os diversos existentes, que diferem basicamente pela maneira pela qual o ajuste dos pesos é feito.

2.4. TIPOS DE RNA

A criação do modelo de um neurônio biológico proposta por McCulloch e Pitts foi a base necessária para o trabalho de Frank Rosenblatt que, no ano de 1958, introduziu o conceito de aprendizado ao propor um modelo de rede neural chamado *Perceptron* [2]. Neste tipo de RNA, os neurônios MCP são organizados em duas camadas, de entrada e de saída e possuíam uma regra de aprendizado. Apesar da grande visibilidade tida por este trabalho, sua eficiência de convergência para uma solução só era possível caso o problema fosse linearmente separável.

Esta limitação no uso de redes neurais para solução de problemas mudou quando surgiram novos tipos de RNA. Uma delas, chamada de *Multilayer Perceptron* (MLP), foi criada a partir de uma modificação na estrutura da rede *Perceptron*, onde se adicionou mais camadas intermediárias à estrutura da rede.

Foram criados também outros tipos de RNAs, como a rede de Hopfield, que se caracteriza pela sua organização autoassociativa, não possuindo camadas. Neste tipo de rede, todos os neurônios são interligados e seu uso é indicado para problemas que envolvem otimização.

A rede de Kohonen faz parte de um grupo de RNA que levam o nome de *Self Organizing Maps* (SOM), possui a característica de ausência de camada intermediária. Trata-se de uma rede com aprendizado não supervisionado, onde as entradas semelhantes deverão produzir uma saída de forma categorizar os dados de entrada inseridos. Por essas características, este tipo de rede é indicado para problemas de classificação.

Já a rede do tipo *Radial Basis Function* (RBF) utiliza como função de ativação, na camada intermediária, do tipo base radial, como a função gaussiana.

Neste trabalho foi utilizado redes do tipo MLP e RBF. A seguir são descritas mais detalhadamente estes tipos de rede.

2.4.1. Redes MLP

Conforme já ilustrado na Figura 2.2, redes do tipo MLP possuem uma camada de entrada, pelo menos uma camada intermediária e uma camada de saída. Como também já mencionado, a partir da adição desta camada intermediária tornou-se possível a solução de problemas não linearmente separáveis.

Era necessário, no entanto, um mecanismo para treinamento da rede, de modo a permitir o ajuste de todos os pesos gradativamente. Diversos algoritmos foram desenvolvidos para possibilitar o treinamento de redes do tipo MLP, normalmente do tipo supervisionado.

O algoritmo mais conhecido e utilizado é o algoritmo *Backpropagation* (BP). Este algoritmo ficou bastante popularizado a partir do ano de 1986, apesar de ter sido proposto anos antes, com diferentes propósitos, por diversos pesquisadores, como Bryson e Ho (1969), Werbos (1974), Parker (1985) e Le Cun (1985) [2]. O nome dado ao algoritmo é em virtude de sua capacidade de retropropagar o erro de saída de um dado instante do treinamento para as camadas anteriores, permitindo o ajuste de todos os pesos da rede neural.

Deste modo, o algoritmo BP tem como minimizar o erro de saída, denominado erro quadrático médio (soma dos quadrados dos erros de saída), através do método do Gradiente Descendente, direcionando o ajuste dos pesos considerando o inverso do caminho de progressão do erro. Esta atualização dos pesos é limitada pela taxa de aprendizado, que assume valores baixos de modo a controlar a atualização, não possuindo um valor exato de utilização, irá depender dos testes e análises do projetista.

Para o cálculo do Gradiente e ajustes dos pesos, é necessário que as funções de ativação sejam não-lineares e diferenciáveis. Entre as funções mais utilizadas está a função sigmoide e tangente hiperbólica.

O funcionamento do algoritmo BP ocorre em duas etapas: *feedforward* e *feedback*. Cada fase percorre a rede em um sentido. A fase *feedforward* é utilizada para definir a saída da rede para um dado padrão de entrada. A fase *feedback* utiliza a saída desejada e a saída fornecida pela rede para atualizar os pesos de suas conexões.

A Equação 2.1 define o início da etapa *feedforward*, onde os neurônios da camada de entrada recebem os dados de entrada e efetua seu somatório, representado por *net*. A entrada do neurônio é representada pela letra *x* e a letra *i* define a posição do neurônio na camada de entrada.

$$net_i = \sum x_i \quad (2.1)$$

Após isso, as saídas dos neurônios da camada anterior são passadas para a camada intermediária. Esses valores são multiplicados pelos pesos correspondentes, gerando o valor net representado na Equação 2.2. A posição do neurônio na camada intermediária é representada por h , n representa a quantidade de neurônios na camada intermediária W representa o valor do peso entre o neurônio i e o neurônio h .

$$net_h = \left(\sum_{i=1}^{i=n} net_i W_{hi} \right) \quad (2.2)$$

É aplicada então a função de ativação aos valores de net_h . Com isso, tem-se o valor de saída da camada intermediária, representado por out na Equação 2.3. Na equação, a função de ativação é representada pelo termo f^h .

$$out_h = f^h (net_h) \quad (2.3)$$

A Equação 2.4 define o valor net da camada de saída, que é gerado a partir da multiplicação das saídas dos neurônios da camada intermediária pelos pesos correspondentes entre o neurônio h da camada intermediária e o neurônio o da camada de saída.

$$net_o = \left(\sum_{h=1}^{h=n} out_h W_{oh} \right) \quad (2.4)$$

Após isso, é calculado o valor de saída de cada neurônio da camada de saída, representado na Equação 2.5 pelo termo y . Para tanto, é aplicada a função de ativação nos valores de net calculados na etapa anterior. Normalmente, se escolhe uma função linear para a função de ativação nesta camada.

$$y = f^h (net_o) \quad (2.5)$$

Como último passo da etapa *feedforward*, é calculado o valor do erro através da subtração do valor de saída desejado, representado por d e o valor de saída calculado no passo anterior, representado por y . Com este valor, pode-se verificar o quanto a rede já possui de aprendizado, a medida que se consegue um valor baixo para o erro.

$$e = d - y \quad (2.6)$$

Com isto, se inicia a etapa *feedback*, onde o erro calculado será retropropagado de modo a alterar os valores dos pesos entre a camada de saída e a camada intermediária e entre a camada intermediária e a camada de entrada. Para isto, calcula-se o erro de determinada

camada, baseado no erro da camada posterior, representado por e_i , como mostra a Equação 2.7. A derivada da função de ativação em relação ao net_i é representada por \dot{f}_i e W corresponde ao peso entre o neurônio i e o neurônio j .

$$e_j = \sum_{i=1}^n e_i \cdot \dot{f}_i \cdot W_{ij} \quad (2.7)$$

Calcula-se então a variação dos pesos entre as camadas, que seja utilizado para atualização dos mesmos. A Equação 2.8 representa este cálculo. Onde o termo ΔW_{ij} representa o valor da variação do peso entre o neurônio i de uma camada posterior e o neurônio j de uma camada anterior. A taxa de aprendizado é representada pelo termo η , e_i representa o erro da camada posterior ao peso a ser atualizado, a derivada da função de ativação em relação ao net_i é representado por \dot{f}_i e out_j corresponde à saída da camada anterior ao peso a ser atualizado.

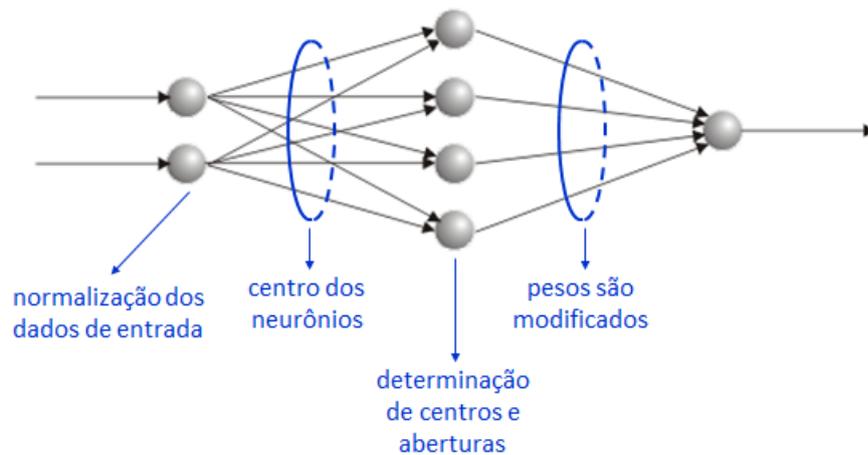
$$\Delta W_{ij} = \eta \cdot e_i \cdot \dot{f}_i \cdot out_j \quad (2.8)$$

Após a atualização dos pesos, inicia-se uma nova etapa *feedforward* e assim sucessivamente até que se atinja um número finito de interações ou até que o valor do erro encontrado seja menor que o erro que se deseja atingir, a ser definido pelo projetista da rede. Este algoritmo é um dos principais algoritmos quando se fala de redes MLP, por apresentar grande eficiência. Neste presente trabalho, o algoritmo BP foi utilizado para compor a metodologia proposta.

2.4.2. Redes RBF

Com estrutura semelhante à da rede MLP, a rede RBF também é organizada em camadas, sendo normalmente constituída de uma camada escondida. Sua principal característica é a utilização de funções de base radial como função de ativação na camada escondida. Outra particularidade desta rede é que o argumento da função de ativação, diferentemente das redes MLP que utiliza o produto escalar entre o vetor de entrada e o vetor de pesos, é dado pela norma euclidiana (distância) entre o vetor de entrada e os centros.

Figura 2.9: Estrutura da RNA do tipo RBF

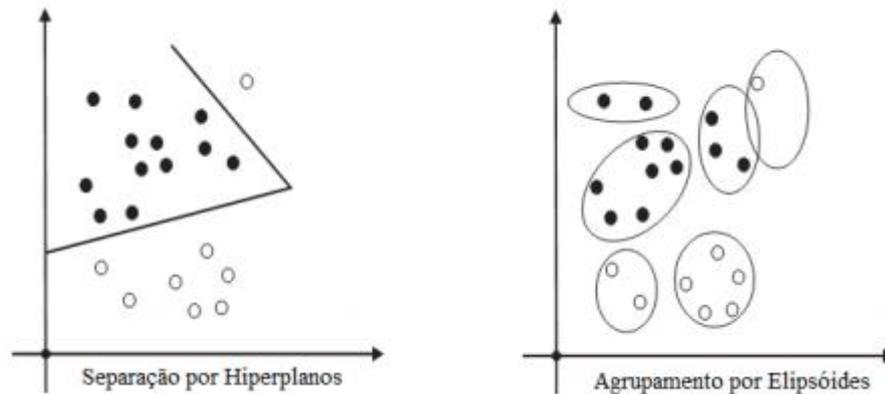


Fonte: de Almeida Neto [3].

A Figura 2.9 ilustra a estrutura e funcionamento de uma rede do tipo RBF. A função de cada camada em uma rede deste tipo é bem específica, de modo que seu treinamento é normalmente realizado em etapas separadas. Após a normalização dos dados de entrada, que são representados na faixa de 0 a 1 ou de -1 a 1 é iniciada a etapa de determinação dos centros e aberturas e servem de parâmetro para a função de ativação, normalmente do tipo gaussiana. Como os centros são calculados baseados na distância dos padrões de entrada fornecidos para a rede, nesta etapa do treinamento os dados de entrada são agrupados em um número finito de regiões, determinadas pelo centro e abertura, de modo que os dados que inicialmente eram não linearmente separáveis são organizados de modo a compor grupos de dados linearmente separáveis, como é ilustrado na Figura 2.10, enquanto as redes do tipo MLP faz a separação de dados por hiperplanos.

Após a determinação dos centros e aberturas e aplicação da função de ativação, inicia-se a etapa de treinamento entre a camada escondida e a camada de saída, que irá considerar o valor gerado na etapa anterior e irá trabalhar na determinação dos pesos entre essas camadas, tendo seu funcionamento, nesta etapa, igual ao treinamento da rede MLP.

Figura 2.10: Classificação com redes MLP(esquerda) e redes RBF(direita)



Fonte: Adaptado de Bragança Pereira e Rao [14].

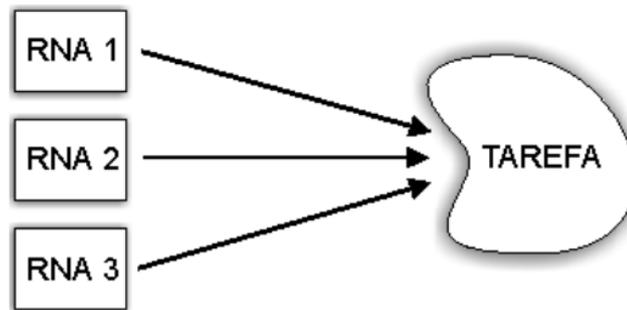
Os tipos de redes apresentados se mostram eficientes para solucionar os mais diversos problemas. Existem, no entanto, novas metodologias sendo estudadas e desenvolvidas, baseadas na junção de uma RNA com outras técnicas computacionais ou com modificações nos algoritmos apresentados de modo a trazer benefícios quanto à rapidez no aprendizado, entre outras adaptações. Neste contexto, pode-se citar a técnica de Múltiplas Redes Neurais Artificiais, técnica esta que foi utilizada para a elaboração deste trabalho. Segue, portanto, uma seção específica para esta abordagem.

2.5. MÚLTIPLAS REDES NEURAIIS ARTIFICIAIS

Múltiplas Redes Neurais (MRN) consiste em um arranjo de várias redes neurais, normalmente do mesmo tipo, solucionando um mesmo problema. A vantagem de usar múltiplas RNAs é a possibilidade de uma rede aprender parte da solução e assim as demais redes fazem o mesmo, desta forma o conjunto aprende a resolver o problema todo.

Ao usar múltiplas RNAs, existem formas de empregar as várias redes: combinação em conjunto, combinação modular e uma junção das duas formas anteriores. . A combinação em conjunto, também chamada de *ensemble*, representa uma junção de redes, onde cada uma tenta resolver o problema inteiro. A Figura 2.11 representa este tipo de combinação.

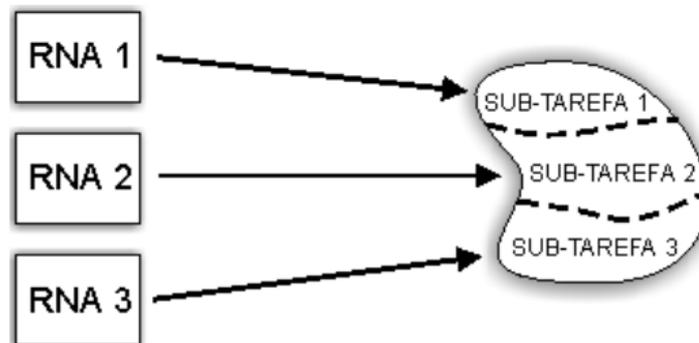
Figura 2.11: Combinação do tipo ensemble



Fonte: de Almeida Neto [3].

Já na combinação do tipo modular, o problema é dividido em partes e cada rede tentará resolver cada parte isoladamente. A Figura 2.12 faz a representação de uma combinação do tipo modular.

Figura 2.12: Combinação do tipo modular



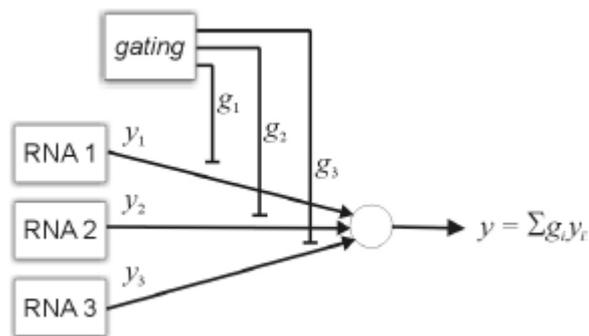
Fonte: de Almeida Neto [3].

Há também a possibilidade de unir características dos dois tipos de combinação, onde o problema é dividido em sub-problemas e várias redes tentam resolver cada um desses sub-problemas criados. Neste trabalho será utilizada a combinação do tipo *ensemble*.

Porém, ao se utilizar mais de uma rede na resolução de um problema, torna necessária a utilização de um mecanismo que faça com que as redes colaborem entre si, evitando que o aprendizado de uma rede seja prejudicado à medida que uma outra rede efetua seu treinamento. Para isto, muitos projetistas utilizam um coordenador para evitar conflitos no aprendizado das redes.

Quanto à sua implementação, um coordenador pode se apresentar como um modelo lógico-matemático, como pode ser visto nos trabalhos de Hashem [5] [6] [7], onde a saída das RNA é processada para formarem uma única saída através de uma combinação linear ótima. Mas também se pode empregar técnicas de IA para realizar a implementação do coordenador, entre elas tem-se o uso de sistemas *fuzzy*, como pode ser visto em [11], algoritmo genético, utilizado em [12], ou mesmo outra RNA, chamada de *gating*, como pode ser visualizado na Figura 2.13. Torna-se, portanto, uma responsabilidade a mais ao projetista ter que projetar e implementar mais um elemento, além das RNAs, que pode muitas vezes se tornar uma tarefa muito custosa, podendo comprometer o trabalho como um todo.

Figura 2.13: Combinação do tipo modular



Fonte: de Almeida Neto [3].

Por esses motivos, de Almeida Neto [3] desenvolveu uma forma de treinamento em que redes neurais do tipo MLP, utilizando o BP como algoritmo de treinamento, colaborem sem a necessidade de implementar um coordenador. Para tanto, o procedimento de treinamento consistiu em treinar redes de forma sequencial, em que cada uma deve realizar o seu aprendizado separadamente, aproveitando o que foi aprendido pelas redes anteriores.

A colaboração entre as redes sem um coordenador acontece na definição do valor desejado a ser aprendido por cada rede. Assim, ao ser treinada, uma rede tem como objetivo apresentar uma saída que, ao ser somada com a saída das outras redes, iguale ao valor desejado para a solução do problema.

Em termos matemáticos, o valor de erro utilizado no cálculo da atualização dos pesos da n -ésima rede é determinado pelo valor desejado para a solução do problema subtraído das saídas de todas as redes já anteriormente treinadas. A Equação apresenta a formulação deste cálculo:

$$e = d - y_1 - y_2 \dots - y_n \quad (2.10)$$

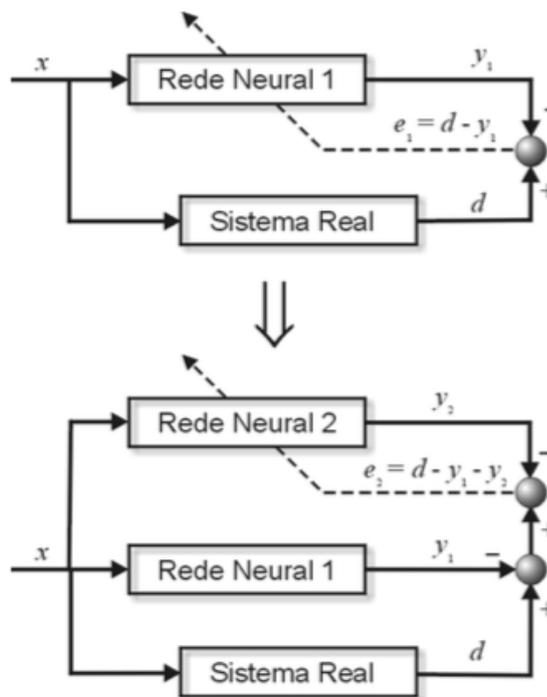
sendo e o erro da n -ésima rede, d o valor desejado para a solução do problema, y_1 a saída da rede 1 e assim sucessivamente até a rede n com saída y_n .

Deste modo, não é necessário um coordenador para determinar a colaboração entre as redes, isto porque o aprendizado da rede anterior é aproveitado pela nova rede, que parte deste aprendizado e o aprimora. Por conta disto, esta técnica pode ser caracterizada como autocoordenada.

Além disso, a existência de colaboração sistemática é evidente, pois ao longo do treinamento, cada rede que é adicionada deve achar uma combinação adequada de pesos para conseguir minimizar o erro, levando em consideração o que já foi produzido por todas as redes anteriores.

Ao aplicar a técnica das MRN autocoordenadas (MRNA) em um sistema real, o conjunto de entradas deverá perpassar todas as redes, na sequência em que foram treinadas, pois como visto, cada rede utiliza o valor de saída das redes anteriores. Na Figura 2.14 pode-se ver o esquema do treinamento de um arranjo com duas redes.

Figura 2.14: Arranjo de MRNA com duas redes



Fonte: de Almeida Neto [3].

Para garantir que o treinamento de todas a partir da segunda rede deve começar exatamente do ponto onde a rede anterior parou, inicialmente, zera-se a saída da nova rede, normalmente zerando os pesos anteriores a camada de saída dessa nova rede, garantindo assim que elas providenciem, em um primeiro momento, saída nula.

Todavia, neste trabalho uma inovação é introduzida ao mecanismo de autocoordenação desenvolvido por de Almeida Neto [3], combinando-se, porém, redes de tipos diferentes: *Multilayer Perceptron* (MLP) e *Radial Basis Function* (RBF).

CAPÍTULO 3 - COLABORAÇÃO ENTRE REDES MLP E RBF

Este capítulo aborda sobre a metodologia desenvolvida, onde redes MLP e RBF atuam em conjunto para reconhecer padrões em duas bases de dados e que gerou o artigo científico publicado no periódico JCIT – *Journal of Convergence Information Technology* (ISSN 1975-9320), que possui Qualis B1 em computação, volume 11, número 5. Uma tradução livre do artigo encontra-se como apêndice deste trabalho.

Apesar de uma rede neural ter a capacidade de aproximar qualquer função [2], é comum que a solução de um problema se torne uma tarefa exaustiva para o projetista. Isto porque não existe uma metodologia que possa definir qual estrutura utilizar, como efetuar o treinamento e com que valores inicializar os parâmetros utilizados na rede.

Além disso, existe a possibilidade de aprisionamento em um mínimo local. Isto porque, ao se basear na procura de um ponto mínimo mais próximo do ponto atual, na busca de um mínimo global, pode-se chegar a um ponto em que a rede não apresenta um erro decrescente, por estar aprisionado em um mínimo local.

Devido essas dificuldades, várias técnicas foram e continuam sendo desenvolvidas com o objetivo de amenizá-las. Destaca-se a técnica desenvolvida por de Almeida Neto [3], que propõe a colaboração e autocoordenação de redes MLP, de modo que o aprendizado de uma rede não é descartado quando seu erro estabiliza e o aprendizado não é suficiente para solucionar o problema. Deste modo é possível que o aprendizado não seja descartado, sendo utilizado como ponto de partida para a nova rede.

Redes MLP e RBF apresentam a característica inerente às RNAs, a capacidade de aproximação universal. No entanto, estas redes apresentam algumas diferenças quanto à estrutura e como os dados de entrada são tratados. Além da diferença quanto à função de ativação utilizada na camada intermediária e do fato que redes RBF apresentam sempre uma única camada intermediária, uma das principais diferenças está na forma como os dados de entrada são processados. Desta forma, buscou-se unir as peculiaridades de cada rede de modo que as diferentes técnicas trabalhem juntas para solucionar problemas.

3.1. DESCRIÇÃO DA METODOLOGIA DE TREINAMENTO

Baseado nas informações tratadas até aqui, este trabalho sugere uma forma de utilizar a colaboração entre redes neurais, visando a resolução de problemas. Tendo como base o

trabalho desenvolvido por de Almeida Neto [3], a colaboração entre as redes é feita de forma autocoordenada, ou seja, sem a utilização de um coordenador entre as redes.

Como já citado, utiliza-se dois tipos diferentes de rede, a MLP e a RBF, de modo que o funcionamento peculiar de cada uma tratar o problema possa contribuir para a resolução do mesmo.

A implementação foi realizada utilizando o software MATLAB. Trata-se de um sistema interativo e uma linguagem de programação baseados em matrizes, onde as soluções e problemas são expressos em linguagem matemática. Por possuir amplo suporte às operações matemáticas, é possível realizar análise numérica, operações com matrizes, construção de gráficos com considerável facilidade comparada a algumas linguagens. Apesar de já possuir bibliotecas específicas para aplicações com Redes Neurais Artificiais, neste trabalho elas não foram utilizadas.

O objetivo é comparar as abordagens de treinamento quanto à qualidade do aprendizado. Para tanto, é verificada a taxa de acerto dos testes de generalização, ou seja, após o treinamento, é verificado o quanto a rede consegue generalizar seu aprendizado para dados que não foram utilizados no processo de treinamento.

Desta forma, foi avaliada a qualidade de generalização em duas bases de dados diferentes. A primeira vem a ser a *Breast Cancer Wisconsin*, que contém 699 eventos associados a nove atributos. Cada um desses nove atributos correspondem às características de células avaliadas, como espessura, uniformidade, entre outras. Dos 699 padrões de entrada, 458 são classificados como benignos e 241 como malignos. Deste modo, o objetivo da MRN é mapear esses nove atributos às suas respectivas classes (malignos e benignos).

A segunda base de dados trata-se da *Vertebral Column*, que contém 310 padrões de entrada, possuindo seis atributos cada. Cada um desses seis atributos corresponde às características biomecânicas de pacientes decorrentes da forma e orientação da pelve e coluna lombar. Esses atributos estão associados à classificação da coluna vertebral como normal ou anormal, para determinar quando apresentam algum distúrbio associado à hérnia.

As duas bases de dados estão disponíveis no repositório UCI e podem ser acessadas pelo endereço disposto em [8] e [9].

3.2. METODOLOGIA DE AVALIAÇÃO DE RESULTADOS

Com o objetivo de avaliar todas as abordagens, tanto com poucos neurônios na camada escondida, quanto com muitos, os dois programas desenvolvidos foram executados

variando-se a quantidade de neurônios na camada escondida em intervalos de sete unidades, começando em 7 e indo até 105, esta variação foi determinada principalmente para possibilitar a comparação com outras metodologias, que efetuaram a variação da mesma forma. Logo, foram geradas 15 configurações distintas de rede.

Para cada quantidade de neurônios na camada escondida, cada programa foi executado 20 vezes, cada execução com valores iniciais dos pesos diferentes. Além disso, 80% dos dados disponíveis foram utilizados para treinamento e 20% para teste e a cada quatro inicializações destas 20, mudou-se o conjunto de dados utilizados durante o treinamento e teste, fazendo com que os dados utilizados no treinamento sejam alternados para que os resultados não sejam dependentes de um conjunto de dados específico. Assim, foi realizado um total de 300 execuções (20 inicializações x 15 configurações) por programa.

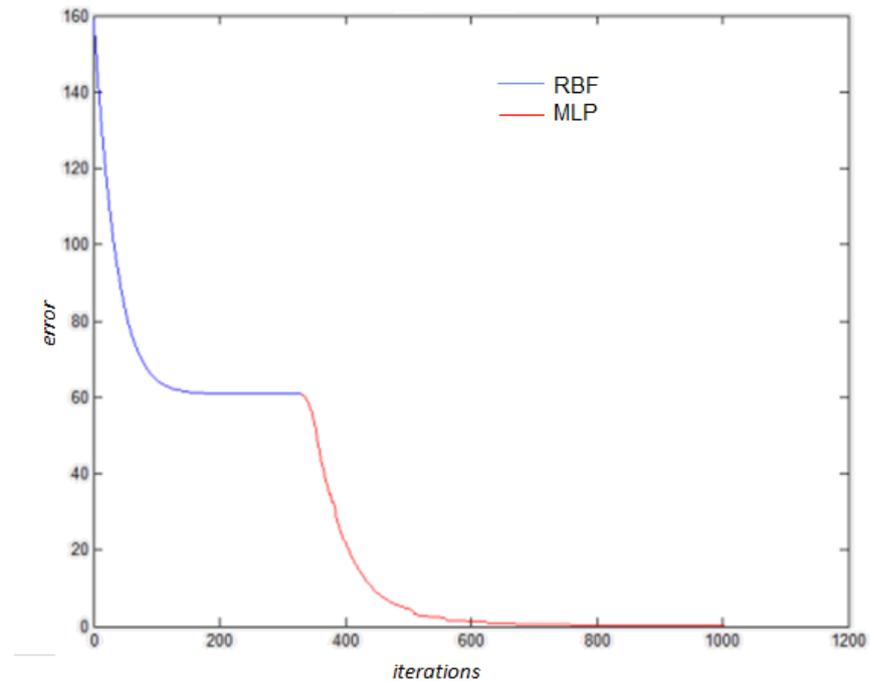
Para validar os testes realizados com a primeira base de dados, utilizou-se o trabalho realizado por de Meneses Carvalho [10], onde é proposta uma variação do algoritmo *Backpropagation*, através da utilização do termo *momentum* seletivo. Desta forma, os resultados obtidos por esse trabalho quanto à qualidade do aprendizado são utilizados para validar esta metodologia.

3.3. RESULTADOS

A Figura 3.1 ilustra o gráfico gerado em um dos treinamentos com a base de dados relacionada ao câncer. O treinamento foi inicializado com uma rede do tipo RBF, percebe-se, no entanto, que em dado momento, o aprendizado da rede, representado pela diminuição do erro de saída, estabiliza. Caso a metodologia não fosse de múltiplas redes, seria necessário iniciar o treinamento novamente, alterando parâmetros da rede neural ou a estrutura da rede para possibilitar melhor aprendizado. Porém, na metodologia proposta, é possível inserir uma nova rede que, partindo do aprendizado da rede anterior, irá dar continuidade ao treinamento. Como pode ser verificado, a nova rede inserida foi MLP.

Neste trabalho, o momento de inserção da nova rede foi feito através da visualização do gráfico do erro, de modo que a decisão fica a cargo do projetista, ao verificar que o erro da rede estabilizou e o aprendizado estagnou. Vale ressaltar também que a quantidade de redes utilizada variou entre 2 e 3 redes para as duas base de dados utilizadas, a depender do valor atingido para o erro. Os valores definidos para a taxa de aprendizado e para o erro quadrático médio foram 0.0001 e 0.0145 respectivamente.

Figura 3.1: Gráfico do Erro Quadrático Médio



Fonte: Elaborado pelo autor.

Assim, ao fim da execução do programa, conforme mencionado anteriormente, a Tabela 3.1 foi gerada. Percebe-se que houve uma melhora nos resultados de acerto quando utilizada a estrutura de múltiplas redes. Vale ressaltar que no trabalho desenvolvido por de Meneses Carvalho [10] houve situações de inicialização que levaram o erro à um mínimo local, de modo que o aprendizado foi fracassado. O mesmo não ocorre na metodologia proposta por esse trabalho, já que, caso uma rede venha a ficar aprisionada em um mínimo local, a nova rede iniciada a partir de uma nova técnica, possibilitará a continuidade do aprendizado.

Tabela 3.1: Taxas de acerto – Base de dados 1

Neurônios na Camada Escondida	BP Tradicional (%)	BP com Momentum Seletivo (%)	MRN com redes MLP e RBF
7	98,04	97,73	98,78
14	98,31	98,09	98,64
21	98,46	98,24	98,48
28	98,36	98,31	98,59
35	98,58	98,21	98,63
42	98,48	98,26	98,67
49	98,46	98,26	98,56

56	98,24	97,92	98,56
63	98,34	97,97	98,52
70	98,43	98,07	98,48
77	98,36	98,02	98,59
84	98,53	98,21	98,52
91	98,46	98,19	98,71
98	98,31	98,26	98,52
105	98,48	98,02	98,71
Média	98,39	98,12	98,60

Fonte: Elaborado pelo autor.

É comum que em problemas de classificação, quando há duas classes possíveis, que se utilizem alguns mecanismos que servem como métricas para os resultados obtidos. Considerando-se um conjunto de amostras, onde uma instância I pode assumir valores no conjunto positivo e negativo, pode-se ter quatro situações, como segue:

- Se a instância é positiva e é classificada como positiva, conta-se como verdadeiro positivo (TP);
- Se a instância é positiva e é classificada como negativa, conta-se como falso negativo (FN);
- Se a instância é negativa e é classificada como tal, conta-se como verdadeiro negativo (TN);
- Se a instância é negativa e é classificada como positiva, conta-se como falso positivo (FP).

Portanto, pode-se construir uma matriz de 2 por 2 com os dados acima, chamada matriz de confusão, que serve como base para muitas métricas que podem ser aplicadas à classificação.

A Tabela 3.2 apresenta a matriz de confusão gerada na generalização em que a rede possuía sete neurônios na camada escondida. É possível verificar valores quanto ao desempenho da rede em classificar corretamente os padrões independentemente da classe. A partir da matriz de confusão, podem-se obter algumas métricas relacionadas à validação de classificadores. Dentre elas, estão valores referentes à sensibilidade e especificidade.

Tabela 3.2:Matriz de Confusão – Base de dados 1

		Classificação		Totais
		Positivo	Negativo	
Real	Positivo	44	0	44
	Negativo	2	90	92
Totais		46	90	136

Fonte: Elaborado pelo autor.

Sensibilidade, ou taxa de verdadeiros positivos, é a proporção de verdadeiros positivos, ou seja, a capacidade do sistema em prever corretamente a condição para casos que realmente a possuem. Seu cálculo é dado como mostra a Equação 3.1, onde TP corresponde à quantidade de verdadeiros positivos e FN de falso negativo.

$$SEN = \frac{TP}{TP+FN} \quad (3.1)$$

Quanto à especificidade, também chamada de taxa de verdadeiro negativo, é a proporção de verdadeiros negativos, ou seja, a capacidade do sistema em prever corretamente os casos que não possuem determinada condição. Por exemplo, dos dados referentes a câncer de mama, seriam os casos que a saída deve ser negativa. A Equação 3.2 determina como este valor é calculado, onde TN se refere à quantidade de verdadeiros negativos e FP de falsos positivos.

$$ESP = \frac{TN}{TN+FP} \quad (3.2)$$

Desta forma, a Tabela 3.3 apresenta os dados obtidos para sensibilidade e especificidade na generalização, considerando uma inicialização e a variação de quantidade de neurônios conforme foi utilizada.

Tabela 3.3:Tabela de dados para curva ROC – Base de dados 1

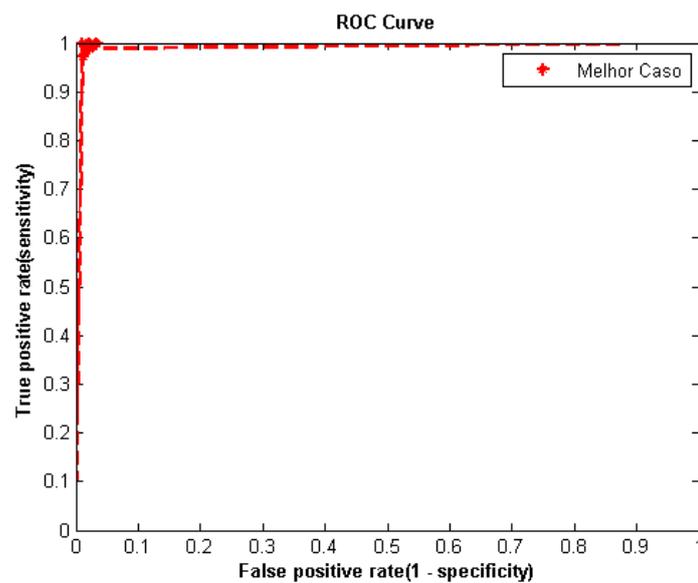
Neurônios na Camada Escondida	SEN	ESP
7	1,000	0,989
14	0,978	0,989
21	0,978	0,989
28	1,000	0,978

35	1,000	0,989
42	1,000	0,989
49	1,000	0,978
56	1,000	0,989
63	1,000	0,978
70	0,978	0,989
77	1,000	0,989
84	0,947	1,000
91	1,000	0,989
98	0,978	0,989
105	0,973	1,000

Fonte: Elaborado pelo autor.

Com os dados de sensibilidade e especificidade, de todos os testes realizados, foi possível gerar o gráfico da curva ROC, que é uma demonstração bidimensional do desempenho de um classificador. Quanto mais próximos os pontos gerados pela relação entre especificidade e sensibilidade estiverem do canto superior esquerdo, mais eficiente é o classificador. A Figura 3.2 representa a curva ROC da configuração que obteve melhor resultado na generalização, com 7 neurônios na camada escondida.

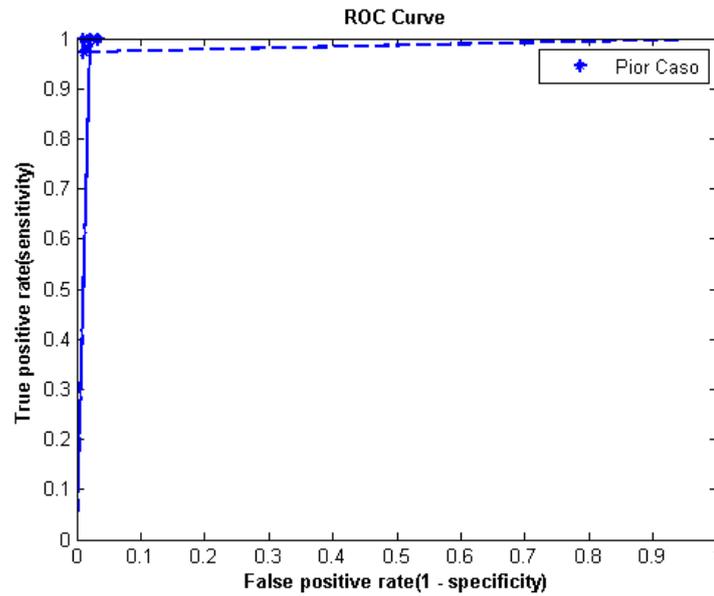
Figura 3.2: Curva ROC (Melhor Caso) – Base de dados 1



Fonte: Elaborado pelo autor.

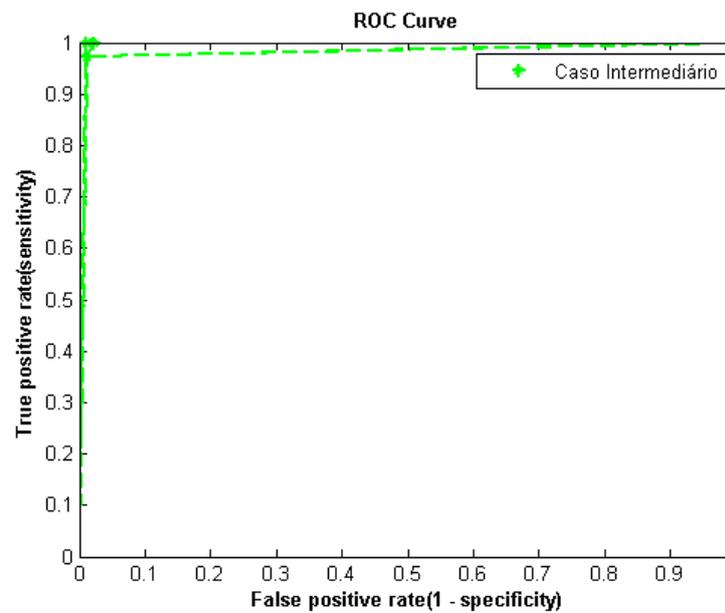
As Figuras 3.3 e 3.4 representam a curva ROC para configurações que apresentaram o pior resultado de generalização e um resultado intermediário, respectivamente.

Figura 3.3: Curva ROC (Pior Caso)– Base de dados 1



Fonte: Elaborado pelo autor.

Figura 3.4: Curva ROC (Intermediário)– Base de dados 1



Fonte: Elaborado pelo autor.

A Tabela 3.4 foi gerada a partir da execução da estrutura de múltiplas redes utilizando como base de dados para classificação de problemas na coluna vertebral, como foi mencionado anteriormente.

Tabela 3.4: Taxas de acerto – Base de dados 2

Neurônios na Camada Escondida	MRN com redes MLP e RBF
7	91,31
14	91,59
21	91,86
28	91,45
35	91,59
42	91,93
49	91,31
56	92,01
63	91,52
70	91,45
77	91,79
84	91,52
91	91,31
98	91,86
105	91,24
Média	91,58

Fonte: Elaborado pelo autor.

Como forma de mensurar o desempenho da estrutura de redes nesta base de dados, utilizou-se o trabalho desenvolvido em [4] onde foi estudada, utilizando a mesma base de dados, a melhoria de se utilizar comitês de classificadores, onde a metodologia foi a escolha de cinco classificadores entre 50 obtidos em cada uma das rodadas de treino, onde estes escolhidos possuem melhores taxas de acerto. O tipo de rede utilizado foi MLP. Neste trabalho, o comitê de redes MLP apresentou média de acerto de 88,48%. Percebe-se uma melhora na classificação quando se compara com os resultados obtidos neste trabalho.

A Tabela 3.5 apresenta os dados obtidos para sensibilidade e especificidade para a segunda base de dados na generalização, considerando uma inicialização e a variação de quantidade de neurônios conforme foi utilizada. A partir dos dados de sensibilidade e especificidade de todos os testes realizados, foi possível gerar a curva ROC conforme pode

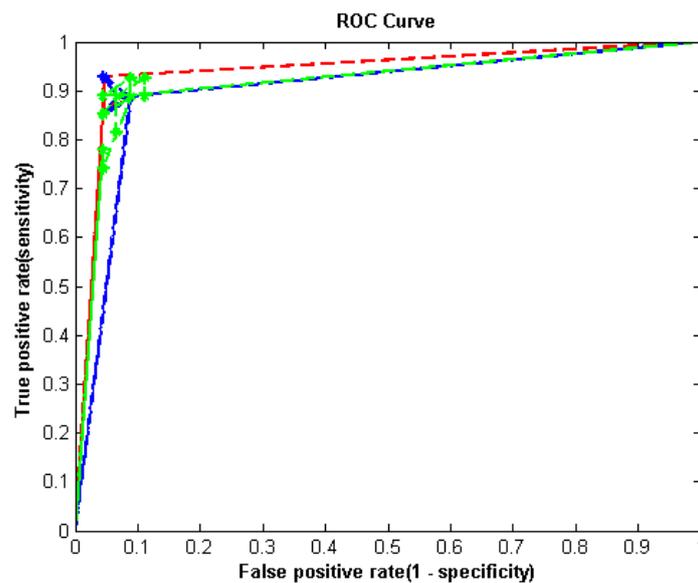
ser visto na Figura 3.5, que apresentam o melhor resultado de generalização (em vermelho), o pior (em azul) e um caso intermediário (em verde).

Tabela 3.5:Tabela de dados para curva ROC - Base de dados 2

Neurônios na Camada Escondida	SEN	ESP
7	0,778	0,956
14	0,741	0,956
21	0,741	0,911
28	0,889	0,911
35	0,815	0,933
42	0,889	0,933
49	0,889	0,956
56	0,926	0,911
63	0,889	0,911
70	0,852	0,956
77	0,889	0,933
84	0,926	0,889
91	0,889	0,889
98	0,889	0,911
105	0,889	0,933

Fonte: Elaborado pelo autor.

Figura 3.5:Curva ROC – Base de dados 2



Fonte: Elaborado pelo autor.

CAPÍTULO 4 - CONCLUSÃO

4.1. CONSIDERAÇÕES FINAIS

Redes Neurais Artificiais constituem uma ampla gama de técnicas para aprendizado e solução de problemas em que os paradigmas tradicionais de desenvolvimento não são adequados por se tratarem de problemas de complexa resolução, onde o modelo lógico-matemático não é conhecido ou é de difícil obtenção.

No entanto, existem problemas que até mesmo uma RNA apresenta dificuldade de encontrar a solução. Isto ocorre porque não há uma metodologia que guie o projetista para a determinação da topologia, inicialização de parâmetros e método de treinamento de modo a garantir que a rede “aprenda”.

Partindo dessa condição, diversas técnicas vêm sendo desenvolvidas de modo a auxiliar os algoritmos de treinamento a cumprir sua tarefa de forma eficiente. Entre as técnicas desenvolvidas, está a que combina redes neurais durante o processo de treinamento, para a solução de um único problema, denominada Múltiplas Redes Neurais.

Nesta técnica, normalmente há a utilização de um elemento coordenador para evitar que o aprendizado de uma rede entre em conflito com as demais. No entanto, de Almeida Neto [3] desenvolveu uma metodologia de treinamento baseada em Múltiplas Redes, mas que dispensa o uso de coordenador.

Durante as pesquisas realizadas, observou-se que, entre os trabalhos que combinam várias redes, não existem trabalhos que utilizam a técnica acima mencionada para promover a colaboração de tipos diferentes de RNAs.

Desta forma, foi analisada a possibilidade de se utilizar a técnica de Múltiplas Redes Neurais autocoordenadas com redes do tipo *Multilayer Perceptron* (MLP) e *Radial Basis Function* (RBF).

A metodologia proposta foi submetida a uma série de testes em duas bases de dados e os resultados foram comparados ao algoritmo BP tradicional, à uma nova metodologia que propõe uma melhora do BP e à uma estratégia em que um conjunto de redes MLP efetuam o treinamento e as redes com melhores resultados são selecionadas. A metodologia proposta neste trabalho apresentou uma melhora quanto à generalização.

Desta forma, as vantagens de cada tipo de rede trabalham de forma conjunta e autocoordenada para a solução de problemas. Observou-se que os resultados referentes à especificidade e sensibilidade do classificador proposto são muito bons, ao apresentar resultados satisfatórios para as duas possíveis classes.

A partir da análise desses resultados, é justo afirmar que utilizar Múltiplas Redes Neurais com tipos de redes MLP e RBF de modo autocoordenado é uma boa alternativa para a busca de solução para problemas complexos. A fim de consolidar esta afirmação, é interessante que outros testes sejam efetuados, através de mais variações quanto a estrutura e parâmetros das redes.

4.2. CONTRIBUIÇÕES DESTE TRABALHO

As principais dificuldades ao se utilizar uma única rede foram mostradas, e foi proposta uma metodologia que possibilita coordenação entre redes de tipos diferentes, destacando sua vantagem de uso e funcionamento.

Desta forma, abriu-se uma linha de pesquisa para se investigar novas possibilidades de autocoordenação entre diferentes técnicas, partindo do trabalho desenvolvido que mostrou ser possível a autocoordenação entre redes MLP e RBF.

No intuito de comparar e validar a metodologia proposta, foi implementado o programa que permite o aprendizado via autocoordenação entre redes MLP e RBF e foi executado diversas vezes para dois problemas diferentes, relacionado ao câncer de mama e de problemas na coluna vertebral. Os resultados produzidos garante a validade da metodologia, além de apresentar uma melhora quando a qualidade do aprendizado.

Além das contribuições citadas, este trabalho serve como referencial teórico para o direcionamento de outros trabalhos que buscam melhorar o treinamento e qualidade de aprendizado através de Múltiplas Redes Neurais, haja vista apresentar um conjunto de pesquisas com o mesmo objetivo.

4.3. TRABALHOS FUTUROS

Dando prosseguimento a pesquisa desenvolvida, uma gama de outros trabalhos pode ser derivada da tentativa de prover e aprimorar a colaboração entre RNAs.

Entre estes, a fim de melhor conhecer o desempenho da nova metodologia, um importante trabalho futuro seria aplicar a metodologia proposta em outros tipos de problemas

para uma validação mais precisa. Esses problemas devem variar principalmente em quantidade de dados e complexidade.

Outro trabalho seria investigar a autocoordenação de outros tipos de RNAs, ampliando ainda mais a abrangência da técnica desenvolvida por de Almeida Neto [3].

Além da integração de outros tipos de RNAs de modo autocoordenado, poderia ser investigada a integração de outros tipos de elementos otimizadores, como algoritmos genéticos, sistemas *fuzzy*, sem a presença do coordenador.

Mais uma alternativa seria a comparação da metodologia proposta neste trabalho com outras técnicas de Inteligência Artificial, como Árvores de Decisão e Máquinas de Vetor Suporte (SVM).

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] HAYKIN, Simon S. **Redes neurais artificiais: princípio e prática**. 2ª Edição, Bookman, São Paulo, Brasil, 2000.
- [2] BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B., **Redes Neurais Artificiais: Teoria e Aplicações**, Editora: LTC, 2014.
- [3] DE ALMEIDA NETO, A., **Aplicações de Múltiplas Redes Neurais em Sistemas Mecatrônicos**, Tese de Doutorado, Instituto Tecnológico da Aeronáutica (ITA), 2003.
- [4] NETO, A.R. Rocha; BARRETO, G. A. **On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis**. IEEE Transactions on Latin America, v. 7, n. 4, p. 487-496, 2009.
- [5] HASHEM, S. et al., **Optimal Linear Combinations of Neural Networks: An Overview**, IEEE World Congress on Computational Intelligence, pp. 1507, 1512, vol. 3, 1994.
- [6] HASHEM, S.; SCHMEISER, B., **Improving Model Accuracy Using Optimal Linear Combinations of Trained Neural Networks**, IEEE Transactions on Neural Networks, pp. 792, 794, vol. 6, no. 3, 1995.
- [7] HASHEM, S., **Algorithms for Optimal Linear Combinations of Neural Networks**, International Conference on Neural Networks, pp. 242, 247, vol. 1, 1997.
- [8] WOLBERG, W. H. **Breast Cancer Wisconsin (Original) Data Set**. Disponível em <[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))>.
- [9] MOTA, H. **Vertebral Column Data Set**. Disponível em <<https://archive.ics.uci.edu/ml/datasets/Vertebral+Column>>.
- [10] DE MENESES CARVALHO, D. S.. **Aceleração do Treinamento via Backpropagation com termo momentum seletivo**. 2016. Dissertação de Mestrado. Universidade Federal do Maranhão.
- [11] CHO, Sung-Bae; KIM, Jin H. **Combining multiple neural networks by fuzzy integral for robust classification**. IEEE Transactions on Systems, Man, and Cybernetics, v. 25, n. 2, p. 380-384, 1995.
- [12] CHO, Sung-Bae. **Fusion of neural networks with fuzzy logic and genetic algorithm**. Integrated Computer-Aided Engineering, v. 9, n. 4, p. 363-372, 2002.

- [13] SILVA, M. M., **Uma Abordagem Evolucionária para o Aprendizado Semi-Supervisionado em Máquinas de Vetor Suporte**, Monografia, Universidade Federal de Minas Gerais (UFMG), 2008.
- [14] DE BRAGANÇA PEREIRA, Basilio; RAO, Calyampudi Radhakrishna; RAO, Marepelli. **Data Mining Using Neural Networks: A Guide for Statisticians**. State College, Pennsylvania 2009.
- [15] WOŹNIAK, Michał; GRAÑA, Manuel; CORCHADO, Emilio. **A survey of multiple classifier systems as hybrid systems**. Information Fusion, v. 16, p. 3-17, 2014.
- [16] KHOTANZAD, Alireza; SADEK, Nayera. **Multi-scale high-speed network traffic prediction using combination of neural networks**. In: Neural Networks, 2003. Proceedings of the International Joint Conference on. IEEE, 2003. p. 1071-1075.
- [17] DHLAMINI, Sizwe M.; MARWALA, Tshilidzi. **Bushing diagnostics using an ensemble of parallel neural networks**. In: Electrical Insulating Materials, 2005.(ISEIM 2005). Proceedings of 2005 International Symposium on. IEEE, 2005. p. 289-292.
- [18] ABDELLA, Mussa; MARWALA, Tshilidzi. **Treatment of missing data using neural networks and genetic algorithms**. In: Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on. IEEE, 2005. p. 598-603.
- [19] WOODS, Kevin; BOWYER, Kevin; KEGELMEYER JR, W. Philip. **Combination of multiple classifiers using local accuracy estimates**. In: **Computer Vision and Pattern Recognition**, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on. IEEE, 1996. p. 391-396.
- [20] WANG, Wenjia; JONES, Phillis; PARTRIDGE, Derek. **Diversity between neural networks and decision trees for building multiple classifier systems**. In: Multiple Classifier Systems. Springer Berlin Heidelberg, 2000. p. 240-249.
- [21] LANGDON, William B.; BARRETT, S. J.; BUXTON, Bernard F. **Combining decision trees and neural networks for drug discovery**. In: Genetic Programming. Springer Berlin Heidelberg, 2002. p. 60-70.
- [22] SOARES, Rodrigo GF et al. **Using accuracy and diversity to select classifiers to build ensembles**. In: Neural Networks, 2006. IJCNN'06. International Joint Conference on. IEEE, 2006. p. 1310-1316.
- [23] NASCIMENTO, Diego SC; COELHO, André LV. **Bagging heterogêneo evolutivo: Caracterização e análise comparativa com ensembles homogêneas de redes neurais RBF**. Anais do IX Simpósio Brasileiro de Automação Inteligente, p. 6, 2009.

- [25] WOLBERG, W. H. **Breast Cancer Wisconsin (Original) Data Set**. Disponível em <[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))>.
- [26] MOTA, H. **Vertebral Column Data Set**. Disponível em <<https://archive.ics.uci.edu/ml/datasets/Vertebral+Column>>.
- [27] ZANCHETTIN, C. et al, **Hybrid Training Method for MLP: Optimization of Architecture and Training**, IEEE Transactions on Systems, Man, and Cybernetics, pp. 1097, 1109, vol. 41, no. 4, 2011.
- [28] AHMAD, Zainal et al. **Improving water quality index prediction in Perak River basin Malaysia through a combination of multiple neural networks**. International Journal of River Basin Management, p. 1-9, 2016
- [29] D'ANDREA, Eleonora; LAZZERINI, Beatrice; PALLESCHI, Vincenzo. **Combining Multiple Neural Networks to Predict Bronze Alloy Elemental Composition**. In: **Advances in Neural Networks**. Springer International Publishing, 2016. p. 345-352.
- [30] LI, Fei; LIU, Jin. **Time Series Prediction Based on Multiple Artificial Neural Network**. 2016.
- [31] WANG, Xiao-li et al. **Online Estimation of the pH Value for Froth Flotation of Bauxite Based on Adaptive Multiple Neural Networks**. IFAC-PapersOnLine, v. 49, n. 20, p. 149-154, 2016.
- [32] KUNG, Hsu-Yang et al. **Accuracy Analysis Mechanism for Agriculture Data Using the Ensemble Neural Network Method**. Sustainability, v. 8, n. 8, p. 735, 2016.
- [33] XIE, Feng-ying et al. **Melanoma classification on dermoscopy images using a neural network ensemble model**. IEEE Transactions on Medical Imaging, 2016.

APÊNDICE A – ARTIGO VERSÃO ORIGINAL

Multiple Neural Networks Using MLP and RBF Networks

^{1,2}Aline Porfiro Teixeira, ³Areolino de Almeida Neto

¹State University of Maranhão-Brazil, alineporfirot@gmail.com

²Federal Institute of Maranhão-Brazil, aline.teixeira@ifma.edu.br

³Federal University of Maranhão-Brazil, areolino@ufma.br

Abstract

This project aims to perform an improvement in pattern recognition using Multiple Artificial Neural Networks of different types (MLP and RBF) in a selfcoordinated way. Multiple Artificial Neural Networks are used in cases where learning with only one network is very hard. Besides, in order to avoid conflicts among networks, a coordinator is normally used. However, this article proposes a model of Multiple Neural Networks by combining two different types of networks without using a coordinator, but avoiding conflicts. Some results obtained are presented in two data bases used and through the comparison with other methodologies, it was possible to verify an improvement in the results. Therefore, the proposed methodology is attractive in the context of Multiple Neural Networks, because it uses different models of Neural Networks in a selfcoordinated way.

Keywords: *Pattern Recognition, Artificial Neural Network, Multiple Neural Networks*

1. Introduction

As a mechanism for classification, Pattern Recognition has been increasingly useful and necessary in computational systems from the most diverse areas. In order to classify objects within a number of categories or classes, this computational mechanism allows the development of a wide range of applications for scientific and technological use, such as speech recognition, document classification, handwriting recognition and face recognition among others.

Among the most varied methods used to perform Pattern Recognition are Artificial Neural Networks (ANNs). From a training process, an ANN is able to accumulate knowledge and generalize it to other cases not yet seen. Thus, learning to identify characteristics of the patterns presented in the training, ANN will be able to recognize other objects with the same characteristics.

However there are situations where finding the solution using only a neural network becomes a very difficult task. Functioning as a functions approximators, in these cases, it would take many adjustments in the topology of the ANN used so that it can get close to the function of the solution during the learning process. Thus, the structure of the ANN is adapted or some parameters of the training are modified so that the desired approximation can be made.

On situations as this, alternatively it is possible to use more than one neural network to facilitate the learning process, which working together are called the Multiple Neural Networks.

Therefore, this work intends to present an innovative model of multiple neural networks combining networks of different types: Multilayer Perceptron (MLP) and Radial Basis Function (RBF). Moreover, another innovative aspect is that the combination occurs without using a coordinator, but nonetheless avoiding conflicts between the networks during the learning process.

2. Artificial Neural Networks

An artificial neural network can be defined as an intense and parallel distributed processor whose processing units are very simple, which have the natural purpose to store knowledge and make it available for use [1].

Thus, a neural network resembles the human brain by acquiring knowledge through a learning process and by having forces of connections between neurons, known as synaptic weights, used to store the acquired knowledge.

Regard to its structure, an ANN is organized into layers of neurons, which can have only two layers, input and output, or intermediate layers (also called hidden or occult layer). Figure 1 represents the basic structure of a neural network with a hidden layer.

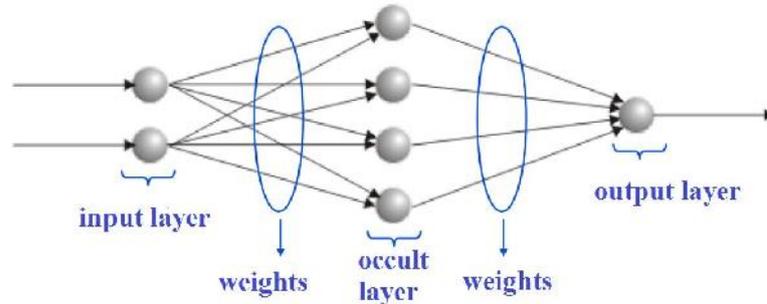


Figure 1. Structure of the ANN of MLP type

Being the basic processing unit of a network, an artificial neuron has the generic model as can be visualized in Figure 2.

Then there is the combination of the input data represented by "net", which usually is the simple summation of the input values. The "out" value is the output of the neuron, being produced by applying the activation function and which will be taken into the input of another neuron or to the output of the ANN. Among the most commonly used activation functions are the linear function, the sigmoid function, or the hyperbolic tangent function.

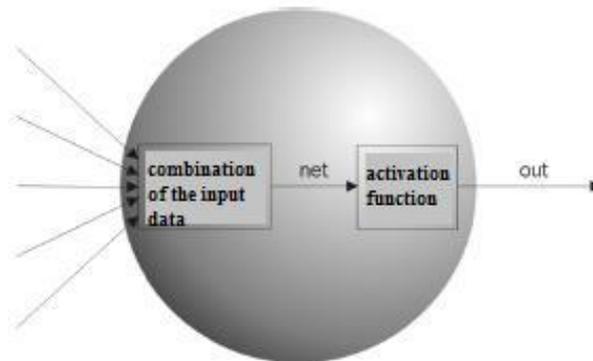


Figure 2. Representation of an artificial neuron

The ANNs are divided into several models, according to the network structure, learning form and neuron model. Among the existing models, we highlight the MLP and RBF networks:

- Multilayer Perceptron (MLP): It is created from the addition of one or more intermediate layers to the Perceptron network [1], where the activation function is necessarily non-linear, making this type of network applicable to problems regardless of the classes being linearly separable. Thus, an MLP network has an input layer, at least one intermediate layer and an output layer. Because it is simple to construct, this type of network is one of the most used for developing solutions to the most diverse problems[2]. There are several learning algorithms used in an MLP network. One of the most used is the Backpropagation algorithm, where the error of the output is backpropagated to the previous layers in order to adjust the weights;

- Radial Basis Function (RBF): This network has the main characteristic of using as an activation function a radial base function, as the Gaussian function, in its hidden layer. Different from MLP networks, RBF networks have only one intermediate layer. The output neurons of the RBF networks are always linear.

In spite of the great applicability of these networks, there are problems in which only using a neural network becomes very costly, due to the need for many modification attempts in the network topology or training parameters or initialization of weights. In these cases, a technique that can be used is the Multiple Neural Networks, since the search for the "ideal network" for any problem can be reduced.

3. Multiple Neural Networks

Multiple neural networks consist of an arrangement of several neural networks, usually of the same type, solving the same problem. The advantage of using multiple ANNs is that a network learns part of the solution and other networks do the same, then the set learns to solve the whole problem.

When using multiple ANNs, there are ways to apply the various networks: combination together, modular combination and a join of the two previous forms. The combination together, also called ensemble, represents a network junction, where each one tries to solve the whole problem. In the combination of the modular type, the problem is divided into parts and each network tries to solve each part separately. There is also the possibility of joining features of the two types of combination, where the problem is divided into sub-problems and several networks try to solve each of these sub-problems created. In this work the ensemble type combination will be used.

However, when using more than one network to solve a problem, it becomes necessary to use a mechanism that makes the networks collaborate, avoiding that the learning of one network is impaired as another Network performs its training. For this, many designers use a coordinator to avoid conflicts in the networks learning.

Regarding to its implementation, a coordinator can pre-sent itself as a logical-mathematical model, as can be seen in the works of Hashem [5] [6] [7], where ANN output is processed to form a single output through an Optimum linear combination and also in the work developed in [13], at which forward selection and backward elimination selective combination methods are used to combine multiple neural networks. But AI techniques can also be used to perform the coordinator's implementation, among them the use of fuzzy systems, as can be seen in [11], genetic algorithm, used in [12], or even another ANN, called gating. It become therefore a further responsibility for the designer to have to design and implement one more element, in addition to ANNs, which can often become a very costly task, and can compromise the work as a whole.

For these reasons, de Almeida Neto [3] developed a form of training in which MLP -type neural networks, using BP as a training algorithm, collaborate without the need to implement a coordinator. To do so, the training procedure consisted in training networks in a sequential way, in which each one must carry out their learning separately, taking advantage of what was learned by previous networks.

Collaboration between networks without a coordinator takes place in the definition of the desired value to be learned by each network. Thus, when being trained, a network has the objective of presenting an output that, when added to the output of the other networks, equals the desired value for the solution of the problem.

In mathematical terms, the error value used in the weights update of the n th network is determined by the desired value for the solution of the problem subtracted from the outputs of all previously trained networks. Eq. 1 presents the formulation of this calculation:

$$e = d - y_1 - y_2 \dots - y_n \quad (1)$$

where e is the n th error of the network, d is the desired value for the solution of the problem, y_1 is the output of the network 1 and so on successively to the network n with output y_n .

In this way, a coordinator is not necessary to determinate the collaboration between the networks, because the learning of the previous network is used by the new network, which starts from this learning and improves it. Because of this, this technique can be characterized as self-coordinated.

Besides that, the existence of systematic collaboration is obvious, because during the training, each network which is added must find an adequate combination of weights to be able to minimize the error, considering what has already been produced by all Networks.

When applying the self-coordinated MNN (Multiple Neural Networks) technique in a real system, the set of inputs must pass through all the networks in the sequence in which they were trained, because as seen, each network uses the output value of the previous networks. In Figure 3 we can see the training scheme of an arrangement with two networks.

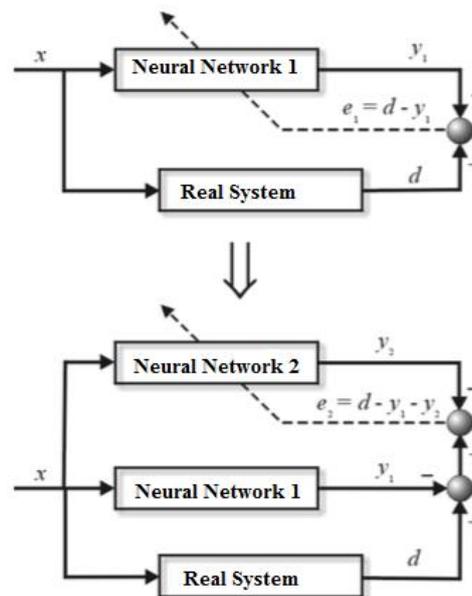


Figure 3. MNN arrangement with two networks.

To ensure that the training of all from the second network must begin exactly from the point where the previous network stopped, initially the output of the new network is zeroed, usually zeroing the previous weights to the output layer of this new network, guaranteeing, as soon as they provide, at first, zero output.

However, in this work an innovation is introduced to the self-coordination mechanism developed by de Almeida Neto [3], combining, however, networks of different types: Multilayer Perceptron MLP and Radial Basis Function (RBF).

4. Methodology

Based on the information treated so far, this work suggests a way to use collaboration between neural networks, aiming at solving a problem. Based on the work developed by de Almeida Neto [3], the collaboration between the networks is done in a self-coordinated way, that is, without the use of a coordinator between the networks.

As already mentioned, two different types of network, the MLP and the RBF, are used, so that the peculiar operation of each one to deal with the problem can contribute to its resolution.

The implementation was performed using MATLAB software. It is an interactive system and a matrix-based programming language, where solutions and problems are expressed in mathematical language. Because it has broad support for mathematical operations, it is possible to perform numerical analysis, operations with matrices, graphing with considerable ease compared to some languages. Although they already have specific libraries for applications with Artificial Neural Networks, in this work they were not used.

The goal is to compare training approaches to learning quality. In order to do this, we verified the success rate of the generalization tests, that is, after the training, it is verified how much the network is able to generalize its learning for data that was not used in the training process.

In this way, the quality of generalization will be evaluated in two different data bases. The first one is the Breast Cancer Wisconsin, which contains 699 events associated with nine attributes. Each of these nine attributes corresponds to the cell characteristics evaluated, such as thickness, uniformity, among others. From the 699 entry patterns, 458 are classified as benign and 241 as malignant. Thus, the purpose of MNN is to map these nine attributes to their respective classes (malignant and benign).

The second database is the Vertebral Column, which contains 310 input patterns, each with six attributes. Each of these six attributes corresponds to the biomechanical characteristics of patients due to the shape and orientation of the pelvis and lumbar spine. These attributes are associated with classification of the spine as normal or abnormal, to determine when they have any associated hernia disorder.

The two data bases are available in the UCI repository and they can be accessed by the address given in [8] and [9].

5. Experiments and Results

In order to evaluate all approaches, both with few neurons in the hidden layer and with many, the two developed programs were performed by varying the amount of neurons in the hidden layer in intervals of seven units, starting in seven up to 105. Thus, 15 different network configurations were generated.

For each number of neurons in the hidden layer, each program was executed 20 times, each execution with initial values of the different weights. In addition, 80% of the data obtained were used for training and 20% for the test and each of the necessary 20 initiations, the data set was changed during training and testing, the results are not dependent on a specific data set. Thus, it was performed in a total of 300 executions (20 initializations x 15 configurations) per program.

To validate the tests performed with the first database, the work done by de Meneses Carvalho [10] was used, where a variation of the Backpropagation algorithm is proposed, through the use of the selective momentum term. In this way, the results obtained by this work regarding the quality of learning are used to validate this methodology.

Figure 4 shows the graph generated in one of the training sessions with the cancer related data base. The training was initialized with a RBF-type network. However, it can be seen that, at a given moment, the network learning, represented by the decrease of the output error, stabilizes. If the methodology were not multiple networks, it would be necessary to start the training again, changing parameters of the neural network or the network structure to enable better learning. However, in the proposed methodology, it is possible to insert a new network that, starting from the previous network learning, will continue the training. As can be seen, the new network entered was MLP.

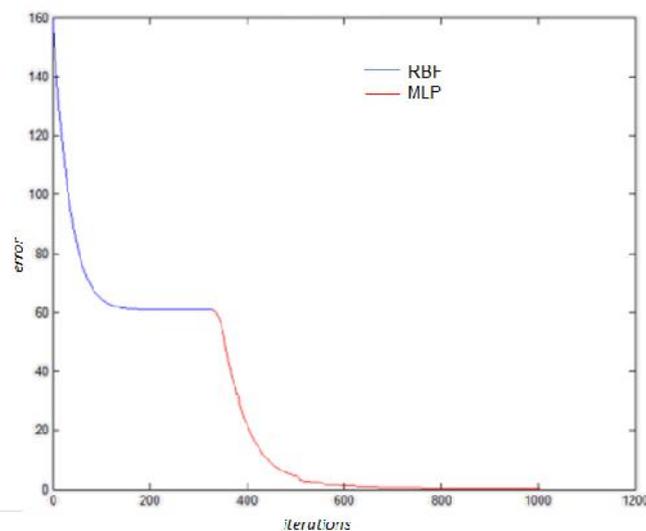


Figure 4. Error Chart

Thus, at the end of program execution, as mentioned earlier, Table 1 was generated. It can be noticed that there was an improvement in the correct results when using the structure of multiple networks.

Table 1. Success rates – Data base 1

Neurons in the occult layer	BP Traditional (%)	BP with Selective Momentum (%)	MNN with MLP and RBF Networks
7	98,04	97,73	98,78
14	98,31	98,09	98,64
21	98,46	98,24	98,48
28	98,36	98,31	98,59
35	98,58	98,21	98,63
42	98,48	98,26	98,67
49	98,46	98,26	98,56

56	98,24	97,92	98,56
63	98,34	97,97	98,52
70	98,43	98,07	98,48
77	98,36	98,02	98,59
84	98,53	98,21	98,52
91	98,46	98,19	98,71
98	98,31	98,26	98,52
105	98,48	98,02	98,71
Average	98,39	98,12	98,60

Table 2 presents the confusion matrix generated in the generalization in which the network had seven neurons in the hidden layer. It is possible to check the values for network performance on correctly classify the standards regardless of the class.

Table 2. Confusion Matrix – Data base 1

		Classification		Total
		Positive	Negative	
Real	Positive	44	0	44
	Negative	2	90	92
Total		46	90	136

Table 3 shows the data obtained for sensitivity and specificity in generalization, considering an initialization and the variation of amount of neurons as used. From this data, it was possible to generate the ROC curve as can be seen in Figure 5.

Table 3. Data for ROC Curve –Data base 1

Neurons in the occult layer	SEN	ESP
7	1,000	0,989
14	0,978	0,989
21	0,978	0,989
28	1,000	0,978
35	1,000	0,989
42	1,000	0,989
49	1,000	0,978
56	1,000	0,989
63	1,000	0,978
70	0,978	0,989
77	1,000	0,989
84	0,947	1,000
91	1,000	0,989
98	0,978	0,989
105	0,973	1,000

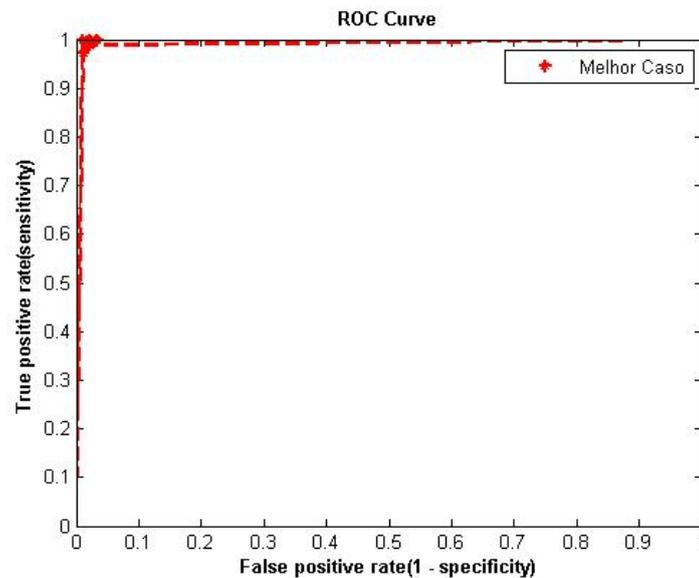


Figure 5. ROC Curve – Data base 1

Table 4 was generated from the execution of the structure of multiple networks using as a database for classification of problems in the spine, as mentioned previously.

Table 4. Success rates – Data base 2

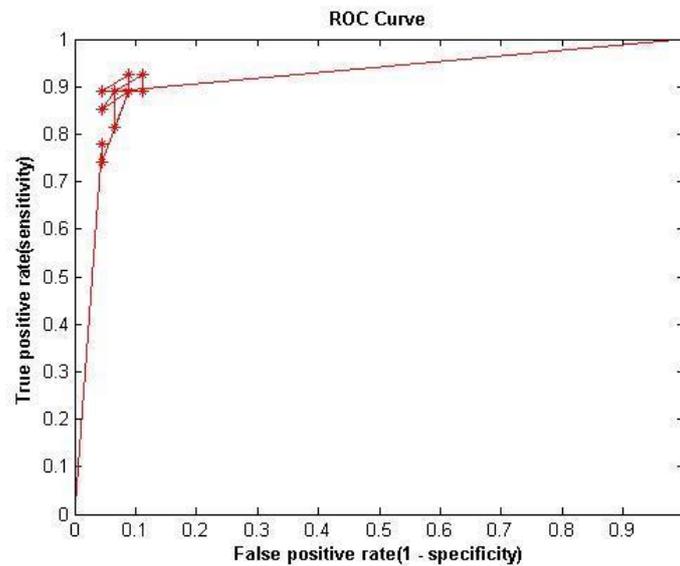
Neurons in the occult layer	MNN with MLP and RBF Networks
7	91,31
14	91,59
21	91,86
28	91,45
35	91,59
42	91,93
49	91,31
56	92,01
63	91,52
70	91,45
77	91,79
84	91,52
91	91,31
98	91,86
105	91,24
Average	91,58

As a way of measuring the performance of the network structure in this database, the work developed in [4] was used, where it was studied, using the same database, the improvement of using classifier committees, where the methodology was the choice of five classifiers out of 50 obtained in each of the training rounds, where these chosen have better hit rates. The type of network used was MLP. In this work, the MLP network committee presented an average of 88,48% accuracy. We can see an improvement in the classification when compared with the results obtained in this work.

Table 5 shows the data obtained for sensitivity and specificity for the second database in the generalization, considering an initialization and the variation of amount of neurons as it was used. From these data, it was possible to generate the ROC curve as can be seen in Figure 6.

Table 5. Data for ROC Curve –Data base 2

Neurons in the occult layer	SEN	ESP
7	0,778	0,956
14	0,741	0,956
21	0,741	0,911
28	0,889	0,911
35	0,815	0,933
42	0,889	0,933
49	0,889	0,956
56	0,926	0,911
63	0,889	0,911
70	0,852	0,956
77	0,889	0,933
84	0,926	0,889
91	0,889	0,889
98	0,889	0,911
105	0,889	0,933

**Figure 6.** ROC Curve – Data base 2

6. Conclusion

The present work proposed the use of an innovative methodology to perform pattern recognition using MLP and RBF type networks. This methodology consists in taking advantage of the learning from the previous network, without the need of a coordinator for this, avoiding that there is a conflict in the networks learning.

The results show that the proposed methodology presents good results when compared to other efficient methodologies. The results obtained in the two databases used present an improvement in the generalization capacity, that is, the capacity to present good results for data not used during the training.

7. Acknowledgment

The authors gratefully acknowledge the support for this research provided by FAPEMA, UEMA, UFMA and IFMA.

8. References

- [1] HAYKIN, Simon S. Artificial neural networks: principle and practice. 2nd Edition, Bookman, Sao Paulo, Brazil, 2000.
- [2] MEHROTRA, Kishan; MOHAN, Chilukuri K.; RANKA, Sanjay. Elements of artificial neural networks. MIT press, 1997.
- [3] ALMEIDA NETO, A. Applications of Multiple Neural Networks in Mechatronic Systems. 2003. Doctoral Thesis. Technological Institute of Aeronautics.
- [4] NETO, A.R. Rocha; BARRETO, G. A. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. IEEE Transactions on Latin America, v. 7, n. 4, p. 487-496, 2009.
- [5] HASHEM, S. et al., Optimal Linear Combinations of Neural Networks: An Overview, IEEE World Congress on Computational Intelligence, pp. 1507, 1512, vol. 3, 1994.
- [6] HASHEM, S.; SCHMEISER, B., Improving Model Accuracy Using Optimal Linear Combinations of Trained Neural Networks, IEEE Transactions on Neural Networks, pp. 792, 794, vol. 6, no. 3, 1995.
- [7] HASHEM, S., Algorithms for Optimal Linear Combinations of Neural Networks, International Conference on Neural Networks, pp. 242, 247, vol. 1, 1997.
- [8] WOLBERG, W. H. Breast Cancer Wisconsin (Original) Data Set. Available in <[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))>.
- [9] MOTA, H. Vertebral Column Data Set. Available in <<https://archive.ics.uci.edu/ml/datasets/Vertebral+Column>>.
- [10] DE MENESES CARVALHO, D. S. Training Acceleration via Backpropagation with selective momentum term. 2016. Masters Dissertation. Federal University of Maranhão.
- [11] CHO, Sung-Bae; KIM, Jin H. Combining multiple neural networks by fuzzy integral for robust classification. IEEE Transactions on Systems, Man, and Cybernetics, v. 25, n. 2, p. 380-384, 1995.
- [12] CHO, Sung-Bae. Fusion of neural networks with fuzzy logic and genetic algorithm. Integrated Computer-Aided Engineering, v. 9, n. 4, p. 363-372, 2002.

APÊNDICE B – ARTIGO TRADUZIDO

Múltiplas Redes Neurais Utilizando Redes MLP e RBF

^{1,2}Aline Porfiro Teixeira, ³Areolino de Almeida Neto

¹Universidade Estadual do Maranhão-Brasil, alineporfirot@gmail.com

²Instituto Federal do Maranhão-Brasil, aline.teixeira@ifma.edu.br

³Universidade Federal do Maranhão-Brasil, areolino@ufma.br

Resumo

Este projeto tem por objetivo realizar uma melhoria no reconhecimento de padrões utilizando Múltiplas Redes Neurais Artificiais de tipos diferentes (MLP e RBF) de forma autocoordenada. Múltiplas redes neurais artificiais são usadas nos casos em que aprender com apenas uma rede é muito difícil. Além disso, para evitar conflitos entre redes, um coordenador é normalmente utilizado. No entanto, este artigo propõe um modelo de múltiplas redes neurais, combinando dois tipos diferentes de redes sem usar um coordenador, mas evitando conflitos. Alguns resultados obtidos são apresentados em duas bases de dados utilizadas e através da comparação com outras metodologias, foi possível verificar uma melhora nos resultados. Portanto, a metodologia proposta é atraente no contexto das Múltiplas Redes Neurais, pois utiliza diferentes modelos de Redes Neurais de forma autocoordenada.

Palavras-chave: Reconhecimento de Padrões, Rede Neural Artificial, Múltiplas Redes Neurais

1. Introdução

Sendo um mecanismo para classificação, o Reconhecimento de Padrões tem sido cada vez mais útil e necessário em sistemas computacionais das mais diversas áreas. Objetivando a classificação de objetos dentro de um número de categorias ou classes, este mecanismo computacional possibilita a criação de uma enorme gama de aplicações para uso científico e tecnológico, como reconhecimento de fala, classificação de documentos, reconhecimento de escrita e reconhecimento de faces, entre outros.

Entre os mais variados métodos utilizados para realizar o Reconhecimento de Padrões, têm-se as Redes Neurais Artificiais (RNA). A partir de um processo de treinamento, uma RNA é capaz de acumular conhecimento e generalizá-lo para outros casos ainda não vistos. Com isso, aprendendo a identificar características dos padrões apresentados no treinamento, a RNA poderá reconhecer outros objetos com as mesmas características.

Porém existem situações em que encontrar a solução utilizando apenas uma rede neural torna-se uma tarefa muito difícil. Funcionando como um aproximador de funções, nestes casos, seriam necessários muitos ajustes na topologia da RNA utilizada para que esta consiga aproximar-se da função da solução durante o aprendizado. Assim, modifica-se a estrutura da RNA ou altera-se alguns parâmetros do treinamento para que se consiga fazer a aproximação desejada.

Em situações como esta, alternativamente é possível utilizar mais de uma rede neural para facilitar o aprendizado, que trabalhando em conjunto passam a ser chamadas de Múltiplas Redes Neurais.

Desta forma, este trabalho pretende apresentar um modelo inovador de múltiplas redes neurais combinando redes de tipos diferentes: Multilayer Perceptron (MLP) e Radial Basis Function (RBF). Além disso, outra inovação é a combinação dá-se sem o uso de um coordenador, mas mesmo assim evitando conflitos entre as redes durante o aprendizado.

2. Redes Neurais Artificiais

Uma rede neural artificial pode ser definida como um processador intenso e paralelamente distribuído, cujas unidades de processamento são bem simples, que têm a propensão natural para armazenar conhecimento e torná-lo disponível para o uso [1].

Desta forma, uma rede neural assemelha-se ao cérebro humano ao adquirir conhecimento através de um processo de aprendizagem e por possuir forças de conexões entre neurônios, conhecidas como pesos sinápticos, utilizadas para armazenar o conhecimento adquirido.

Quanto à sua estrutura, uma RNA é organizada em camadas de neurônios, podendo ter apenas duas camadas, de entrada e saída, ou também, camadas intermediárias (também chamada camada escondida ou oculta). A Figura 1 representa a estrutura básica de uma rede neural com uma camada escondida.

Sendo a unidade básica de processamento de uma rede, um neurônio artificial tem o modelo genérico conforme pode ser visualizado na Figura 2.

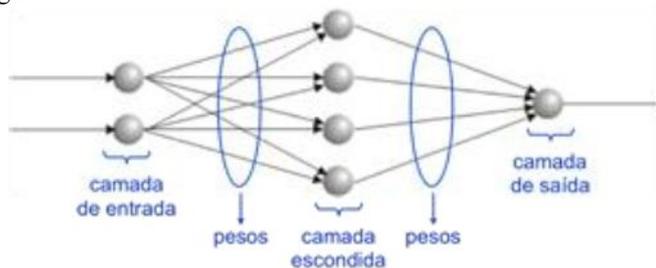


Figura 1. Estrutura da RNA do tipo MLP

Tem-se então a combinação dos dados de entrada representado por “net”, que geralmente é o somatório simples dos valores de entrada. O valor “out” é a saída do neurônio, sendo produzido pela aplicação da função de ativação e que será levado até a entrada de outro neurônio ou para a saída da RNA. Entre as funções de ativação mais utilizadas tem-se a função linear, a função sigmóide ou a função tangente hiperbólica.



Figura 2. Representação de um neurônio artificial.

As RNAs dividem-se em diversos modelos, de acordo com a estrutura da rede, forma de aprendizado e modelo de neurônio. Entre os modelos existentes, destacam-se as redes MLP e RBF:

- **Multilayer Perceptron (MLP):** Criada a partir da adição de uma ou mais camadas intermediárias à rede Perceptron [1], onde a função de ativação é necessariamente não linear, fazendo com que este tipo de rede seja aplicável a problemas independentemente das classes serem linearmente separáveis. Desta forma, uma rede MLP possui uma camada de entrada, pelo menos uma camada intermediária e uma camada de saída. Por ser de simples construção, este tipo de rede é uma das mais utilizadas para desenvolvimentos de soluções aos mais diversos problemas[2]. Existem diversos algoritmos de aprendizado utilizados em uma rede MLP. Um dos mais utilizados é o algoritmo *Backpropagation*, onde o erro da saída é retropropagado para as camadas anteriores com o objetivo de ajuste dos pesos;

- **Radial Basis Function (RBF):** Esta rede possui a característica principal de utilizar como função de ativação uma função de base radial, como a função gaussiana, em sua camada escondida. Ao contrário das redes do tipo MLP, as redes RBF apresentam apenas uma camada intermediária. Os neurônios de saída das redes RBF são sempre lineares.

Apesar da grande aplicabilidade destas redes, existem problemas em que usar apenas uma rede neural se torna muito custoso, em virtude da necessidade de muitas tentativas de modificação na topologia da rede ou nos parâmetros de treinamento ou na inicialização dos pesos. Nestes casos, uma técnica que pode ser utilizada são as Múltiplas Redes Neurais, pois a procura pela “rede ideal” para qualquer problema pode ser reduzida.

3. Múltiplas Redes Neurais

Múltiplas redes neurais consiste em um arranjo de várias redes neurais, normalmente do mesmo tipo, solucionando um mesmo problema. A vantagem de usar múltiplas RNAs é a possibilidade de uma rede aprender parte da solução e assim as demais redes fazem o mesmo, desta forma o conjunto aprende a resolver o problema todo.

Ao usar múltiplas RNAs, existem formas de empregar as várias redes: combinação em conjunto, combinação modular e uma junção das duas formas anteriores. A combinação em conjunto, também chamada de ensemble, representa uma junção de redes, onde cada uma tenta resolver o problema inteiro. Já na combinação do tipo modular, o problema é dividido em partes e cada rede tentará resolver cada parte isoladamente. Há também a possibilidade de unir características dos dois tipos de combinação, onde o problema é dividido em sub-problemas e várias redes tentam resolver cada um desses sub-problemas criados. Neste trabalho será utilizada a combinação do tipo ensemble.

Porém, ao se utilizar mais de uma rede na resolução de um problema, torna necessária a utilização de um mecanismo que faça com que as redes colaborem entre si, evitando que o aprendizado de uma rede seja prejudicado a medida que uma outra rede efetua seu treinamento. Para isto, muitos projetistas utilizam um coordenador para evitar conflitos no aprendizado das redes.

Quanto à sua implementação, um coordenador pode se apresentar como um modelo lógico-matemático, como pode ser visto nos trabalhos de Hashem [5] [6] [7], onde a saída das RNA é processada para formarem uma única saída através de uma combinação linear ótima. Mas também pode-se empregar técnicas de IA para realizar a implementação do coordenador, entre elas tem-se o uso de sistemas *fuzzy*, como pode ser visto em [11], algoritmo genético, utilizado em [12], ou mesmo outra RNA, chamada de *gating*. Toma-se, portanto, uma responsabilidade a mais ao projetista ter que projetar e implementar mais um elemento, além das RNAs, que pode muitas vezes se tornar uma tarefa muito custosa, podendo comprometer o trabalho como um todo.

Por esses motivos, de Almeida Neto [3] desenvolveu uma forma de treinamento em que redes neurais do tipo MLP, utilizando o BP como algoritmo de treinamento, colaborem sem a necessidade de implementar um coordenador. Para tanto, o procedimento de treinamento consistiu em treinar redes de forma sequencial, em que cada uma deve realizar o seu aprendizado separadamente, aproveitando o que foi aprendido pelas redes anteriores.

A colaboração entre as redes sem um coordenador acontece na definição do valor desejado a ser aprendido por cada rede. Assim, ao ser treinada, uma rede tem como objetivo apresentar uma saída que, ao ser somada com a saída das outras redes, iguale ao valor desejado para a solução do problema.

Em termos matemáticos, o valor de erro utilizado no cálculo da atualização dos pesos da n-ésima rede é determinado pelo valor desejado para a solução do problema subtraído das saídas de todas as redes já anteriormente treinadas. A Eq. 1 apresenta a formulação deste cálculo:

$$e = d - y_1 - y_2 \dots - y_n \quad (1)$$

sendo e o erro da n-ésima rede, d o valor desejado para a solução do problema, y_1 a saída da rede 1 e assim sucessivamente até a rede n com saída y_n .

Deste modo, não é necessário um coordenador para determinar a colaboração entre as redes, isto porque o aprendizado da rede anterior é aproveitado pela nova rede, que parte deste aprendizado e o aprimora. Por conta disto, esta técnica pode ser caracterizada como autocoordenada.

Além disso, a existência de colaboração sistemática é evidente, pois ao longo do treinamento, cada rede que é adicionada deve achar uma combinação adequada de pesos para conseguir minimizar o erro, levando em consideração o que já foi produzido por todas as redes anteriores.

Ao aplicar a técnica das MRN (Múltiplas Redes Neurais) autocoordenadas em um sistema real, o conjunto de entradas deverá perpassar todas as redes, na sequência em que foram treinadas, pois como visto, cada rede utiliza o valor de saída das redes anteriores. Na Figura 3 pode-se ver o esquema do treinamento de um arranjo com duas redes.

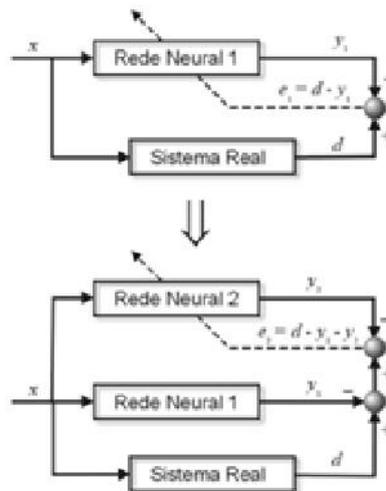


Figura 3. Arranjo de MRNA com duas redes.

Para garantir que o treinamento de todas a partir da segunda rede deve começar exatamente do ponto onde a rede anterior parou, inicialmente, zera-se a saída da nova rede, normalmente zerando os pesos anteriores a camada de saída dessa nova rede, garantindo assim que elas providenciem, em um primeiro momento, saída nula.

Todavia, neste trabalho uma inovação é introduzida ao mecanismo de autocoordenação desenvolvido por de Almeida Neto [3], combinando-se, porém, redes de tipos diferentes: *Multilayer Perceptron* (MLP) e *Radial Basis Function* (RBF).

4. Metodologia

Baseado nas informações tratadas até aqui, este trabalho sugere uma forma de utilizar a colaboração entre redes neurais, visando a resolução de um problema. Tendo como base o trabalho desenvolvido por de Almeida Neto [3], a colaboração entre as redes é feita de forma autocoordenada, ou seja, sem a utilização de um coordenador entre as redes.

Como já citado, utiliza-se dois tipos diferentes de rede, a MLP e a RBF, de modo que o funcionamento peculiar de cada uma tratar o problema possa contribuir para a resolução do mesmo.

A implementação foi realizada utilizando o software MATLAB. Trata-se de um sistema interativo e uma linguagem de programação baseados em matrizes, onde as soluções e problemas são expressos em linguagem matemática. Por possuir amplo suporte à operações matemáticas, é possível realizar análise numérica, operações com matrizes, construção de gráficos com considerável facilidade comparada à algumas linguagens. Apesar de já possuir bibliotecas específicas para aplicações com Redes Neurais Artificiais, neste trabalho elas não foram utilizadas.

O objetivo é comparar as abordagens de treinamento quanto à qualidade do aprendizado. Para tanto, é verificada a taxa de acerto dos testes de generalização, ou seja, após o treinamento, é verificado o quanto a rede consegue generalizar seu aprendizado para dados que não foram utilizados no processo de treinamento.

Desta forma, será avaliada a qualidade de generalização em duas bases de dados diferentes. A primeira vem a ser a *Breast Cancer Wisconsin*, que contém 699 eventos associados a nove atributos. Cada um desses nove atributos correspondem à características de células avaliadas, como espessura, uniformidade, entre outras. Dos 699 padrões de entrada, 458 são classificados como benignos e 241 como malignos. Deste modo, o objetivo da MRNA é mapear esses nove atributos às suas respectivas classes (malignos e benignos).

A segunda base de dados trata-se da *Vertebral Column*, que contém 310 padrões de entrada, possuindo seis atributos cada. Cada um desses seis atributos correspondem às características biomecânicas de pacientes decorrentes da forma e orientação da pelve e coluna lombar. Esses atributos estão associados à classificação do coluna vertebral como normal ou anormal, para determinar quando apresentam algum distúrbio associado à hérnia.

As duas bases de dados estão disponíveis no repositório UCI e podem ser acessadas pelo endereço disposto em [8] e [9].

5. Resultados e Experimentos

Com o objetivo de avaliar todas as abordagens, tanto com poucos neurônios na camada escondida, quanto com muitos, os dois programas desenvolvidos foram executados variando-se a quantidade de neurônios na camada escondida em intervalos de sete unidades, começando em sete e indo até 105. Logo, foram geradas 15 configurações distintas de rede.

Para cada quantidade de neurônios na camada escondida, cada programa foi executado 20 vezes, cada execução com valores iniciais dos pesos diferentes. Além disso, 80% dos dados disponíveis foram utilizados para treinamento e 20% para teste e a cada quatro inicializações destas 20, mudou-se o conjunto de dados utilizados durante o treinamento e teste, fazendo com que os dados utilizados no treinamento sejam alternados para que os resultados não sejam dependentes de um conjunto de dados específico. Assim, foi realizado um total de 300 execuções (20 inicializações x 15 configurações) por programa.

Para validar os testes realizados com a primeira base de dados, utilizou-se o trabalho realizado por de Meneses Carvalho[10], onde é proposta uma variação do algoritmo *Backpropagation*, através da utilização do termo *momentum* seletivo. Desta forma, os resultados obtidos por esse trabalho quanto à qualidade do aprendizado são utilizados para validar esta metodologia.

A Figura 4 ilustra o gráfico gerado em um dos treinamentos com a base de dados relacionada ao câncer. O treinamento foi inicializado com uma rede do tipo RBF, percebe-se, no entanto, que em dado momento, o aprendizado da rede, representado pela diminuição do erro de saída, estabiliza. Caso a metodologia não fosse de múltiplas redes, seria necessário iniciar o treinamento novamente, alterando parâmetros da rede neural ou a estrutura da rede para possibilitar melhor aprendizado. Porém, na metodologia proposta, é possível inserir uma nova rede que, partindo do aprendizado da rede anterior, irá dar continuidade ao treinamento. Como pode ser verificado, a nova rede inserida foi MLP.

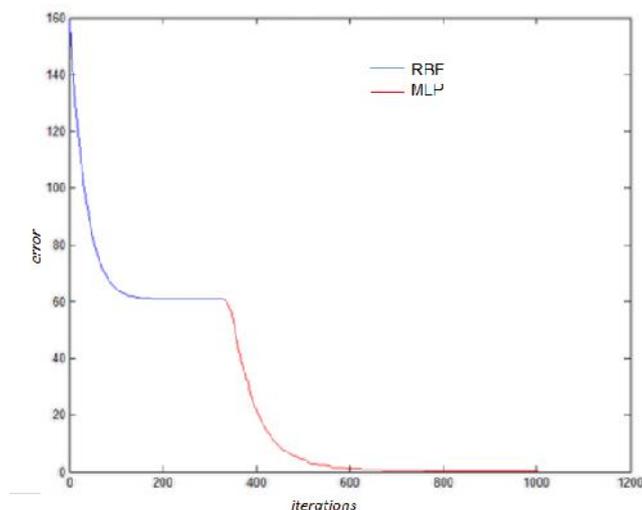


Figura 4. Gráfico do Erro Quadrático Médio.

Assim, ao fim da execução do programa, conforme mencionado anteriormente, a Tabela 1 foi gerada. Percebe-se que houve uma melhora nos resultados de acerto quando utilizada a estrutura de múltiplas redes.

Tabela 1. Taxas de acerto – Base de Dados 1

Neurônios na camada escondida	BP Tradicional (%)	BP com Momentum Seletivo (%)	MRN com redes MLP e RBF (%)
7	98,04	97,73	98,78
14	98,31	98,09	98,64
21	98,46	98,24	98,48
28	98,36	98,31	98,59

35	98,58	98,21	98,63
42	98,48	98,26	98,67
49	98,46	98,26	98,56
56	98,24	97,92	98,56
63	98,34	97,97	98,52
70	98,43	98,07	98,48
77	98,36	98,02	98,59
84	98,53	98,21	98,52
91	98,46	98,19	98,71
98	98,31	98,26	98,52
105	98,48	98,02	98,71
Média	98,39	98,12	98,60

A Tabela 2 apresenta a matriz de confusão gerada na generalização em que a rede possuía sete neurônios na camada escondida. É possível verificar valores quanto ao desempenho da rede em classificar corretamente os padrões independentemente da classe.

Table 2. Matriz de Confusão – Base de Dados 1

		Classificação		Total
		Positivo	Negativo	
Real	Positivo	44	0	44
	Negativo	2	90	92
Total		46	90	136

A Tabela 3 apresenta os dados obtidos para sensibilidade e especificidade na generalização, considerando uma inicialização e a variação de quantidade de neurônios conforme foi utilizada. A partir desses dados, foi possível gerar a curva ROC conforme pode ser visto na Figura 5.

Tabela 3. Dados para curva ROC – Base de Dados 1

Neurônios na camada escondida	SEN	ESP
7	1,000	0,989
14	0,978	0,989
21	0,978	0,989
28	1,000	0,978
35	1,000	0,989
42	1,000	0,989
49	1,000	0,978
56	1,000	0,989
63	1,000	0,978
70	0,978	0,989
77	1,000	0,989
84	0,947	1,000
91	1,000	0,989
98	0,978	0,989
105	0,973	1,000

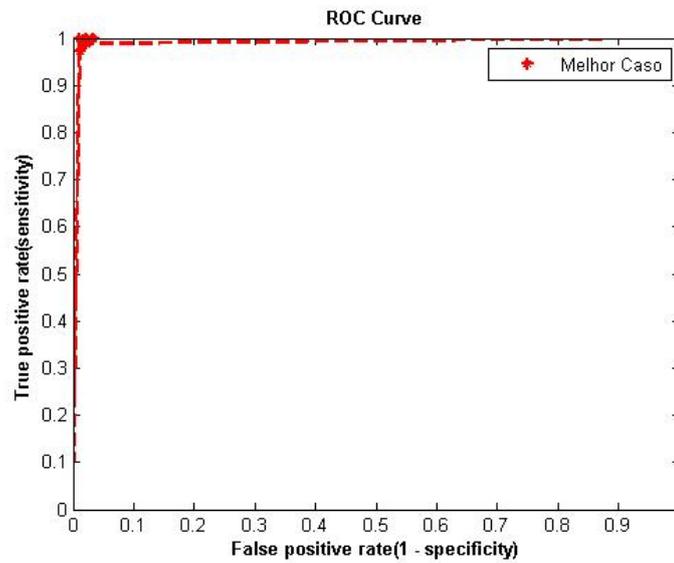


Figura 5. Curva ROC – Base de Dados 1

A Tabela 4 foi gerada a partir da execução da estrutura de múltiplas redes utilizando como base de dados para classificação de problemas na coluna vertebral, como foi mencionado anteriormente.

Tabela 4. Taxas de acerto – Base de Dados 2

Neurônios na camada escondida	MRN com redes MLP e RBF (%)
7	91,31
14	91,59
21	91,86
28	91,45
35	91,59
42	91,93
49	91,31
56	92,01
63	91,52
70	91,45
77	91,79
84	91,52
91	91,31
98	91,86
105	91,24
Média	91,58

Como forma de mensurar o desempenho da estrutura de redes nesta base de dados, utilizou-se o trabalho desenvolvido em [4] onde foi estudada, utilizando a mesma base de dados, a melhoria de se utilizar comitês de classificadores, onde a metodologia foi a escolha de cinco classificadores entre 50 obtidos em cada uma das rodadas de treino, onde estes escolhidos possuem melhores taxas de acerto. O tipo de rede utilizado foi MLP. Neste trabalho, o comitê de redes MLP apresentou média de acerto de 88,48%. Percebe-se uma melhora na classificação quando se compara com os resultados obtidos neste trabalho.

A Tabela 5 apresenta os dados obtidos para sensibilidade e especificidade para a segunda base de dados na generalização, considerando uma inicialização e a variação de quantidade de neurônios conforme foi utilizada. A partir desses dados, foi possível gerar a curva ROC conforme pode ser visto na Figura 6.

Tabela 5. Dados para curva ROC –Base de Dados 2

Neurônios na camada escondida	SEN	ESP
7	0,778	0,956
14	0,741	0,956
21	0,741	0,911
28	0,889	0,911
35	0,815	0,933
42	0,889	0,933
49	0,889	0,956
56	0,926	0,911
63	0,889	0,911
70	0,852	0,956
77	0,889	0,933
84	0,926	0,889
91	0,889	0,889
98	0,889	0,911
105	0,889	0,933

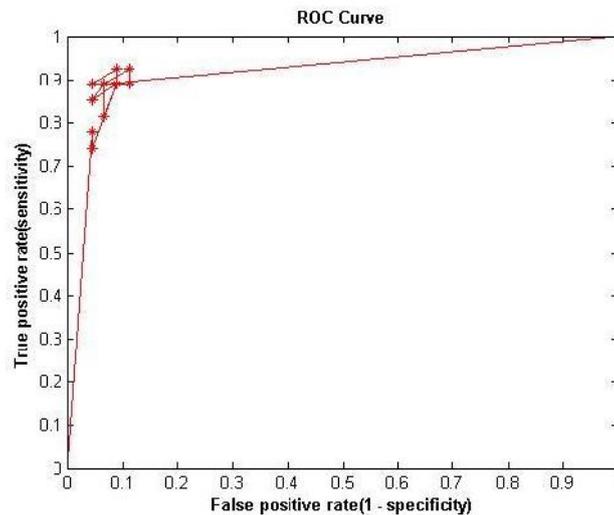


Figura 6. Curva ROC – Base de Dados 2

6. Conclusão

O presente trabalho propôs a utilização de uma metodologia inovadora para efetuar o reconhecimento de padrões utilizando redes do tipo MLP e RBF. Esta metodologia consiste no aproveitamento do aprendizado da rede anterior, sem a necessidade de um coordenador para isto, evitando que haja conflito no aprendizado das redes.

Os resultados mostram que a metodologia proposta apresenta bons resultados quando comparados a outras metodologias também eficientes. Os resultados obtidos nas duas bases de dados utilizadas apresentam uma melhora quanto à capacidade de generalização, ou seja, à capacidade de apresentar bons resultados para dados não utilizados durante o treinamento.

7. Agradecimento

Os autores agradecem o apoio a esta pesquisa à FAPEMA, UEMA, UFMA e IFMA.

8. Referências

- [1] HAYKIN, Simon S. Redes neurais artificiais: princípio e prática. 2ª Edição, Bookman, São Paulo, Brasil, 2000.
- [2] MEHROTRA, Kishan; MOHAN, Chilukuri K.; RANKA, Sanjay. Elements of artificial neural networks. MIT press, 1997.
- [3] ALMEIDA NETO, A. Aplicações de Múltiplas Redes Neurais em Sistemas Mecatrônicos. 2003. Tese de Doutorado. Instituto Tecnológico de Aeronáutica.
- [4] NETO, A.R. Rocha; BARRETO, G. A. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. IEEE Transactions on Latin America, v. 7, n. 4, p. 487-496, 2009.
- [5] HASHEM, S. et al., Optimal Linear Combinations of Neural Networks: An Overview, IEEE World Congress on Computational Intelligence, pp. 1507, 1512, vol. 3, 1994.
- [6] HASHEM, S.; SCHMEISER, B., Improving Model Accuracy Using Optimal Linear Combinations of Trained Neural Networks, IEEE Transactions on Neural Networks, pp. 792, 794, vol. 6, no. 3, 1995.
- [7] HASHEM, S., Algorithms for Optimal Linear Combinations of Neural Networks, International Conference on Neural Networks, pp. 242, 247, vol. 1, 1997.
- [8] WOLBERG, W. H. Breast Cancer Wisconsin (Original) Data Set. Disponível em <[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))>.
- [9] MOTA, H. Vertebral Column Data Set. Disponível em <<https://archive.ics.uci.edu/ml/datasets/Vertebral+Column>>.
- [10] DE MENESES CARVALHO, D. S.. Aceleração do Treinamento via Backpropagation com termo momentum seletivo. 2016. Dissertação de Mestrado. Universidade Federal do Maranhão.
- [11] CHO, Sung-Bae; KIM, Jin H. Combining multiple neural networks by fuzzy integral for robust classification. IEEE Transactions on Systems, Man, and Cybernetics, v. 25, n. 2, p. 380-384, 1995.
- [12] CHO, Sung-Bae. Fusion of neural networks with fuzzy logic and genetic algorithm. Integrated Computer-Aided Engineering, v. 9, n. 4, p. 363-372, 2002.