

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

ADRIANO DOS SANTOS RÊGO

**PREDIÇÃO DE PARÂMETROS DE QUALIDADE DO BIODIESEL
UTILIZANDO REDES NEURAS ARTIFICIAIS**

São Luís
2013

ADRIANO DOS SANTOS RÉGO

**PREDIÇÃO DE PARÂMETROS DE QUALIDADE DO BIODIESEL
UTILIZANDO REDES NEURAS ARTIFICIAIS**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão para obtenção do título de Mestre em Engenharia de Eletricidade, área de Concentração Ciência da Computação.

Orientador: Dr. Sofiane Labidi

São Luís

2013

Rêgo, Adriano dos Santos

Predição de parâmetros de qualidade do biodiesel utilizando redes neurais artificiais / Adriano dos Santos Rêgo. – São Luís, 2013.
119f.

Orientador: Prof. Dr.Sofiane Labidi

Dissertação (Mestrado) – Programa de Pós Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, 2013.

1. Predição. 2. Biodiesel. 3. Redes Neurais Artificiais. 4. Sys-PANN.. I. Título.

CDU 662.756.3: 004.8

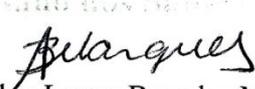
PREDIÇÃO DE PARÂMETROS DE QUALIDADE DO BIODIESEL UTILIZANDO REDES NEURAIS ARTIFICIAIS

Adriano dos Santos Rêgo

Dissertação aprovada em 11 de dezembro de 2013.



Prof. Sofiane Labidi, Dr.
(Orientador)



Prof. Aldaíza Lopes Brandes Marques, Dra.
(Membro da Banca Examinadora)



Prof. Carlos Alberto de Francisco, Dr.
(Membro da Banca Examinadora)



Prof. Zair Abdelouahab, Ph.D.
(Membro da Banca Examinadora)

Aos meus pais e conselheiros Nilson e Tereza, minhas irmãs, Viviane e Leiliane, e a minha namorada e companheira, Ana Carolina.

AGRADECIMENTOS

Em primeiro lugar, a Deus, Jesus e Maria, pela minha vida que foi me concedida, abençoada e protegida além da inteligência, calma, perseverança e energia, que são muito úteis para os desafios diários da vida, em especial para realizar e concluir este trabalho.

A meus pais, Nilson Cristino e Raimunda Tereza, as duas pessoas que me criaram e moldaram a pessoa que sou hoje. Que sempre estiveram comigo e me apoiaram em tudo. A todo esforço e sacrifícios que eles tiveram para que eu pudesse chegar até aqui. Sem eles eu não teria chegado onde estou, não seria o que sou. Devo tudo a eles.

As minhas irmãs, amigas e conselheiras, Viviane e Leiliane por sempre ter me apoiado e ajudado, pelos conselhos e preocupações, a quem eu sempre posso contar para o que precisar. E pela confiança passada.

A minha namorada Ana Carolina, por todo apoio e confiança passada a mim em suas conversas. Por nestes anos do mestrado, ter entrado em minha vida e proporcionado grandes alegrias e felicidades, necessárias para ficar tranquilo e desenvolver este trabalho.

A Meus Tios, Ana Lourdes, Socorro, Naldo e Vânia, que sempre acreditaram e me apoiaram na vida. A eles, que sempre estiverem presentes na alegria e tristeza. Em especial a Lourdes.

A meus avôs, Pedro e Joana, que sempre torceram pelo meu sucesso.

Aos companheiros do LSI e LPQA que me acompanharam no desenvolvimento deste trabalho. Em especial a Alex que idealizou este trabalho e sem ele, este trabalho não teria sido finalizado, e a Isabelle, por ter sido minha conselheira e fonte informações na área de Biodiesel, a vocês dois, meu sincero Obrigado.

À Nilson Costa, por ter me orientado e tido toda paciência comigo nesta dissertação. Obrigado por, desde a graduação, ter sempre acreditado em meu potencial e se hoje finalizo o mestrado, é pelo seu incentivo constante.

Aos meus amigos do NEAD, pela torcida e alegrias proporcionadas, e em especial a Leonardo, que sempre acredita em meu potencial.

A todos os meus parentes e amigos pelo carinho, apoio e pela torcida incondicional.

A meu orientador, Sofiani Labidi, pela chance dada.

Ao PRH-ANP em parceria com a CAPES pelo financiamento de minha pesquisa.

E a todos aqueles que diretamente ou indiretamente contribuíram para a realização deste trabalho.

“Evoluir é compreender que a distância entre os sonhos e a realidade, é apenas a sua força de vontade.”

desconhecido

RESUMO

A predição é uma ação para tomada de decisão. Quanto menor for o erro da predição, menor será o prejuízo pela decisão tomada. Uma técnica que vem sendo muito utilizada com sucesso para predição é a Rede Neural Artificial (RNA). Ela é uma ferramenta computacional, inspirado no cérebro humano e que possui uma grande capacidade de modelar problemas complexos do mundo real através de uma fase chamada de treinamento que possibilita a ela aprender as características do problema proposto. As RNAs têm o potencial de descobrir relações complexas entre os dados de entrada e a resposta por isso elas se tornaram muito úteis na área da predição. Pelo ótimo desempenho nas mais diversas áreas de conhecimento, decidiu-se verificar a sua eficácia no Biodiesel. Ele é um combustível renovável gerado a partir de óleos de origem animal ou vegetal para ser utilizado em motores a diesel. Ele possui diversos benefícios sobre diesel mineral e no momento, sua produção e utilização recebe grandes estímulos do governo federal, através do Programa Nacional de Produção e Uso do Biodiesel (PNPB). Com isso, o interesse pela pesquisa de Biodiesel cresceu no Estado do Maranhão. A verificação de qualidade do Biodiesel é realizada pelo Laboratório de Pesquisas Química e Analítica (LPQA/UFMA). Verificou-se, então que suas pesquisas podem ser aceleradas se as análises de qualidade do biodiesel que são influenciadas pela composição química forem substituídas por uma única análise de composição química. Esta composição foi utilizada para prever os resultados dessas outras análises de qualidade. Para isso, desenvolveu-se o Sys-PANN, um software capaz de encontrar a melhor RNA para cada tipo de análise e prever seus resultados.

Palavras-Chave: Predição, Biodiesel, Redes Neurais Artificiais, Sys-PANN.

ABSTRACT

The prediction is an action for decisions making. The smaller the error of the prediction, the lower the loss by decision taken. One technique that has been used very successfully to predict is the Artificial Neural Network (ANN). It is a computational tool, inspired by the human brain and has a great ability to model complex real-world problems through a phase called training enables it to learn the characteristics of the proposed problem. The ANNs have the potential to discover complex relations between input data and response so they become very useful in the prediction area. For optimal performance in various areas of knowledge, it was decided to assess its efficiency in a sector, the Biodiesel. It is a renewable fuel generated from oils of animal or vegetable to be used in diesel engines. It has several benefits over mineral diesel and at the time, their production and use gets great stimuli federal government, through the National Program for Production and Use of Biodiesel. Thus, interest in research grew and Biodiesel in the state of Maranhão is performed by the Laboratory of Chemical Research & Analytics (LPQA / UFMA). It was then that their research can be accelerated if the analyzes of quality biodiesel that are influenced by the chemical composition are replaced by a single analysis of chemical composition and this composition is used to predict the results of other analyzes of quality. For this, was developed the Sys-PANN, software able to find the best ANN for each type of analysis and predict your results.

Keywords: Prediction, Biodiesel, Artificial Neural Networks, Sys-PANN.

SUMÁRIO

LISTA DE FIGURAS	14
LISTA DE TABELAS	17
LISTA DE SIGLAS	18
1. INTRODUÇÃO.....	19
2. BODIESEL.....	23
2.1. Estrutura Química do Biodiesel	24
2.2. Produção de Biodiesel.....	25
2.3. O PNPB e os Parametros de Qualidade de Biodiesel	28
2.4. Considerações Finais	29
3. REDES NEURAIS ARTIFICIAIS.....	31
3.1. Inspiração das Redes Biológicas.....	32
3.2. Histórico	33
3.3. O Neurônio Artificial.....	34
3.4. A Função de Ativação	36
3.5. Topologia das RNAs.....	38
3.6. Modelos de RNA	40
3.7. Aprendizagem de RNA.....	42
3.7.1. Treinamento Supervisionado	44
3.7.2. Treinamento Não-Supervisionado.....	46
3.8. Aplicações da RNA.....	48
3.8.1. Associação de Padrões.....	48
3.8.2. Reconhecimento de Padrões	49
3.8.3. Aproximação de Funções.....	50
3.8.4. Recuperação e Filtragem de Informação	51

3.9.	Desenvolvimento de um projeto de RNA.....	51
3.9.1.	Definição e Configuração da RNA	52
3.9.2.	Preparação dos Dados.....	52
3.9.3.	Treinamento da RNA.....	53
3.9.4.	Teste da RNA.....	53
3.9.5.	Aplicação.....	53
3.10.	Considerações Finais.....	54
4.	REDE PERCEPTRON DE MÚLTIPLAS CAMADAS (MLP).....	55
4.1.	Características da Rede MLP	56
4.2.	Treinamento BackPropagation (BP).....	58
4.2.1.	Descrição do Algoritmo BackPropagation	59
4.2.2.	BackPropagation com <i>momentum</i>	63
4.2.3.	Treinamento Resilient Propagation	63
4.2.4.	Treinamento Levenberg-Marquardt(LM)	65
4.3.	Configurando uma MLP.....	66
4.4.	Considerações finais	70
5.	SISTEMA PROPOSTO.....	72
5.1.	Software auxiliar OleoData.....	74
5.1.1.	Modelagem do OleoData	75
5.1.2.	Implementação do OleoData.....	76
5.2.	Sys-PANN – Módulo de treinamento.....	82
5.2.1.	Definição e Configuração da RNA	83
5.2.2.	Preparação dos Dados.....	86
5.3.	Sys-PANN – Módulo de Predição.....	97
5.4.	Considerações Finais	99
6.	RESULTADOS E DISCUSSÕES.....	100

7. CONSIDERAÇÕES FINAIS.....	108
REFERÊNCIA	111
ANEXO I - Resolução nº. 14 da ANP - Especificação do Biodiesel	117

LISTA DE FIGURAS

Figura 2.1 - Reação de transesterificação reversível (GERIS <i>et al</i> , 2007).	25
Figura 2.2 - Fluxograma do Processo de Produção de Biodiesel (PARENTE, 2003)	26
Figura 3.1-Exemplo de arquitetura de uma Rede Neural Artificial [Adaptado de (Haykin, 2001)].	31
Figura 3.2 - Representação do neurônio biológico. [Adaptado de ARBIB, 2002]	33
Figura 3.3- Modelo básico de um Neurônio artificial (Fonte: Haykin,2001)	35
Figura 3.4- Gráfico da função limiar	37
Figura 3.5- Gráfico da função Sigmoide	37
Figura 3.6- Gráfico da função hiperbólica-tangente	38
Figura 3.7- Rede Feed-Forward	39
Figura 3.8 - Rede Feed-Forward	39
Figura 3.9- Modelos de Redes Neurais por Topologia	42
Figura 3.10- Aprendizado Supervisionado	45
Figura 3.11-Aprendizado Não-Supervisionado	47
Figura 3.12- Modelo de uma Auto-Associação.....	49
Figura 3.13- Modelo de uma Hetero-Associação	49
Figura 3.14 - Modelo de Reconhecimento de Padrão	50
Figura 3.15 - Modelo de Apróximação de Função.....	50
Figura 3.16 - Fluxograma de Desenvolvimento de RNA	51
Figura 4.1- Classes de problemas: a) linearmente separáveis e b) não-linearmente separáveis	55
Figura 4.2- Arquitetura da Rede MLP(adaptada de Haykin, 2001).....	56
Figura 4.3- Fases e fluxo de dados em um treinamento Backpropagation	58

Figura 4.4-Pseudocódigo do backpropagation	62
Figura 4.5- Gráfico representando a condição de parada	70
Figura 5.1- Arquitetura geral do Sistema.....	73
Figura 5.2- Arquitetura do módulo de predição do Sys-PANN	74
Figura 5.3-Diagrama de Caso de uso do OleoData na visãodo analista.....	75
Figura 5.4-Tela Inicial do OleoData	77
Figura 5. 5- Lista de Biodiesel cadastrados no OleoData	78
Figura 5.6- Telas de cadastro de Biodiesel e geração de ID automático ao Biodiesel no OleoData.....	79
Figura 5.7- Tela de Cadastro de Amostras de Biodiesel no OleoData	80
Figura 5.8- Tela de Relatório de Biodiesel no Oleodata.....	81
Figura 5.9-Diagrama de Sequência do modulo de treinamento do Sys-PANN	82
Figura 5.10- Código de implementação no encog das RNA	89
Figura 5.11- Diagrama de Classes da RNA	90
Figura 5.12-XML da configuração de uma RNA.....	91
Figura 5.13- Gráfico de teste de validação cruzada	93
Figura 5.14- Tela inicial do Sys-PANN	94
Figura 5.15-Modulo de treinamento do Sys-PANN	95
Figura 5.16-Treino Finalizado no Sys-PANN.....	96
Figura 5.17- Módulo de predição do Sys-PANN.....	98
Figura 6.1- Gráfico de dispersão do Sys-PANN	101
Figura 6.2- Gráfico de dispersão do Statistica.....	101
Figura 6.3 - Gráfico de dispersão do grupo de validação do Índice de íodo ..	104
Figura 6.4 - Gráfico de dispersão do grupo de treinamento do Índice de íodo	104
Figura 6.5- Gráfico de dispersão do grupo de validação da Massa Específica	105

Figura 6.6- Gráfico de dispersão do grupo de treinamento da Massa Específica	105
Figura 6.7 - Gráfico de dispersão do grupo de validação da Densidade	106
Figura 6.8 - Gráfico de dispersão do grupo de treinamento da Densidade	106

LISTA DE TABELAS

Tabela 2.1 - Exemplo de disposição dos ésteres no biodiesel (adaptado de Melo <i>et al</i> ,2010 e Santos <i>et al</i> , 2007).....	24
Tabela 5.1 - Comparativo de Algoritmos: BP, LMA e RProp.....	92
Tabela 6.1- Configurações dos Softwares Sys-PANN e Statistica.....	100
Tabela 6.2- Valores de predição de Índice de Iodo.....	102
Tabela 6.3 - Melhores configurações de RNA encontrada para os 3 parâmetros de qualidade.....	103

LISTA DE SIGLAS

ANP - Agência Nacional do Petróleo

ART - Teoria da Ressonância Adaptativa

BP - Treinamento BackPropagation

IA - Inteligência Artificial

LAPQAP/UFMA - Laboratório de Análise e Pesquisa em Química Analítica do Petróleo da UFMA

LM - Levenberg-Marquardt

LPQA/UFMA - Laboratório de Pesquisa em Química Analítica

MCP - Neurônio McCulloch e Pitts

MLP - Perceptron Multi-Camadas/ Multi-Layer Perceptron

MSE - Erro médio quadrático

PNPB - Programa Nacional de Produção e Uso do Biodiesel

RBF - Função de Base Radial

RNA - Rede Neural Artificial

RPROP - Treinamento Resilient Propagation

Sys-PANN – System of Prediction by Artificial Neural Networks

UFMA - Universidade Federal do Maranhão

1. INTRODUÇÃO

A busca pela predição e previsão é um desejo antigo da sociedade. Eles são importantes para as tomadas decisões. Quanto menor for o erro da predição ou previsão, menor será o prejuízo pela decisão tomada.

De acordo com a enciclopédia Infopédia (2013), etmológicamente, predição significa: "dizer ou anunciar com antecedência o que vai acontecer, seja pelo cálculo ou pela previsão."

Pórem, este termo é muito empregado na literatura científica e confundido como o termo previsão, pois eles recebem significados mais abrangentes. A palavra previsão sugere que se quer ver alguma coisa, antes que ela exista. Alguns autores preferem a palavra predição, para indicar algo que deverá existir no futuro (Morettin & Tolo, 2004).

A previsão se baseia no estudo de observações consecutivas dependentes uma das outras para se antecipar um futuro (Samohyl, Rocha & Mattos, 2001) enquanto a predição necessita de dados atuais. Assim, A previsão necessita de dados históricos para descobrir se algo vai existir e a predição indica como algo que existe irá se comportar a partir de dados atuais.

Por isso, muitos trabalhos vem sendo desenvolvidos para se conseguir os melhores métodos de predição, com o intuito de diminuição de risco na tomada de decisão, a partir de poucos dados, sem a necessidade de grande investigação do fato.

Atualmente, o emprego de Redes Neurais Artificiais (RNA) vem sendo utilizado para predição com grande entusiasmo pela comunidade científica devido a resposta que se retorna com baixo erro, facilitando a tomada de decisão.

A RNA é uma ferramenta computacional, inspirado no cérebro humano e possui a capacidade de modelar problemas complexos do mundo real (Basheer, 2000). As redes neurais possuem uma fase chamada de treinamento que possibilita a ela aprender as características do problema proposto e então modela-lo.

Após aprenderem as características do problema, são capazes de responder a novas características daquela natureza a que ela é submetida. Assim, ela trabalha sempre com entradas e saídas, no qual a entrada são as características que se tem e a saída, o resultado esperado.

Essas redes possuem o potencial de descobrir relações complexas entre os dados de entrada e a variável da resposta (Gianola *et al.*,2011), por isso elas se tornaram muito uteis na area da predição, em que a partir de dados atuais, se deseja encontrar um um valor ou estado futuro.

Muitos trabalhos com RNA foram e estão sendo desenvolvidos para predição nas mais diversa áreas, a exemplo de Okut *et al.* (2011) utiliza RNA para a predição do índice de massa corpórea em ratos. Gianola *et al.* (2011) desenvolveram um estudo envolvendo a utilização de RNA para a predição de características complexas. Kalogirou *et al.* (2014) utiliza RNA para predizer o desempenho energético diário de um sistema previsão de energia solar.

Pela sua grande utilização nas mais diversas áreas de conhecimento e sua eficácia, decidiu-se verificar a eficácia das RNAs para predição, para isso, verificou-se que um setor que é muito importante e necessita de ajuda nas pesquisas por tecnologias para o Biodiesel.

O biodiesel é um combustível renovável gerado a partir de oleos de origem animal ou vegetal para ser utilizado em motores a diesel (Rathmann,2007). Ele é menos poluente que o diesel oriundo do petróleo.

Diante da atual situação de ameaça futura de diminuição das reservas de petróleo e a crescente consciencia ambiental pela diminuição da poluição, o Biodiesel vem se tornando um item muito pesquisado no mundo em busca de uma produção mais econômica.

O Brasil começou a estimular a produção, uso e pesquisas de Biodiesel em 2003 com o lançamento do Programa Nacional de Produção e Uso do Biodiesel. Esse programa determinou o diesel no país, deveria receber um percentual de Biodiesel.

Com esse programa, muitas universidades e centros de pesquisas iniciaram suas pesquisas sobre biodiesel. No Maranhão, o Laboratório de Análises e Pesquisa em Química Analítica de Petróleo e Biocombustíveis da UFMA (LAPQAP / UFMA) e o Laboratório de Pesquisa em Química Analítica (LPQA / UFMA) desenvolvem pesquisas com Biodiesel.

Nessas pesquisas, verificou-se que eles devem sempre produzir o biodiesel e encontrar seus índices de qualidade. Cada índice de qualidade se dá por um tipo de análise laboratorial, acarretando certo gasto de tempo.

Uma das análises é a cromatografia gasosa que apresenta a composição química, mais especificamente, o percentual de ésteres que compõem o biodiesel produzido. Pesquisas realizadas, como a de Knothe (2005), afirmam que alguns itens de qualidade do biodiesel são fortemente influenciados pela composição química.

Diante da necessidade de aumento da produção de biodiesel e dos altos investimentos em pesquisas, torna-se necessário acelerar as pesquisas quanto ao potencial de produção de biodiesel no Maranhão. Para tal, esta pesquisa vem para desenvolver um software que auxilie e acelere nesta descoberta.

Utilizando a relação que há entre ésteres e índices de qualidade, objetiva-se utilizar as redes neurais artificiais para prever esses a partir de uma única análise laboratorial, ou seja, utilizando o número de ésteres. Este trabalho pretende apresentar um estudo sobre redes neurais e então apresentar a melhor RNA para prever esses índices.

Para realizar esta previsão, foi desenvolvido um protótipo de software baseado em RNA, o Sys-PANN (System of Prediction by Artificial Neural Network) que deverá encontrar a melhor RNA para prever esses Índices. Com este software, as pesquisas serão aceleradas e algumas análises laboratoriais serão dispensadas.

Diante de tais objetivos apresentados, esta dissertação está estruturada de forma que possa transmitir o conhecimento necessário para o bom entendimento do trabalho

Assim, este capítulo 1, apresenta uma introdução contendo uma descrição geral do trabalho para que se tenha um rápido entendimento, além de apresentar a problemática e objetivos do trabalho.

O capítulo 2 apresenta uma abordagem sobre o Biodiesel, sua importância e como ele poderá ser usado pela RNA, assim apresentando um estudo sobre a sua estrutura química, como é sua produção e sua avaliação de qualidade.

O capítulo 3 irá apresentar uma revisão bibliográfica sobre as Redes Neurais Artificiais, incluindo uma descrição geral desde sua unidade básica até seus tipos.

O capítulo 4 apresenta um estudo aprofundado sobre um tipo específico e muito utilizado de RNA, o Perceptron de Multi-Camadas. Este capítulo dará ênfase as particularidades que podem ser utilizadas na predição.

O capítulo 5 apresenta o desenvolvimento do sistema proposto, Sys-PANN. Será explicado como ele foi projetado, implementado e testado. Ele incluirá a descrição do teste de validação do sistema e seus resultados.

O capítulo 6 apresenta as considerações finais desta pesquisa, os resultados obtidos, limitações e trabalhos futuros.

2. BIODIESEL

O biodiesel é um combustível renovável constituído de uma mistura de ésteres monoalquílicos de ácidos graxos de cadeia longa que é produzida a partir de óleo de vegetais, como girassol, mamona, soja, babaçu e demais plantas oleaginosas, ou de gorduras animais, para ser usado em motores a diesel. (Rathmann,2007; BIODIESELBR,2012; ANP, 2012).

A lei Nº 11.097, de 13 de janeiro de 2005 estabelece a seguinte definição para Biodiesel :

“Biodiesel é biocombustível derivado de biomassa renovável para uso em motores a combustão interna com ignição por compressão ou, conforme regulamento, para geração de outro tipo de energia, que possa substituir parcial ou totalmente combustíveis de origem fóssil”.

Entre as principais vantagens do biodiesel está o fato de poluir menos que o diesel fóssil. Ele reduz a emissão de dióxido de carbono, de particulados e fuligem, de hidrocarbonetos não queimados e de óxidos de enxofre em relação ao diesel derivado de petróleo (MDICE, 1985; Rathmann,2007). Em comparação, os motores a diesel emitem 60% a mais de CO e 80% a mais de NOx do que os motores a biodiesel (Guarrieiro, 2011)

Como parte de sua matéria-prima são as plantas oleaginosas¹, a produção de biodiesel também contribui na diminuição dos efeitos da emissão de CO₂, uma vez que o seqüestro de carbono feito pela plantação de oleaginosas, retira o gás carbônico produzido na queima do biodiesel e outros combustíveis fósseis (Rathmann,2007).

No Brasil, o biodiesel é visto com bons olhos pelo fato do país possuir um grande potencial para produção de biodiesel oriundo as grandes vantagens agrônomas, como o clima, abundância hídrica e extensas áreas agricultáveis a serem explorada, sem a necessidade de avançar sobre áreas florestais ou reservas demarcadas, além das áreas não propícias ao cultivo de gêneros alimentícios, mas com condições favoráveis ao plantio de inúmeras

¹ São vegetais possuidores de óleos e gorduras no seu caule, ou folhas ou frutos, e podem ser extraídos e transformados em biodiesel.

oleaginosas (BIODIESELBR,2012). Esta vantagem possibilita a grande diversidade de plantas oleaginosas cultiváveis por todo território nacional, tais como a palma e o babaçu no norte, a soja, o girassol e o amendoim nas regiões sul, sudeste e centro-oeste, e a mamona, que além de ser a melhor opção do semi-árido nordestino, apresenta-se também como alternativa às demais regiões do país.

Devido a estas vantagens e seu potencial o biodiesel será estudado neste trabalho para lhe fornecer uma contribuição em sua produção. Mas para isso, será detalhado a diante sua estrutura química, como é sua produção e sua avaliação de qualidade.

2.1. Estrutura Química do Biodiesel

O biodiesel é constituído quimicamente por uma mistura de espécies de ésteres metílicos de ácidos graxos. Essas espécies de ésteres metílicos diferem entre si pela variação na quantidade de carbono com cadeias de 8 a 24 átomos de carbono e na ligação entre esses átomos, com diferentes graus de insaturação (Hoekman *et al.*, 2012).

As variações nas características do biodiesel são determinadas pela concentração das espécies de ésteres no biodiesel. A tabela 1 abaixo demonstra a disposição dos ésteres no biodiesel de cutieira e mamão:

Tabela 2.1 - Exemplo de disposição dos ésteres no biodiesel (adaptado de Melo *et al.*,2010 e Santos *et al.*, 2007)

Ésteres metílicos	Estrutura	Mamão de Golden (%)	Cutieira (%)
Palmitato de metila	C 16:0	6,00	18,89
Estearato de metila	C 18:0	3,20	5,19
Oleato de metila	C 18:1	4,00	69,77
Linoleato de metila	C 18:2	76,60	4,84
Outros	--	10,2	1,31

Pode-se visualizar também a convenção de nomenclatura dos ésteres na tabela 1. Esta convenção é composto por dois números separados por ":" , o primeiro número refere-se ao número de átomos de carbono na

cadeia e o segundo número indica o número de ligações duplas carbono-carbono na cadeia (Hoekman *et al.*, 2012).

Atualmente, para se realizar a identificação preliminar e a composição relativa dos ésteres metílicos que compõem um biodiesel, utiliza-se a análise de cromatografia gasosa do biodiesel que consegue identificar os ésteres e suas concentrações.

2.2. Produção de Biodiesel

Os óleos extraídos das plantas são constituídos geralmente de ácidos graxos, fosfolípidos, água, substâncias odorantes e outras impurezas (Murugesan *et al.*, 2009). Com esta composição, o óleo vegetal puro não pode ser utilizado diretamente como combustível pois não possui as propriedades adequadas, acarretando problemas para os motores diesel.. Para superar estes problemas, o óleo necessita de um processo que modifique este óleo quimicamente (Demirbas, 2008).

O processo de produção de biodiesel engloba três metodologias: esterificação, craqueamento e transesterificação (Coelho, 2007). Destas metodologias, a transesterificação de óleos vegetais é atualmente o método mais utilizado, devido, principalmente, as características físicas dos ésteres de ácidos graxos que são muito próximas daquelas do diesel além de ter um desenvolvimento tecnológico maior em relação aos demais (GERIS *et al.*, 2007).

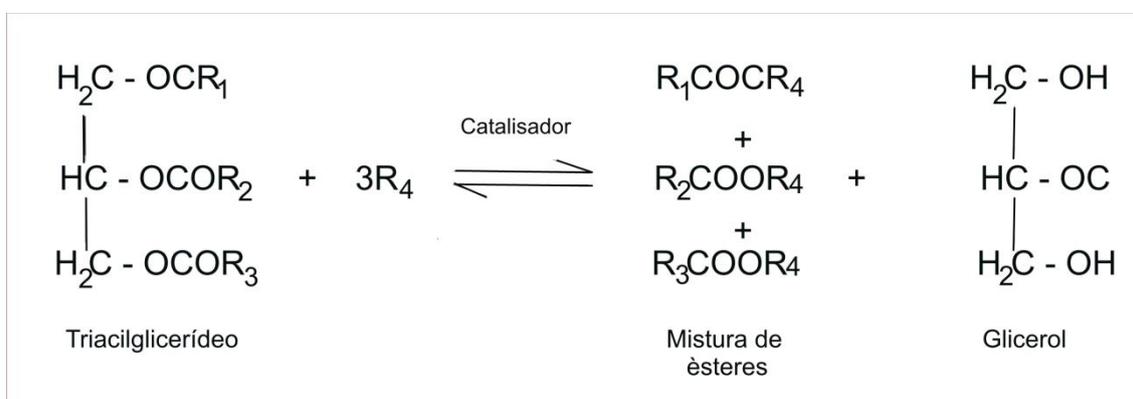


Figura 2.1 - Reação de transesterificação reversível (GERIS *et al.*, 2007).

Na transesterificação de óleos vegetais, o biodiesel é obtido através da reação de triacilglicerídeos de ácidos graxos (óleo) com álcool na presença de uma base ou ácido forte na função de catalisador, produzindo uma mistura de ésteres de ácidos graxos (biodiesel) e glicerol (glicerina) (Tolmasquim, 2000; Geris *et al.*, 2007). Na figura 2.1, segue-se essa reação.

No Brasil, a tecnologia de produção mais utilizada é a rota tecnológica de transesterificação metílica. Nesse processo, os óleos vegetais ou sebo animal são misturados com metanol. A escolha do metanol é justificada pelo alto custo do etanol em nosso país (PNPB, 2011).

Todo processo de produção envolve várias etapas operacionais. A figura 2.2 ilustra o fluxograma deste processo de transesterificação que é descrito a seguir.

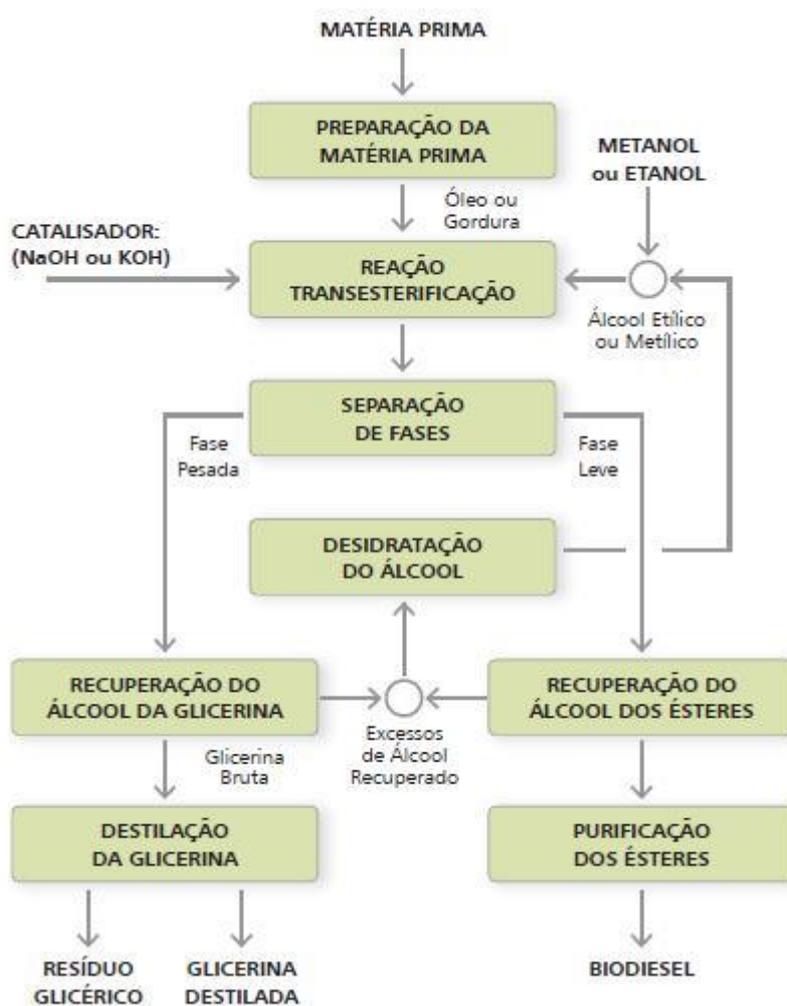


Figura 2.2 - Fluxograma do Processo de Produção de Biodiesel (PARENTE, 2003)

Neste processo de produção, a matéria prima é o óleo vegetal, que antes de tudo, necessita ser extraído dos grãos. Para isso, utiliza-se um processo de cozimento e rompimento das células vegetais, chamado de extrusão. A extrusão consiste em submeter os grãos a altas temperaturas e pressão, durante um curto período de tempo, para que ocorra a trituração, rompimento das células oleaginosas e cozimento apropriado para eliminar fatores anti-nutricionais, e deixando o óleo exposto e facilmente extraível. Feito isso, é gerado uma massa que é comprimida gradativamente e o óleo flui para um filtro-prensa para separação dos sólidos que ainda permanecem em suspensão, obtendo-se o óleo bruto (DFBIO, 2011).

Na preparação da matéria-prima, o objetivo é obter as melhores condições para efetivação da reação de transesterificação, com a máxima taxa de conversão. A reação de transesterificação é a etapa da conversão, ou seja, do óleo ou gordura, em ésteres metílicos ou etílicos de ácidos graxos, que constitui o biodiesel (Parente, 2003).

Em seguida, a próxima etapa consiste em duas partes: Fase Pesada e Fase Leve, ambas separáveis por decantação e/ou centrifugação. A fase pesada é formada de glicerina bruta, excessos utilizados de álcool, de água, e de impurezas inerentes à matéria-prima. A fase menos densa é formada por mistura de ésteres metílicos com possíveis excessos de álcool e de impurezas (Parente, 2003). Nesta fase, o álcool, o catalisador e óleo são combinados em um reator e agitados por aproximadamente 1h a 60 °C (Coelho, 2007).

Os alcoóis mais utilizados na reação são os de baixo peso molecular, como o metanol e o etanol. O catalisador da reação mais utilizado é a base, pela sua capacidade de acelerar o processo e com melhor aproveitamento em relação ao ácido (Coelho, 2007).

A recuperação do álcool da glicerina inicializa na evaporação da fase pesada, constituída de água e álcool, onde esses constituintes voláteis são eliminados da glicerina bruta, os vapores são liquefeitos num condensador apropriado. Na recuperação do álcool dos ésteres, o processo é similar ao de recuperação do álcool da glicerina, o álcool é recuperado da fase mais leve, liberando para as etapas seguintes, os ésteres metílicos ou etílicos (Parente, 2003).

Na desidratação do álcool, os excessos residuais de álcool, após os processos de recuperação, possuem quantidades relevantes de água que devem ser separadas. A desidratação do álcool é realizada geralmente por destilação responsável pela separação do álcool e água (Parente, 2003).

Para a etapa de purificação dos ésteres, os ésteres são lavados por centrifugação e desumidificados, resultando no biodiesel, o qual deverá ter suas características conforme as especificações das normas técnicas estabelecidas para o biodiesel como combustível (Parente, 2003).

2.3. O PNPB e os Parametros de Qualidade de Biodiesel

Com o intuito de estimular a produção e consumo de biodiesel oriundo de diversas fontes oleaginosas provendo inclusão social e no desenvolvimento regional, o Governo Federal criou por decreto, em dezembro de 2003, o Programa Nacional de Produção e Uso do Biodiesel (PNPB). A partir de estudos do PNPB, foi sancionada a Lei nº 11.097/05, o marco regulatório do biodiesel no Brasil que estabelece percentuais mínimos de mistura de biodiesel ao diesel e o monitoramento da inserção do novo combustível no mercado.

De acordo com essa lei, ficou instituído a adição de 2% de biodiesel ao diesel (mistura B2) a partir de janeiro de 2008 e torna obrigatória a mistura de 5% (mistura B5) até 2013. Além da adição, ficou definido também a Agência Nacional do Petróleo (ANP) a atribuição de provedor da regulação, contratação e fiscalização do conjunto de atividades econômicas relacionadas a Biodiesel.

Com o intuito de garantir esta qualidade foi necessário estabelecer padrões de qualidade, objetivando fixar teores limites dos contaminantes que não venham prejudicar a qualidade das emissões da queima, bem como o desempenho, a integridade do motor e a segurança no transporte e manuseio (Lôbo, 2009). Através da Resolução nº 07 de 2008, a ANP criou especificações técnicas para verificar a qualidade do biodiesel a ser misturado no diesel mineral no Brasil.

Atualmente, esta resolução foi substituída pela de Nº 14 de 2012, no qual estabelece 23 critérios de especificação para qualidade do Biodiesel. Para

verificar todos esses critérios assim como os métodos que eles devem seguir, vide o anexo I.

Alguns dos critérios de qualidade do biodiesel são fortemente influenciados pelas estruturas moleculares dos alquil ésteres constituem estes biodieseis (Knothe, 2005). Dentre as 23 propriedades estipuladas pela ANP, alguns desses parâmetros analíticos foram adotados visando o controle deste feito sobre o desempenho do combustível. Dentre estes, estão (Lôbo, 2009):

- Massa específica,
- Viscosidade cinemática,
- Índice de iodo,
- Fração de destilados,
- Número de cetano,
- Ponto de névoa,
- Ponto de entupimento de filtro a frio e
- Ponto de fluidez.

As diferenças nas propriedades físicas e química entre os combustíveis de biodiesel podem ser explicadas, em grande parte pelos perfis de ésteres dos combustíveis. Duas características influentes são: o tamanho da cadeia carbônica e o grau de insaturação no âmbito das estruturas de ésteres (Hoekman *et al.*, 2012).

Com a influencia da concentração dos ésteres em algumas propriedades, verifica-se que se pode, a partir de uma única análise de cromatografia gasosa num biodiesel, identificar seus ésteres e ocasionalmente, os valores das suas (algumas) propriedades sem a necessidade de se realizar várias análises laboratoriais destinadas a encontrar as propriedades qualitativas.

2.4. Considerações Finais

O Biodiesel é um combustível de grandes vantagens ecológicas sobre o diesel, tanto que tende a ser utilizado em grande escala no futuro. Por isso este trabalho foca em produzir uma contribuição a este combustível. Para

isso, foi necessário entender, sua composição química, sua produção e sua análise de qualidade.

De porte desse conhecimento, percebe-se que a análise química do biodiesel pode ser facilitada a partir de um sistema computacional de predição de dados. Atualmente, a técnica computacional mais aconselhada para predição é a Redes Neurais Artificiais que será apresentado nos próximos capítulos.

3. REDES NEURAIS ARTIFICIAIS

A Rede Neural Artificial (RNA) é uma ferramenta de modelagem computacional que tem o seu funcionamento inspirado no modelo biológico do cérebro humano. Ela possui uma grande capacidade de modelar problemas complexos do mundo real (Basheer, 2000).

As RNAs possuem uma abordagem conexionista, uma forma de computação não-algorítmica que acaba sendo uma alternativa à computação algorítmica convencional, sendo esta baseada na lógica (Braga, 2007; Kratzer, 1991). A abordagem convencional possui a capacidade de resolver muitos problemas complexos, porém não conseguem ter um desempenho adequado para tarefas que são de fácil solução para o ser humano (Brunak, 1993).

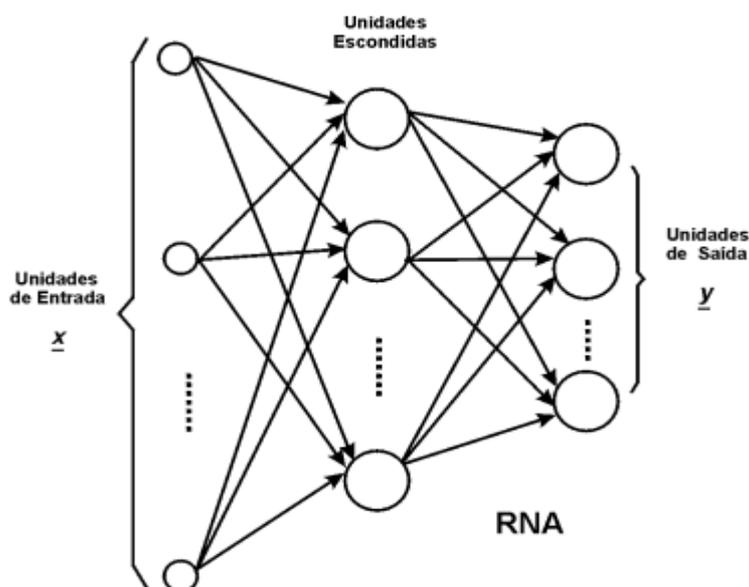


Figura 3.1-Exemplo de arquitetura de uma Rede Neural Artificial [Adaptado de (Haykin, 2001)].

A RNA pode ser definida como uma estrutura constituída por elementos de processamento simples que são adaptativos e densamente interligadas que são capazes de realizar cálculos massivamente paralelos para processamento de dados e a representação do conhecimento (Basheer, 2000; Haykin, 2001). Tais elementos estão dispostos em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais (Braga, 2007). A figura 3.1 exemplifica um modelo de rede neural artificial. As

RNAs são capazes de representar o conhecimento através de pesos associados às conexões.

A estrutura neural do cérebro humano processa a informação por meio de interação entre os neurónios. Quando essa interação é realizada por 1 trilhão de neurónios no sistema nervoso, ela permite que complexas tarefas serem executadas instintivamente pelo ser humano (Yabanova, 2013) e aliado a este sistema, o cérebro basicamente aprende com a experiencia.

Baseado no cérebro humano, uma rede neural extrai seu poder computacional de duas características:

1. a estrutura paralelamente distribuida e
2. a habilidade de aprender e de generalizar.

A generalização se refere ao fato de a rede neural produzir saidas adequadas para entradas que não estavam presentes durante o treinamento (aprendizagem). Estas duas capacidades de processamento de informação permitem que as redes neurais resolvam problemas complexos que são atualmente intratáveis (Haykin,2001)

3.1. Inspiração das Redes Biológicas

O cérebro humano é uma poderosa máquina de processamento de informação altamente complexa que trabalha de forma não linear e paralela. Sua capacidade de processamento é muito melhor que o computador mais rápido da atualidade (Haykin, 2001) .

As unidades básicas do cérebro são os neurónios. Estas unidades são responsáveis por processar as informações e transmiti-las. Este elemento é estruturado em três partes, assim como pode ser visualizado na figura 3.2:

- Dendritos: recebem as informações oriundos de outros neurónios e as tranfere ao corpo celular;
- Corpo celular: processa as informações e gera novas informações a serem transmitidas;
- Axônio: responsável por transmitir a outros neurónios as novas informações geradas pelo corpo celular.

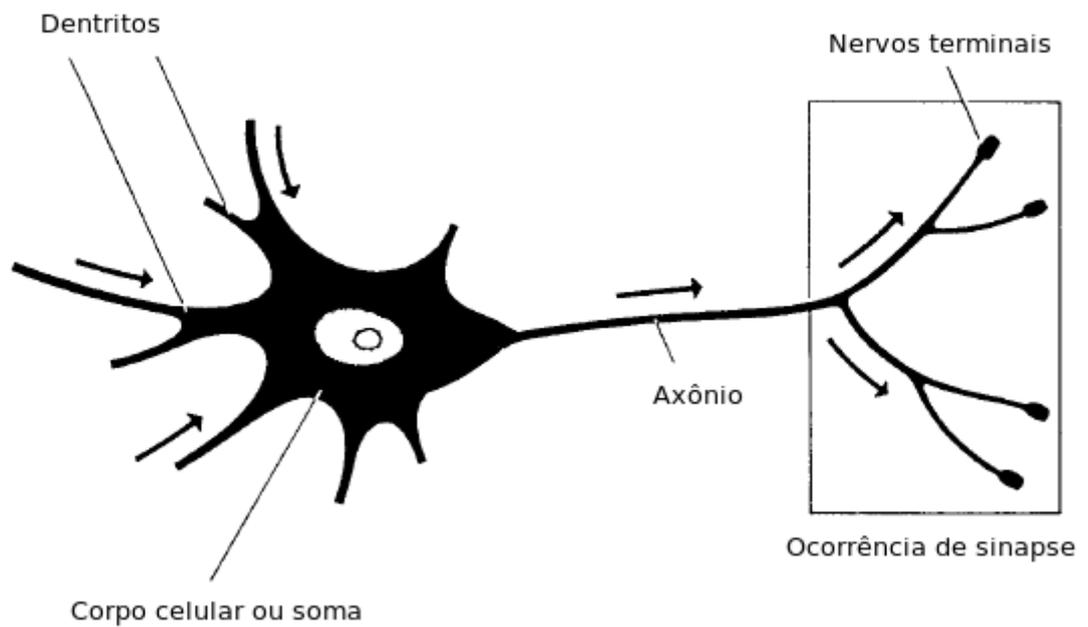


Figura 3.2 - Representação do neurônio biológico. [Adaptado de ARBIB, 2002]

O ponto de contato entre o axônio e o dentrito é conhecido como sinápsse. Nela que acontece a transmissão de informação que é realizada na forma de impulsos elétricos de um neurônio a outro. (Braga, 2007 ; Arbib, 2002). O processamento da informação no corpo celular se dá uma forma teoricamente simples: Os sinais oriundos dos dentritos são comparados e se eles forem suficientes para produzirem uma excitação limite, o corpo celular produzirá um impulso nervoso (Braga, 2007; Haykin, 2001).

3.2. Histórico

A partir da compreensão sobre o sistema nervoso biológico e seu poder de processamento, vários pesquisadores tentaram modelar e simular-los. Os pioneiros nos estudos de redes Neurais Artificiais foram McCulloch e Pitts (McCulloch,1943) quando, em 1943, lançaram o primeiro modelo artificial de um neurônio biológico, um modelo matemático de neurônio que implementa de maneira simplificada os componentes e o funcionamento de um neurônio biológico. Este modelo, ficou conhecido também como neurônio MCP.

O modelo MCP seguia a lei do “tudo ou nada”, em que tanto os sinais de entrada quanto a saída são valores binários. Baseado nesse funcionamento, eles mostraram que seu modelo seria capaz de computar qualquer função computável (Braga, 2007; Haykin, 2001). Posteriormente, Donald Hebb (Hebb, 49) realizou, em 1949, o primeiro trabalho relacionado a aprendizagem de uma RNA, lançando o postulado de aprendizagem.

Após 15 anos do lançamento do neurônio MCP, Frank Rosenblatt (Rosenblatt, 1958) apresentou em 1958 o Perceptron, um modelo de RNA estruturado em camadas que é baseado no neurônio MCP como unidade básica e em uma regra de aprendizagem. Ele é um método pioneiro que utiliza a aprendizagem supervisionada que é muito útil para reconhecimento de padrões. Ele demonstrou um modelo de perceptron que possui uma camada e que é limitado a problemas linearmente separáveis (Braga, 2007; Haykin, 2001).

Na década de 80, John Ropfield descreveu o algoritmo de aprendizagem back-propagation assim como mostrou que as RNAs de múltiplas camadas são capazes de resolver problemas difíceis (Braga, 2007). A partir desse momento, as redes neurais artificiais ganharam força pois ela seria teoricamente capaz de resolver todo tipo de problema.

3.3. O Neurônio Artificial

Um neurônio é a unidade básica de processamento de informação de uma rede neural biológica. Assim como a biológica, a RNA tem como elemento básico o neurônio artificial. O modelo básico de neurônio artificial que se utiliza hoje é baseado no MCP que pode ser visualizado na figura 3.3.

O modelo básico de neurônio artificial, ao qual passará a ser chamado neste trabalho apenas de neurônio, é constituído pelos elementos:

- $X_1 - X_n$: sinais de entrada;
- $W_{k1} - W_{kn}$: pesos sinápticos de um neurônio k ;
- Σ : combinador linear ou junção aditiva;
- B_k : Bias;
- $\varphi(\bullet)$: Função de ativação e
- Y_k ; Porta de saída de um neurônio K

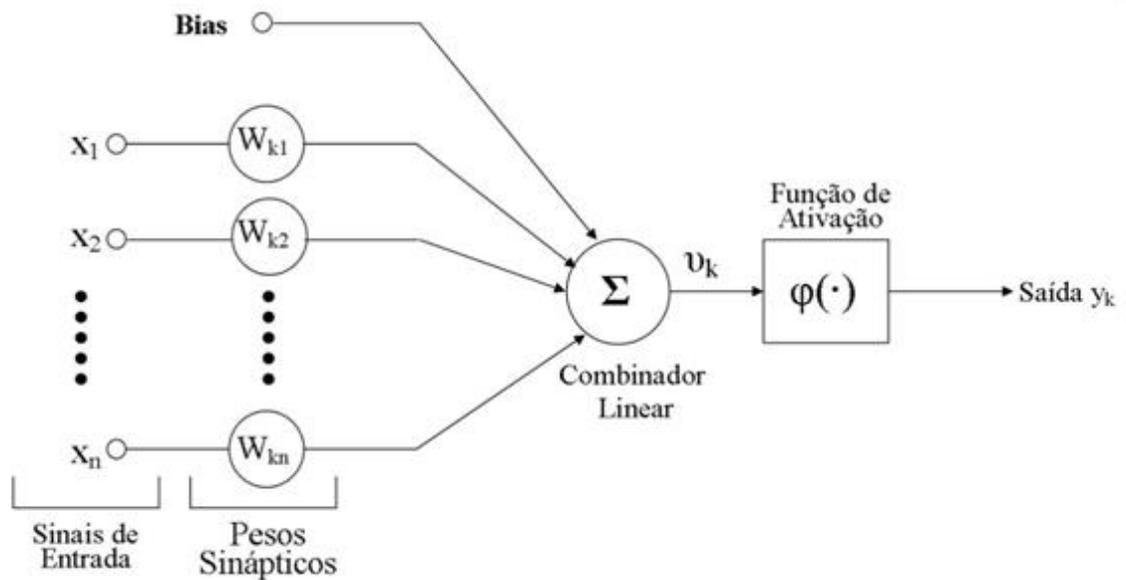


Figura 3.3- Modelo básico de um Neurônio artificial (Fonte: Haykin,2001)

Neste modelo, as entradas do neurônio k podem ser entradas externas ou saídas de outros neurônios. Essas entradas assim como a saída, são sinais variando entre 1 e -1, diferentemente do MCP por ser de natureza binária, trabalha com sinais de entrada e saída, 0 ou 1. Cada sinal de entrada X_j tem seu peso sináptico W_j referente.

Básicamente, uma entrada i de um neurônio k recebe um sinal X_i , este sinal é multiplicado pelo peso sináptico W_{ki} . Estes pesos podem ter valores positivos ou negativos. Os pesos servem para determina qual grau de relevancia que o neurônio deve adotar aos sinais desta entrada i .

Depois que cada sinal de entrada ter sido ponderado pelo seu respectivo peso, acontece a junção aditiva no combinador linear com a soma de todas as multiplicações anteriores. Posteriormente, uma função de ativação é utilizada para limitar a amplitude da saída do neurônio entre -1 e 1. No MCP, função de ativação funciona ativando ou não a saída, assim produzindo um sinal de saída de valor 0 ou 1 (Braga, 2007; Haykin, 2001).

Diferentemente do MCP, o modelo atual de neurônio possui também o bias b_k que é reponsavel por por aumentar ou diminuir o valor de entrada da função de ativação (Haykin, 2001).

Em termos matemáticos a junção aditiva do neurônio k pode ser representado pela equação 3.1:

$$u_k = \sum_{j=0}^n w_{kj} x_j \quad (3.1)$$

Onde n representa o número de sinais de entrada no neurônio k e u_k representa a saída do combinador linear que é enviada, posteriormente, a função de ativação resultando no sinal de saída y_k do neurônio, que é representado pela equação 3.2:

$$y_k = \varphi(u_k + b_k) \quad (3.2)$$

O bias b_k é um parâmetro externo do neurônio artificial k que tem a função de aplicar uma transformação à saída u_k do combinador linear (Haykin, 2001).

3.4. A Função de Ativação

A função de ativação $\varphi(u)$ define a saída do neurônio em termos do potencial de ativação partir do modelo MCP que possuía uma função de ativação que era baseado no "tudo ou nada", várias outras funções foram derivadas o qual permitem a produção de uma saída qualquer, não necessariamente zero ou um. Abaixo segue as principais e mais utilizadas funções de ativação:

a) Função Limiar: Também conhecida como *Threshold* ou degrau, é a função de ativação utilizada no modelo MCP. Ela que modela a propriedade do "tudo ou nada" deste modelo.

Esta função é representada da seguinte forma:

$$\varphi(u) = \begin{cases} 1 & \text{se } u \geq 0 \\ 0 & \text{se } u < 0 \end{cases} \quad (3.3)$$

Os neurônios que possuem função de ativação produzirão um sinal de saída y de valor 0, caso o valor de ativação u seja negativo e 1 nos casos em que o valor de ativação seja positivo. Esse valor varia conforme figura 3.4

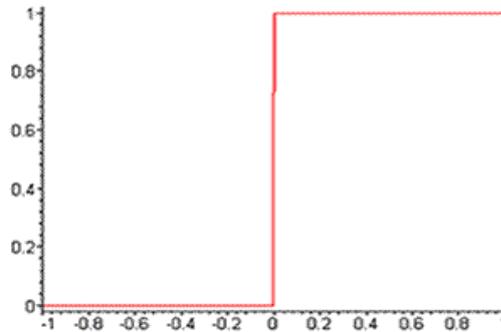


Figura 3.4- Gráfico da função limiar

b) Função Sigmóide: Esta é a função de ativação mais utilizada na construção de RNAs. Simon Haykin (Haykin,2001) define ela como uma função estritamente crescente que exibe um interessante balanço entre o comportamento linear e o comportamento não-linear. Possuindo a forma de S, pode assumir todos os valores entre 0 e 1.

É possível definir várias funções sigmoidais e a representação mais utilizada para ela é a função logística, definida por:

$$\varphi(u) = \frac{1}{1 + e^{-ua}} \quad (3.4)$$

Onde a é o parâmetro de suavização da curva da função sigmóide. Essa curva pode ser visualizada no figura 3.5

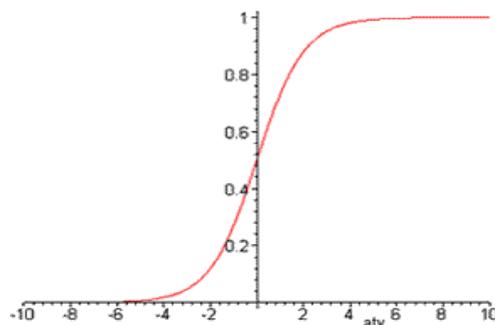


Figura 3.5- Gráfico da função Sigmoidal

c) Função Hiperbólica-Tangente: Semelhante a função sigmóide, ela também possui um gráfico em forma de S porém difere ao assumir todos os valores entre 1 e -1. Por possuir um intervalo maior de valores, ela também é uma das mais utilizadas. Sua equação matemática é definida por:

$$\varphi(u) = a \frac{e^{2u} - 1}{e^{2u} + 1} \quad (3.5)$$

Onde a é o parâmetro de suavização da curva da função hiperbólica-tangente. Essa curva pode ser visualizada na figura 3.6

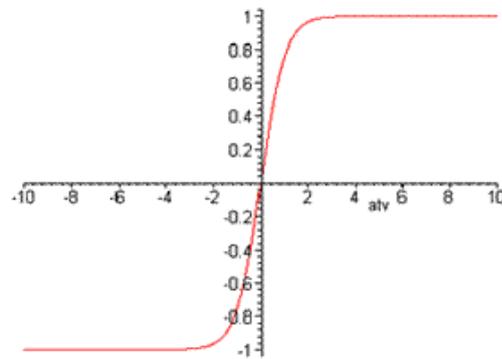


Figura 3.6- Gráfico da função hiperbólica-tangente

Além dessas três principais funções citadas, outras também são utilizadas:

- Função de Gauss
- Função Linear
- Função de Log
- Função Senóide
- Função Signum.

3.5. Topologia das RNAs

A Topologia de uma rede neural diz respeito a disposição dos neurônios na rede, como são estruturados e principalmente, qual o fluxo de transmissão de dados. Existem 2 classes: Feed-Forward e Back-Forward

a. Feed-Forward

Também chamadas de redes alimentadas adiante ou cíclica, o fluxo de informação é transferido de camada em camada adiante, no qual o fluxo de saída de uma camada é usado como fluxo de entrada da próxima camada e assim sucessivamente (Haykin,2001) como apresentado na figura 3.7

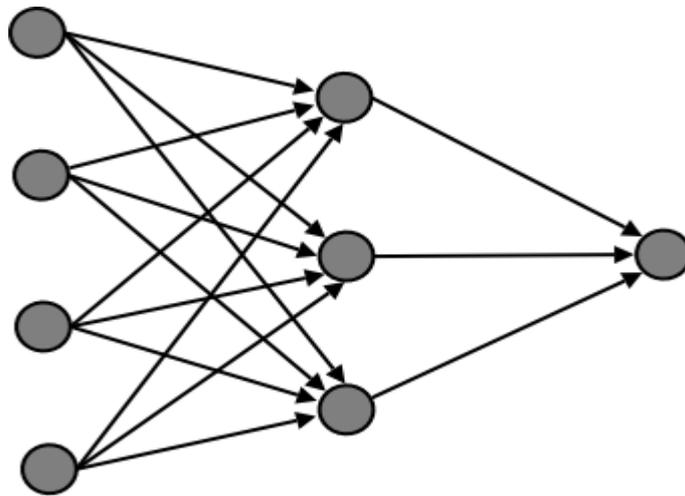


Figura 3.7- Rede Feed-Forward

b. Feed-Backward

Também conhecidas como redes recorrentes ou acíclica, nelas acontece a realimentação de neurônios, na qual a saída de um neurônio é utilizada como entrada pelo proprio neurônio e/ou por outros neurônios de camadas anteriores ou da mesma camada conforme a figura 3.8.

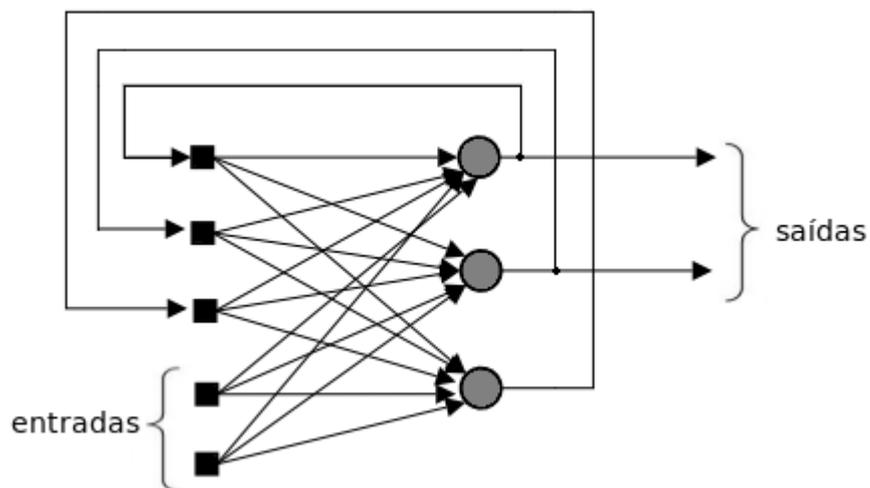


Figura 3.8 - Rede Feed-Forward

3.6. Modelos de RNA

Atualmente, há um grande número de modelos de redes neurais artificiais, no qual se considera o mais adequado para diferentes classes de problemas e sistemas (Novakovic, 1991). Esses modelos determinam como funcionará a rede e qual sua arquitetura. Dentre estes modelos, os mais importantes serão apresentados brevemente e podem ser visualizados na figura 3.9:

a) Modelo Perceptron

Proposto por F.Rosenblatt em 1957, foi o primeiro modelo conexionista desenvolvido nesta área. Ele é uma rede acíclica constituída de duas camadas de neurônios: uma camada de entrada e outra de saída, sendo que apenas a última camada é constituída de neurônios MCP enquanto a camada de entrada é constituída por receptadores. Por isso que são conhecidos também como perceptron de uma única camada.

Também chamado de Perceptron simples, é limitada à classe de problemas linearmente separáveis podendo ser utilizada em tarefas de classificação simples.

b) Modelo Perceptron Multi-Camadas(MLP)

Proposto por P.Werbos, D.Parker, D.Rumelhart em 1974, é uma extensão do Perceptron simples, no qual, além de possuírem camadas de entrada e saída, possuem múltiplas camadas intermediárias.

Ele é capaz de trabalhar com problemas não-linearmente separáveis, sendo muito utilizado para reconhecer padrões e características.

c) Função de Base Radial (RBF)

Foi descrito inicialmente Broomhead e Lowe(1988) e se assemelha com os perceptrons por se organizar em camadas e ser acíclica.. Ela organiza

seus neurônios em três camadas: camada de entrada, camada intermediária e camada de saída.

Neste modelo, apenas as camadas intermediárias e de saída são computáveis. Na camada intermediária as funções de ativação dos neurônios utilizam funções de base radial (ver Powell, 1988).

d) Sistema ART

Proposto por G.Carpenter e S.Grossberg em 1978, este modelo ficou conhecido como Classificador de Carpenter-Grossberg. Ele é uma rede cíclica baseada na Teoria da Ressonância Adaptativa (ART) (Grossberg, 1976) onde as informações são trocadas entre as camadas da rede.

O treinamento acontece quando a atividade neural entra em ressonância, ou seja, quando um padrão já treinado é apresentado e a rede o reconhece; ou quando um novo padrão de entrada é apresentado e a rede percebe que o padrão constitui uma nova informação e, então, entra em estado ressonante para memorizá-lo.

e) Rede de Kohonen

Proposto por Teuvo Kohonen em 1982, X, em que se é formado um mapa topográfico dos padrões de entrada no qual os neurônios são colocados em nós de uma grade, sendo que as posições dos neurônios são indicativas das características estatísticas contidas nos padrões de entrada (Kohonen, 1982).

f) Rede de Hopfield

Proposto por J.Hopfield (1982), é um modelo matricial não-linear recorrente que é mais utilizado para modelar situações com representação binária.

Esse modelo armazena padrões que são recuperados a partir de estímulos de entrada. Uma característica importante deste modelo é a chamada recorrência: as saídas ligam-se as entradas por um atraso de tempo;

com efeito, a resposta da rede sempre depende de seu estado anterior (Hopfield, 1982).

Estes modelos que foram apresentados anteriormente podem ser resumidos e definidos de forma geral agrupando-os em dois grupos de acordo com suas topologias assim como se observa na figura 3.9

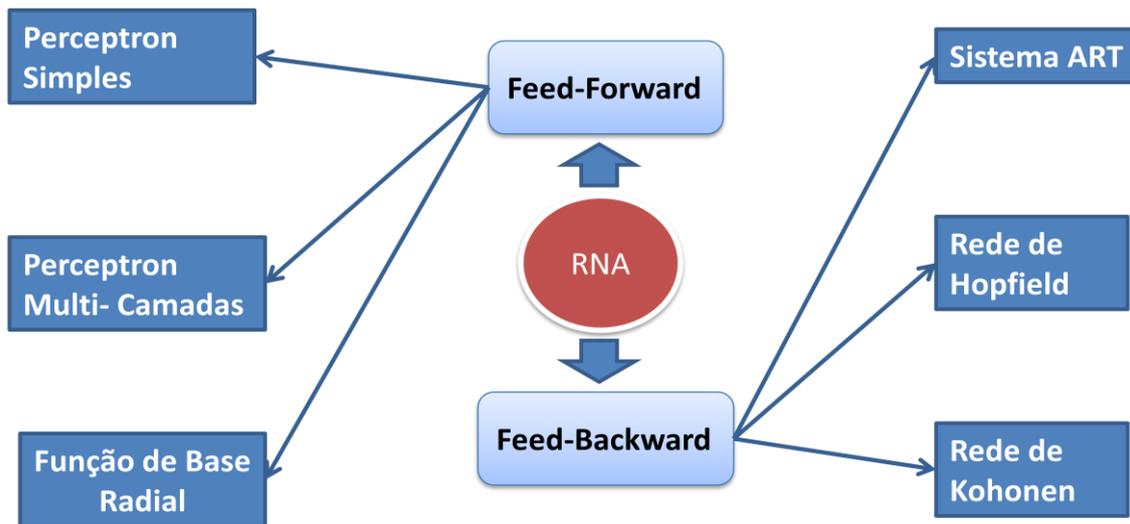


Figura 3.9- Modelos de Redes Neurais por Topologia

Atualmente, o modelo mais utilizado e que será utilizado por este trabalho é o modelo de rede Perceptron de Multi-Camadas ou Multilayer Networks. Ele será explicado no próximo tópico neste capítulo.

3.7. Aprendizagem de RNA

As Redes Neurais Artificiais são utilizadas para solucionar problemas e para que ela possa encontrar uma solução, ela necessita receber os dados do problema através de um estímulo externo na forma de dados ou sinais e processar esse estímulo para que assim possa produzir uma saída que será a solução do problema. Esse processamento que gera uma solução acontece nos neurônios no decorrer de toda RNA com as multiplicações dos sinais pelos pesos sinápticos e posteriormente, a função de ativação.

O segredo da RNA para resolver problemas está na adaptação dos pesos sinápticos que irão modificar os sinais de entrada produzindo a saída desejada, criando uma representação própria para o problema. Essa adaptação se dá através do processo de Aprendizagem ou Treinamento da Rede Neural.

Desta forma, a habilidade de aprender a partir do seu ambiente e melhorar seu desempenho através do aprendizado é de grande importância para uma Rede Neural. Uma boa definição sobre aprendizagem é feita por (Haykin,2001):

"Aprendizagem é um processo pelo qual os parâmetros livres de uma RNA são adaptados através de um processo de estimulação do ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros é efetuada".

Heaton (2010) também afirma que o treinamento é o processo onde os pesos da rede neural são ajustados para melhor produzir o resultado esperado e este acontecerá por várias iterações até que a taxa de erro da rede esteja abaixo de um nível aceitável.

Baseando-se nestas afirmações pode-se dizer que o treinamento é um processo interativo e iterativo. A Interação se dá com estímulos do ambiente com entradas que provocam uma resposta na forma de ajustes nos pesos sinápticos dos neurônios da rede neural. A RNA terá um melhor desempenho a partir de um processo de iterações desses ajustes de pesos.

De acordo com (Basheer e Hajmeer, 2000), uma rede neural só estará treinada quando ela for capaz de executar os seguintes objetivos:

- lidar com informações imprecisas, distorcidas e probabilísticas, sem causar efeitos adversos significativos sobre a qualidade de resposta e;
- generalizar respostas a partir das tarefas que aprendeu para os dados desconhecidos.

Para que RNA possa executar esses objetivos, ela necessita de um conjunto de regras e instruções bem definidas para a solução de um problema de aprendizado, chamado de Algoritmo de Treinamento (Haykin,2001). Diversos tipos desses algoritmos foram criados com características específicas

para determinados modelos de redes neurais, no qual diferem entre si pelo modo como modificam os pesos.

Esses algoritmos foram desenvolvidos baseados em métodos de treinamento, que poderiam ser ditos como regras gerais que os algoritmos devem seguir. Assim como os algoritmos, cada método também foi criado para específicos modelos de RNA. Esses métodos também podem ser reunidos em grupos ou paradigmas de aprendizagem. Atualmente, os dois principais paradigmas de treinamento são: Treinamento Supervisionado (com tutor) e o não-Supervisionado (sem tutor).

3.7.1. Treinamento Supervisionado

Este aprendizado é caracterizado por possuir um tutor ou professor que irá supervisionar e guiar o treinamento da rede. Nele, tutor detém o conhecimento do ambiente que é representado por um conjunto de exemplos entrada e saída, no qual a rede neural desconhece. O professor também irá direcionar o processo de treinamento indicando se seu comportamento é bom ou ruim para a rede.

No geral, o objetivo é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos pelo professor (Braga, 2007). Para cada entrada, uma resposta é fornecida pela rede neural e então é comparada à saída esperada. O erro verificado é informado à rede para que sejam feitos ajustes na rede. Este mecanismo é ilustrado na figura 3.10.

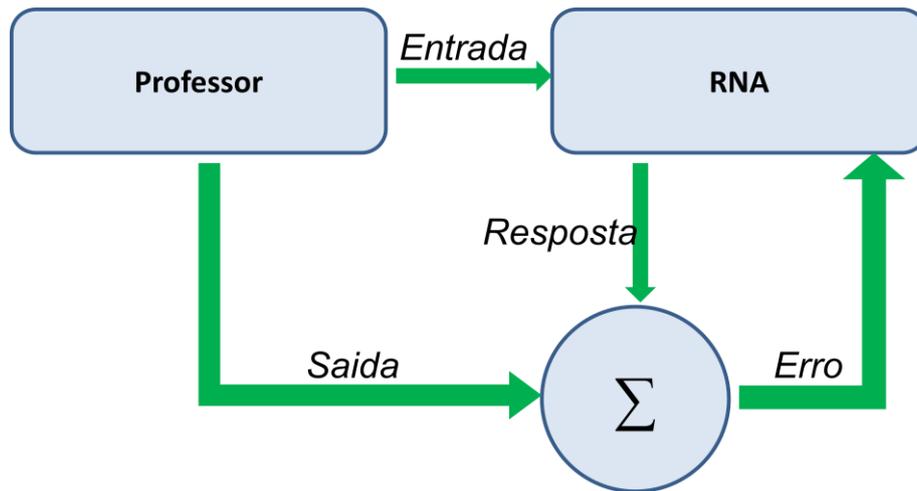


Figura 3.10- Aprendizado Supervisionado

Um dos métodos mais conhecidos que é baseado neste paradigma é o método da **Aprendizagem por Correção de Erro**. Este método ajusta os pesos sinápticos por meio do erro, que é obtido pela diferença entre a saída desejada e a resposta obtida da rede em um ciclo de treinamento. Com isso, a rede vai gradualmente, por iterações diminuindo o erro geral da rede.

Considerando um neurônio k estimulado por sinal x produz uma resposta y_k tendo uma saída esperada d_k . O sinal de erro denotado por e_k num período de n iteração do treinamento é explicitado:

$$e_k(n) = d_k(n) - y_k(n) \quad (3.6)$$

Após o erro do neurônio ter sido encontrado, o peso deverá ser ajustado a fim de minimizar o erro na próxima iteração. Para isso, o ajuste deverá seguir a chamada Regra Delta (Widrow e Hoff, 1960) que estipula que o ajuste de um peso deverá ser proporcional ao produto do erro pelo valor de entrada na sinapse naquele instante (Haykin, 2001). Sendo assim, o ajuste é definido como:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \quad (3.7)$$

onde $\Delta w_{kj}(n)$ é o ajuste do peso sináptico do neurônio k na sinapse j em uma iteração n e η é uma constante positiva que determina a taxa de aprendizado. Após encontrar este ajuste, o valor final do peso na próxima iteração será definida na equação 3.8.

$$w_{kj}(n + 1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (3.8)$$

Para avaliar o desempenho da rede em uma dada iteração utiliza-se a soma dos erros quadráticos sobre o conjunto de treino demonstrado na equação

$$E(n) = \frac{1}{2} \sum_{i=0} (d_i(n) - y_i(n))^2 \quad (3.9)$$

onde i representa a quantidade de exemplos utilizados para o treinamento. A cada ciclo de treinamento, quanto menor for o valor, melhor treinada estará a rede.

É importante ressaltar que a Regra Delta aparece tanto no algoritmo de treinamento do modelo perceptron quanto na sua generalização para redes de múltiplas camadas do perceptron, o algoritmo back-propagation. Ou seja, estes algoritmos utilizam o método de correção de erro (Braga, 2007).

3.7.2. Treinamento Não-Supervisionado

Este aprendizado é caracterizado por não possuir um tutor ou professor para supervisionar e guiar o treinamento da rede. Diferentemente do treinamento supervisionado, este treinamento utiliza apenas padrões entradas para se organizar e chegar numa regularidade de padrões conforme na figura 3.11. Por esta característica que ele também é chamado de aprendizado auto-organizado.

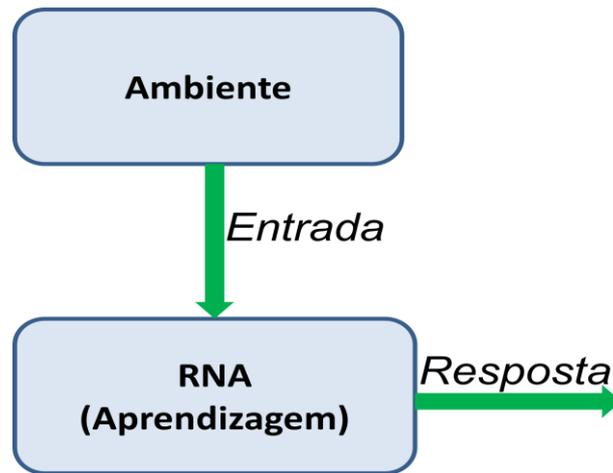


Figura 3.11-Aprendizado Não-Supervisionado

Este tipo de aprendizado só pode ser empregado quando existe redundância nos dados de entrada. Sem redundância é impossível encontrar quaisquer padrões ou características dos dados de entrada(Braga, 2007). Pois, ao processar as entradas, a rede tenta detectar suas regularidades estabelecendo representações internas nas RNAs para codificar características e classificá-las automaticamente.

Um dos métodos mais conhecidos que é baseado neste paradigma é o método de Aprendizado Hebbiano (Hebb, 1949) que é utilizado no Modelo de Hopfield. Este método baseado no postulado Hebb que é definido basicamente em duas regras:

- Se a ativação de um neurônio tiver grande influência na ativação de um outro neurônio, o suficiente para serem ativados simultaneamente, a ligação entre estes dois neurônios deve ser reforçada;
- Se a ativação de um neurônio não influencia a ativação de um segundo neurônio, sem haver um sincronismo de ativações, a ligação entre estes dois neurônios deve ser enfraquecida ou até mesmo removida;

Este postulado de Hebb pode ser expresso em termos matemáticos através da equação 3.11, que consiste em uma regra para a mudança do peso sináptico w_{kj} conhecida como a regra do produto de atividade:

$$\Delta w_{kj}(n) = \eta y_k(n)x_j(n) \quad (3.7)$$

onde k é neurônio com sinais pré-sinápticos e pós-sinápticos denotados por X_j e y_k , respectivamente e η é uma constante positiva que determina a taxa de aprendizado.

3.8. Aplicações da RNA

A escolha de um algoritmo de aprendizado específico é motivada pela tarefa ao qual se deseja que uma RNA desempenhe. Pelo fato de as RNAs serem aptas a resolver problemas de cunho geral, a área de aplicação das redes neurais é bastante extensa. Porém, todos esses problemas se enquadram em categorias de problemas que as RNAs podem realizar.

As categorias de problemas se diferem entre si pela maneira que tratam os padrões, atributos e classes. Um Padrão é uma descrição quantitativa de um objeto. Este padrão é definido por um conjunto de valores de atributos e o conjunto de padrões com características comuns forma uma classe.

Basheer & Hajmeer(2000) e Haykin(2001) apresentam em seus trabalhos categorias de problemas e tarefas aos quais as RNA podem trabalhar. As principais categorias que eles destacaram serão apresentadas a seguir.

3.8.1. Associação de Padrões

A associação trabalha com o desenvolvimento de um padrão associador através do treinamento e posteriormente, através da RNA, associa os dados aos padrões. Esta categoria pode ser empregada de duas formas:

- a) **Autoassociação de padrões:** No treinamento, um conjunto de padrões são armazenados na RNA. Posteriormente, a RNA deverá um padrão com ruído ou não a RNA e deverá identifica-lo, conforme na figura 3.12.

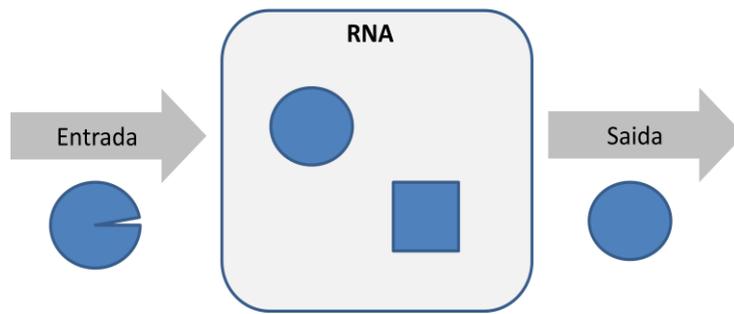


Figura 3.12- Modelo de uma Auto-Associação

- b) **Heteroassociação:** Neste treinamento, um conjunto de padrões é armazenado na RNA associado a outros padrões também armazenados. Posteriormente, a RNA receberá um padrão e deverá recuperar o outro padrão a que ele está associado, conforme mostrado na figura 3.13

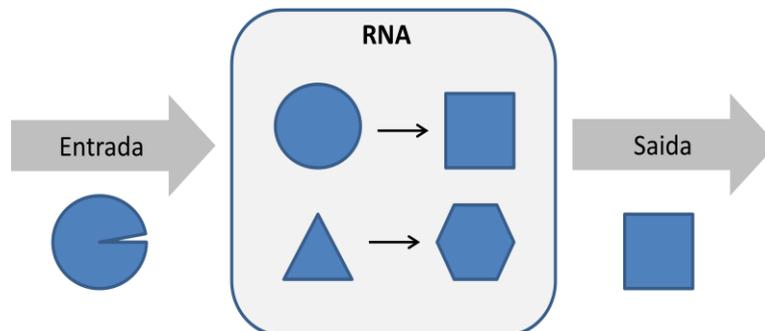


Figura 3.13- Modelo de uma Hetero-Associação

O treinamento comumente utilizado nesta categoria é o paradigma de aprendizagem não-supervisionado.

3.8.2. Reconhecimento de Padrões

É o processo através do qual um padrão de entrada é associado a alguma classe, dentro de grupo de classes previamente conhecido. Seu treinamento se dá pela apresentação a RNA de um conjunto de padrões, cada um deles associado uma classe a que pertence, sendo que cada classe é definida por uma série de valores de atributos.

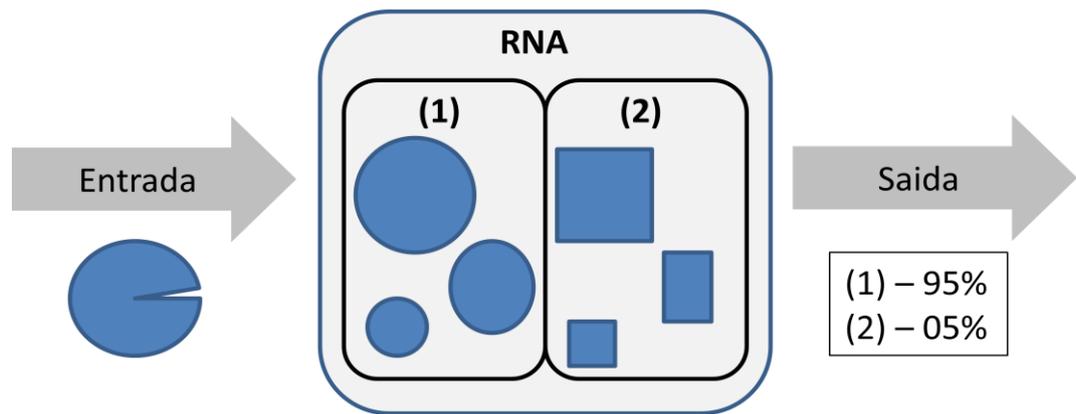


Figura 3.14 - Modelo de Reconhecimento de Padrão

Após o treinamento, a RNA receberá um padrão e deverá indicar/identificar a qual classe ele pertence, conforme na figura 3.14. O paradigma que se utiliza principalmente neste treinamento é o Supervisionado.

3.8.3. Aproximação de Funções

Nesta categoria, as RNA são utilizadas para aproximar de funções complexas e com alta não-linearidade, quando algoritmos exatos não o conseguem representa-los. Nesta situação, cada atributo de um padrão será uma variável da função desejada.

Para isso, durante treinamento, a RNA deve aproximar as regras subjacentes que correspondem ao elo dos dados de entradas para os saídas, gerando uma aproximação da função desconhecida. Posteriormente ao treinamento, a RNA receberá um conjunto de atributos e deverá devolver a resposta da função, conforme na figura 3.15

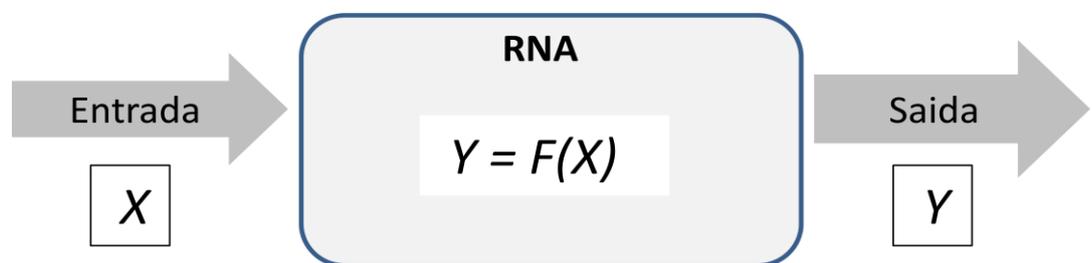


Figura 3.15 - Modelo de Aproximação de Função

Para esta categoria, o paradigma de aprendizagem utilizado é o supervisionado. As redes Perceptron de Multi-Camadas (MLP), que utilizam

treinamento supervisionado, são conhecidas como apróximadoras univesais de funções (Hornik *et al*, 1989).

3.8.4. Recuperação e Filtragem de Informação

A recuperação e fitragem de informação trabalha para retornar ao usuário a melhor informação de um grupo de informações. Ela trata com documentos, termos de indexação e as expressões de buscas dos usuários (Ferneda, 2006).

Nesta categoria, RNA tem que extrair a informação de interesse sobre um determinado grupo de dados ruidosos num determinado tempo n. O paradigma utilizado é o Aprendizado supervisionado.

3.9. Desenvolvimento de um projeto de RNA

Para que um sistema baseado em Redes Neurais possa ser utilizado para resolver um problema, ele necessita antes de tudo, preparar sua RNA para este problema. Para esta preparação da RNA e posteriormente sua utilização, é necessário o desenvolvimento de um projeto de RNA.

A utilização das RNAs segue uma metodologia que pode ser decomposta em sucessivas etapas (Kinnebrock, 1992). No geral, o desenvolvimento de um projeto baseado em RNA tende a seguir as etapas de desenvolvimento visualizados no fluxograma da figura 3.16.

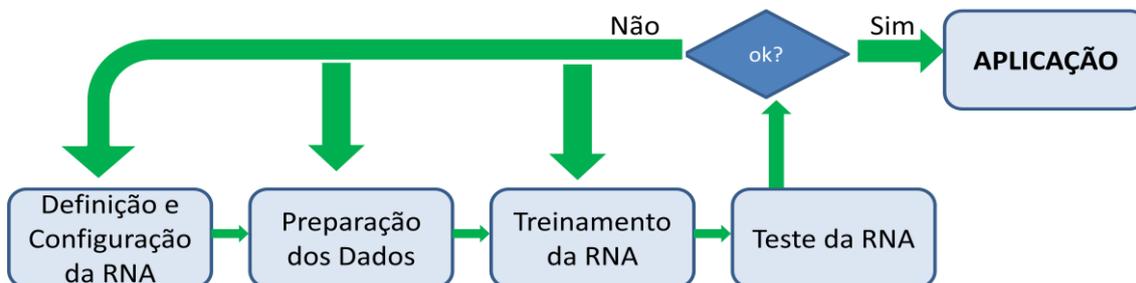


Figura 3.16 - Fluxograma de Desenvolvimento de RNA

3.9.1. Definição e Configuração da RNA

Neste primeiro passo para o desenvolvimento da rede neural artificial, o problema proposto e seus dados são analisados para se defina a rede neural mais adequada. Analisa-se a qual categoria de problema que o problema proposto se enquadra. Baseando-se pela categoria de problema e analisando-se os dados, a rede pode ser definida e configurada seguindo as etapas:

- I. Escolher o paradigma neural apropriado;
- II. Escolher o modelo apropriado;
- III. Escolher o algoritmo de treinamento e suas configurações; e
- IV. Escolher as funções de ativação

Após a escolha destes itens, a rede pode ser implementada. A definição dos itens de configuração de redes neurais é ainda considerada uma arte, que requer grande experiência dos projetistas (Almeida,2001).

3.9.2. Preparação dos Dados

Nesta etapa, os dados relativos ao problema são coletados em quantidade e qualidade suficientes para se realizar um bom treinamento da rede neural. Este grupo de dados deve ser grande para que possa minimizar ambigüidades resultante de dados de entradas semelhantes e erros nos dados. A qualidade dos dados se refere ao grupo ter que englobar todos os aspectos do problema, cobrindo as situações normais e exceções. Isso é necessário para que a rede crie um modelo suficientemente representativo e genérico do problema.

Os dados coletados, normalmente, são divididos em dois grupos:

- dados de treinamento: para o treinamento da rede
- dados de teste: para verificar a qualidade da rede pós treinamento.

Além da coleta e separação dos dados, é necessário pré-processar esses dados para deixa-los no formato utilizado pela rede, mais especificamente, pelas funções de ativação que trabalham geralmente com valores entre -1 e 1. Por isso, se os dados coletados forem não numéricos deverão ser transformados para dados numéricos. Posteriormente, se os dados

numéricos não estão dentro das magnitudes ($[1,0]$ ou $[1,-1]$) utilizadas pela rede, estes devem ser normalizados, ou seja, adequados para as magnitudes da função de ativação utilizado.

3.9.3. Treinamento da RNA

Nesta etapa, o algoritmo de treinamento escolhido e implementado é aplicado para ajustar os pesos das conexões. Nesta fase a RNA deverá ser alimentada pelo grupo de dados destinados ao treinamento.

O treinamento acontece iterativamente até que a rede atinja o comportamento desejado. Para que isso aconteça, é necessário atenção a inicialização dos pesos, o número de reinícios do treinamento e os critérios de parada que, assim como na etapa anterior, são definidos de acordo com a evolução da rede ao problema.

3.9.4. Teste da RNA

Nesta etapa, o desempenho da RNA será testada para verificar se o modelo criado após o treinamento da rede tem um índice de aceitável de acertos em relação a dados novos. Para realizar o teste, a RNA deverá ser alimentada pelo grupo de dados de teste a fim de se verificar seu comportamento.

Se a rede tiver um bom resultado nos testes, o projeto poderá avançar ao próximo passo. Caso contrário, deverá ser verificado qual o problema com a rede e então o projeto retornará a um dos passos anteriores para ser corrigida. A rede necessita apenas aumentar o ciclo de treinamento ou as vezes se percebe que o modelo escolhido foi errado, tendo que se iniciar todo o projeto.

3.9.5. Aplicação

Após a RNA ter sucesso no seu treinamento, o modelo criado poderá ser aplicado ao sistema proposto deixando para trás o algoritmo de treinamento. Neste momento, a rede formada será alimentada apenas por dados novos para retornarem uma resposta que se espera correta.

3.10. Considerações Finais

Este capítulo apresentou as Redes Neurais Artificiais e suas características inspiradas do cérebro humano. De forma genérica, foi apresentado o funcionamento de uma RNA desde sua unidade básica, o neurônio MCP até os modelos de RNA.

Esta apresentação foi necessária para se entender por quê a rede neural pode ser utilizada para prever as propriedades do biodiesel. Atestando sua utilidade, agora é necessário escolher a rede neural para o problema proposto.

Partindo dos conhecimentos apresentados no tópico 3.8, verifica-se que o problema necessita prever uma propriedade a partir dos níveis de ésteres no biodiesel, indicando que se deseja encontrar uma função matemática cuja variáveis da equação são os ésteres e o resultado é o valor propriedade.

Dessa forma, o problema se enquadra na categoria de Aproximação de funções, cujo paradigma destinado a esta categoria é o Supervisionado. Para esta situação o modelo mais adequado e que possui a “fama” de aproximador universal de funções é o modelo do Perceptron multi-camadas.

Por isso, o proximo capítulo irá tratar de forma mais profunda o Modelo MLP e seu algoritmos de treinamento na busca do mais adequado para a situação.

4. REDE PERCEPTRON DE MÚLTIPLAS CAMADAS (MLP)

A rede perceptron, como explicada no capítulo anterior, é uma rede constituída de duas camadas, no qual apenas a camada de saída é computável através dos neurônios MCP. A MLP é capaz de resolver apenas problemas linearmente separáveis. A figura 4.1 demonstra de forma simples a diferença entre problemas linearmente e não-linearmente separáveis (Braga, 2007).

Dez anos após o lançamento do Perceptron por Rosenblatt (1958), Minsky e Papert(1969) analisaram matematicamente este trabalho e demonstraram que as redes perceptron de uma camada não são capazes de solucionar problemas que não sejam linearmente separáveis assim como não haveria um treinamento que sustentasse multiplas camadas no perceptron.

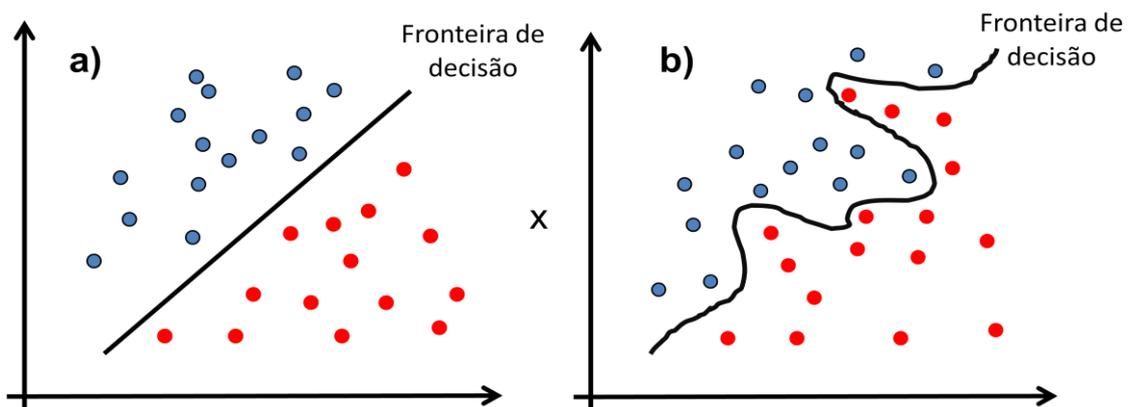


Figura 4.1- Classes de problemas: a) linearmente separáveis e b) não-linearmente separáveis

O estudo de Minsky e Papert gerou um período de esmorecimento no estudo das redes neurais nos anos 70 (Haikin, 2001). Porém, Rumelhart, Hinton e Williams(1986) apresentaram o desenvolvimento do algoritmo de treinamento de retropropagação (Backpropagation) e como ele funciona numa RNA. Neste trabalho, eles demonstraram que era possível treinar eficientemente redes perceptron de multiplas camadas e o quão eficientes poderiam ser, possibilitando assim o seu funcionamento.

Atualmente, a rede perceptron de multipla camada, também conhecida como rede MLP (Multi-Layer Perceptron) (Hornik *et al*, 1989) é o

mais popular e o mais utilizado modelo de redes neurais (Patnaik & Misra, 2000). Ele é aplicado principalmente para aproximação de funções, podendo também ser utilizado para classificação de padrões e outras tarefas.

4.1. Características da Rede MLP

A estrutura de uma MLP é caracterizada por ser constituída por três camadas. Uma camada de entrada, que é constituída de um número de nós de entrada que recebem os dados do exterior. A camada de saída é projetada para conter os nós de saída. Um conjunto de camadas intermediárias contém uma ou mais camadas (camadas escondidas) que possuem um número de nós que não têm nenhuma interação com o ambiente externo (Najjar, Basheer e Hajmeer, 1997). Com exceção da camada de entrada, todas as outras camadas possuem capacidade computacional pois são constituídas por neurônios MCP.

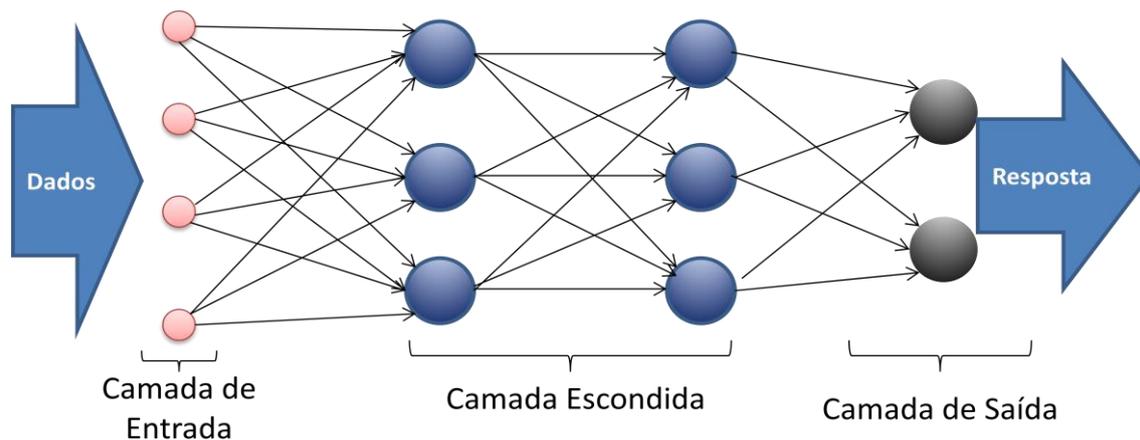


Figura 4.2- Arquitetura da Rede MLP (adaptada de Haykin, 2001)

Uma representação esquemática do MLP pode ser visualizada na figura 4.2, em que a camada de entrada recebe um vetor X de dados e os seus nós estão representados de forma diferentes demonstrando que eles são apenas receptores de dados e não realizam computação diferente das outras camadas que são constituídas de neurônios MCP. De acordo com a figura 4.2, este modelo demonstra que uma rede neural do tipo alimentada adiante e os nós de uma camada estão ligados a todos os nós da camada adjacente.

A MLP é uma generalização da rede perceptron. O acréscimo da camada oculta a rede perceptron adiciona um poder maior a esta rede de camada única, possibilitando classificação de padrões não-linearmente separáveis, sendo os neurônios ocultos responsáveis por capturar a não-linearidade dos dados.

A utilização de camadas ocultas possibilita a rede representar internamente qualquer conjunto de padrões de entrada. A quantidade de camadas vai de acordo com a tarefa desejada à rede. Cybenko (1988) e (1989) definiu a quantidade de camadas ocultas a partir das funções das redes:

- Uma camada intermediária para aproximar qualquer função contínua
- Duas camadas intermediárias para aproximar qualquer função matemática.

Outra grande característica ligada a este modelo está relacionado a função de ativação utilizado nos seus neurônios. Estas funções devem ser não-lineares e diferenciáveis para poderem possibilitar melhor resultado nos treinamentos (Braga,2007; Haykin,2001.) (posteriormente serão explicadas as razões e motivos da diferenciabilidade das funções). Atualmente, a função de ativação mais utilizada neste modelo é o sigmoide logística.

No aspecto que envolve o treinamento da rede, a MLP é caracterizado por utilizar algoritmo de treinamento do tipo supervisionado, sendo que eles se baseiam na técnica de propagação do erro para poder corrigir os pesos nos neurônios nas múltiplas camadas. O algoritmo de treino mais popular e utilizado pela MLP é o algoritmo de retro-propagação do erro ou Back-Propagation-of-error(BP) (Najjar,Basheer e Hajmeer,1997).

Devido aos motivos explicitados, este trabalho se voltará no quesito de treinamento da MLP ao algoritmo BackPropagation seguido de 3 variações desse treinamento. Na última seção deste capítulo será explicado qual a melhor maneira de se configurar uma MLP. O treinamento BackPropagation será descrito a seguir baseado em Braga (2007) e Haykin (2001).

4.2. Treinamento BackPropagation (BP)

O treinamento backpropagation se tornou um dos algoritmos para treinamento de maior popularidade devido a sua eficiência e, principalmente, pela sua relativa simplicidade de implementação e de armazenamento de informação relativo ao modelo de problemas, nos pesos sinápticos da rede.

Ele é um algoritmo supervisionado que utiliza um conjunto pareado de dados de treinamento (entradas e saídas desejadas) para, num um processo iterativo, ajustar os pesos da rede utilizando a regra de aprendizagem por correção de erro dos pesos sinápticos. Este processo consiste basicamente de dois passos, um para frente e outro para trás, conforme pode ser visualizado na figura 4.3.

O passo para frente é a fase forward que define a saída da rede para um dado entrada. Os valores provindos da camada de entrada são aplicados aos neurônios ocultos da camada adjacente e posteriormente suas saídas são transferidas como entradas aos neurônios da camada de saída gerando a resposta da rede. Durante este passo, os pesos mantem se fixos.

O passo para trás é a fase backward que utiliza o erro gerado entre a saída desejada e a saída fornecida pela rede para atualizar os pesos de suas conexões. Os pesos são ajustados inicialmente na camada de saída e posteriormente ajustados os pesos das camadas antecessoras de acordo com uma regra de correção de erro.

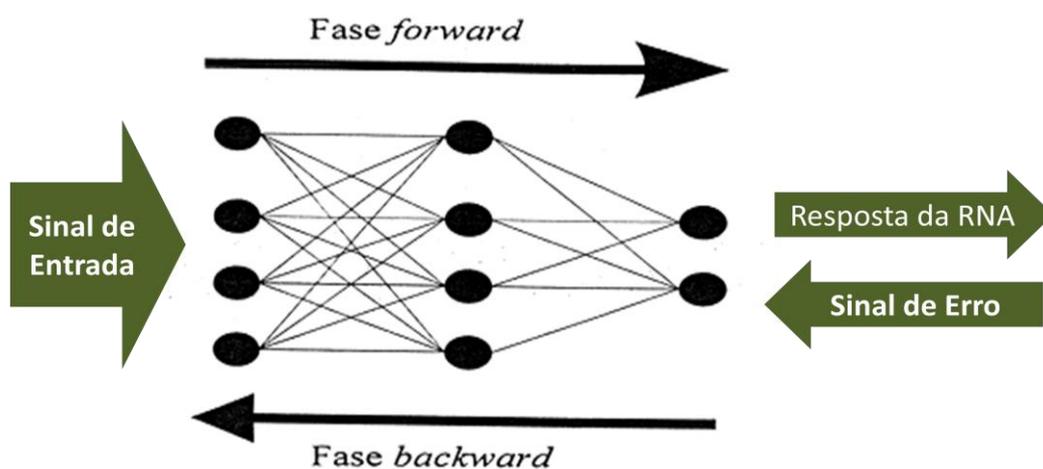


Figura 4.3- Fases e fluxo de dados em um treinamento Backpropagation

A figura 4.3 demonstra também os fluxos de dados ou sinais que se propagam pela rede. Dois tipos de sinais são utilizados pela rede:

- Sinal de entrada ou funcional: é o fluxo de sinal que se propaga, camada a camada, da entrada para a saída da rede. É o sinal utilizado pela fase forward.
- Sinal de erro: é o fluxo de sinal que se propaga (camada a camada) da camada de saída da rede para a camada de entrada, resultando na correção dos pesos. É o sinal utilizado pela fase backforward.

O algoritmo back-propagation é uma generalização do algoritmo de aprendizado do mínimo quadrado médio (LMS-Least Mean Square), que também conhecido como Regra Delta (Widrow e Hoff, 1960). Este algoritmo propõe uma forma de se encontrar o erro nas camadas ocultas, possibilitando o ajuste de seus pesos dos neurônios ocultos. Os ajustes dos pesos são realizados utilizando-se o método do gradiente descendente.

4.2.1. Descrição do Algoritmo BackPropagation

Para descrever este algoritmo, considere uma RNA que é alimentada por N padrões no qual $x[N]$ é o conjunto de entrada, $d[N]$ é o conjunto de saída desejado e $y[N]$ é o conjunto de saída gerada pela rede. Cada iteração deste treino será correspondente a um padrão utilizado, logo ele terá N iterações.

O objetivo principal deste algoritmo é minimizar o erro total da rede pelos neurônios da rede definido por E . Para encontrar este erro total, considere um neurônio j da camada de saída na n -ésima iteração, terá seu erro definido por:

$$e_j = d_j(n) - y_j(n) \quad (4.1)$$

Com o erro de um neurônio de saída, pode-se encontrar o erro instantâneo gerado por um padrão na rede através da soma dos erros quadráticos sobre todos os k neurônios da camada de saída na iteração n em:

$$\varepsilon(n) = \frac{1}{2} \sum_k e_j^2(n) \quad (4.2)$$

O erro total E é encontrado através do erro médio quadrático (MSE) do erro instantâneo de todas as n iterações ou de todos os padrões.

$$E = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad (4.3)$$

É importante ressaltar os pesos são atualizados a cada apresentação de um padrão n . Ao finalizar todos os N padrões, o algoritmo completou um período chamado de Época. A cada época completada, calcula-se um erro total para verificar se ocorreu a minimização deste.

Tendo os erros calculados, pode-se partir ao principal deste algoritmo: o ajuste dos pesos. Para este ajuste, necessita-se dos sinais de saída dos neurônios provenientes da função de ativação e da junção aditiva que foram apresentados no capítulo 3. Neste caso, além de se considerar o neurônio j tem-se a camada l ao qual o neurônio pertence. Sendo assim, a junção aditiva é calculada por:

$$u_k^{(l)} = \sum_{i=0}^{i+1} w_{ji}^{(l)}(n) y_i^{(l-i)}(n) \quad (4.4)$$

sendo o i representando os neurônios da camada anterior a l ($l-1$) conectados a j . A função de ativação é calculada por:

$$y_j^{(l)} = \varphi(u_j(n)) \quad (4.5)$$

Assim, o que se deseja é corrigir o valor das sinapses w de cada neurônio de cada camada. A correção é efetuada por

$$w_{ji}^{(l)}(n) = w_{ji}^{(l-1)}(n) + \Delta w_{ji}^{(l)}(n) \quad (4.6)$$

sendo Δw representando a correção que se deve empregar a w da sinapse de j da camada l conectada ao i .

A regra delta modificada define que a correção dos pesos seja realizada através do método do gradiente descendente do erro com relação ao peso. A correção é definida por:

$$\Delta w_{ji}^{(l)}(n) = \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (4.7)$$

η é o parametro da taxa de aprendizagem que define o tamanho do passo de atualização. $\delta_j^{(l)}$ é o gradiente local do neurônio j , ele é definido de forma diferente para os neurônios de camada oculta e de saída.

O gradiente local do neurônio j da camada de saída l é:

$$\delta_j^{(l)}(n) = \varphi'(u_j^{(l)}(n)) e_j^{(l)}(n) \quad (4.8)$$

O gradiente local do neurônio j de uma camada oculta l é:

$$\delta_j^{(l)}(n) = \varphi'(u_j^{(l)}(n)) \left(\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \right) \quad (4.9)$$

Em resumo, o gradiente local é o produto do erro gerado do neurônio j pela derivada de sua função de ativação. Nas camadas ocultas, o erro gerado é calculado recursivamente pelo gradiente de todos os neurônios que o sucedem com o peso de suas conexões a esse neurônio.

Como se pode notar, para se calcular o gradiente local, é necessário derivar a função de ativação, por esse motivo que o MLP se caracteriza por utilizar funções de ativação diferenciáveis.

Após a exposição matemática do BP, seu algoritmo pode ser melhor entendido com um pseudocódigo apresentado na figura 4.4 abaixo. Para facilitar o entendimento, este código está sendo empregado a uma RNA que possui apenas um neurônio de saída.

```

1 INICIO
2   iniciar os pesos das sinapses
3   entrar com padrões(X[N],D[N])
4   PARA n=0 ATÉ qtd_padrões FAÇA{
5     ; (gerando a saída de cada neurônio)
6     PARA k =0 ATÉ num_de_camadas FAÇA
7       calcular a junção aditiva de cada neurônio da
8       camada com a eq 4.4
9
10      calcular a função de ativação de cada neurônio
11      da camada com a eq 4.5
12     FIM_PARA
13     ; (cálculo da correção a ser empregada nos neurônios)
14     calcular o erro do neurônio de saída com a eq.4.1
15
16     calcular o gradiente do neurônio de saída com a eq.4.8
17
18     PARA k =num_de_camadas-1 ATÉ 0 FAÇA
19       calcular o gradiente dos neurônios da cam oculta k
20       com a eq.4.8
21     FIM_PARA
22     ; (correção dos pesos neurônios)
23     PARA k =0 ATÉ num_de_camadas FAÇA
24       calcular valor de correção dos pesos dos neurônios
25       da cam k com a eq.4.7
26
27       realizar a correção dos pesos dos neurônios da cam k
28       com a eq.4.6
29     FIM_PARA
30   FIM_PARA
31 FIM

```

Figura 4.4-Pseudocódigo do backpropagation

Esse ciclo de correção de erro do backpropagation continuará num loop enquanto uma condição de parada não for cumprida. Normalmente, a condição de parada deste loop será quando se encontrar o erro total das iterações menor que um valor determinado.

Embora o algoritmo BP seja um sucesso, ele também possui defeitos. Verificou-se que ele é muito lento para várias aplicações e seu desempenho piora sensivelmente para problemas maiores e mais complexos. Vários estudos foram desenvolvidos afim de melhorá-lo resultando em várias modificações.

4.2.2. BackPropagation com *momentum*

Proposta também por Rumelhart (1986), essa modificação foi lançada junto com o algoritmo BP afim de acelerar o processo desse treinamento. Ela é fácil de implementar pois consiste apenas do acréscimo do termo de momento a regra delta generalizada. O termo momentum é representado pela Equação 4.8.

$$\Delta w_{ji}^{(l)}(n) = \alpha \Delta w_{ji}^{(l)}(n-1) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (4.8)$$

onde α é a constante de momento, um número positivo que varia entre 0 e 1.

O termo momentum aproveita a contribuição da última alteração do peso, fazendo com que a variação do peso movimente-se na direção média fornecida pelas alterações anteriores, sem levar em consideração apenas a última alteração sofrida. O momentum acelera a atualização pois, tendo uma trajetória definida, aumenta o valor da variação do peso naquela iteração, sendo que sem o momentum, só ocorreria mais a frente.

4.2.3. Treinamento Resilient Propagation

Conhecido também como RProp, ele foi lançado por Riedmiller(1993). Ele acelera o processo de treinamento e acaba com um problema do BP conhecido como superfície plana, que é a influência negativa da derivada da função de ativação. Essa influência acontece quando derivada se aproxima de zero ocasionando num ajuste mínimo dos pesos.

Para acabar com este problema, o Rprop propõe que não se utilize a derivada para atualizar os pesos e sim, apenas utiliza-la para direcionar a atualização indicando se os pesos devem ser aumentados ou diminuídos. O valor do ajuste é determinado pelo delta de atualização individual de cada peso apresentado na equação 4.9:

$$\Delta_{ji}^{(l)}(n) = \begin{cases} \eta^+ \Delta_{ji}^{(l)}(n-1), & \text{se } \frac{\partial E(n-1)}{\partial w_{ji}} \cdot \frac{\partial E(n)}{\partial w_{ji}} > 0 \\ \eta^- \Delta_{ji}^{(l)}(n-1), & \text{se } \frac{\partial E(n-1)}{\partial w_{ji}} \cdot \frac{\partial E(n)}{\partial w_{ji}} < 0 \\ \Delta_{ji}^{(l)}(n-1) & \end{cases} \quad (4.9)$$

onde $\Delta_{ji}^{(l)}$ é o delta de atualização que cada peso possui e ele irá evoluir durante o treinamento de acordo com o erro.

O Rprop considera que se derivada parcial do erro em relação peso w_{ji} mantém o seu sinal, o que indica que seu último ajuste reduziu o erro cometido, o delta de atualização $\Delta_{ji}^{(l)}$ é aumentado pelo fator η^+ . mas se a derivada mudar de sinal, significa que seu último ajuste aumentou o erro cometido, então o delta é diminuído por η^- .

$$\Delta w_{ji}^{(l)} = \begin{cases} -\Delta, & \text{se } \frac{\partial E}{\partial w_{ji}} > 0 \\ +\Delta, & \text{se } \frac{\partial E}{\partial w_{ji}} < 0 \\ 0 & \end{cases} \quad (4.10)$$

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \Delta w_{ji}^{(l)}(n) \quad (4.11)$$

O RPROP altera o valor da actualização $\Delta w_{ji}^{(l)}$ do peso, directamente sem considerar o valor da derivada parcial, apenas o seu sinal, como visualizado na equação 4.10. Tendo o valor de actualização, o peso pode ser actualizado na equação 4.11.

Os parametros utilizados neste algoritmo são definidos apenas uma vez, e no caso, o autor já define quais valores os parametros devem ter sempre:

- $\Delta_0 = 0.1$
- $\eta^- = 0.5$
- $\eta^+ = 1.2$

O Δ_0 representa o valor inicial de todo delta dos pesos, posteriormente eles serão alterados durante o processo. Isso é um diferencial pois não se necessita ficar encontrando o valor mais adequado do parâmetro, diferente do BP que necessita descobrir o valor da aceleração e do momento mais adequado.

4.2.4. Treinamento Levenberg-Marquardt(LM)

Este algoritmo foi desenvolvido por Levenberg(1944) para resolver iterativamente problemas de minimização de funções não-lineares pelo método de mínimos quadrados e posteriormente foi melhorado por Marquardt(1963). Posteriormente o LM foi incorporado ao algoritmo de retropropagação por Hagan(1994).

Neste algoritmo, assumi-se que \mathbf{x} representa o vetor de pesos de um neurônio j e a função $V(\mathbf{x})$ é uma soma dos quadrados do erros em relação ao peso na equação 4.12:

$$V(\mathbf{x}) = \frac{1}{2} \sum_j e_j^2(\mathbf{x}) \quad (4.12)$$

A partir da função $V(\mathbf{x})$, calcula-se $\mathbf{J}(\mathbf{x})$, a sua matriz jacobiana². Tendo \mathbf{J} , pode-se calcular a variação do peso pelo método de Gauss-Newton na equação 4.13

$$\Delta \mathbf{x} = [\mathbf{J}(\mathbf{x})\mathbf{J}^T(\mathbf{x}) + \mu \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}) e(\mathbf{x}) \quad (4.12)$$

$$\mathbf{x}(\mathbf{n}) = \Delta \mathbf{x} + \mathbf{x}(\mathbf{n} - 1) \quad (4.13)$$

² matriz das derivadas parciais de uma variável específica (x^*) de uma função de múltiplas variáveis

sendo I a matriz identidade, J^T a matriz transposta e $e(\mathbf{x})$ o erro em relação ao peso. O parâmetro μ tem a função de estabilizar o treinamento ajustando a aproximação pois ele é multiplicado por um fator β quando o erro aumenta e é dividido pelo fator quando o erro diminui. Hagan(1994) recomenda que estes parâmetros tenham os seguintes valores:

- $\mu = 0.01$
- $\beta = 10$

Antes de atualizar o fator μ deve-se atualizar o peso com a equação 4.13 e depois realizar uma nova soma dos quadrados. O passo chave neste algoritmo é o cálculo da matriz Jacobiana que é empregado a partir de uma simples modificação do algoritmo backpropagation (Hagan,1994).

4.3. Configurando uma MLP

O capítulo 2 apresentou, entre outras coisas, qual a melhor forma de se criar um projeto de uma RNA. Uma das fases mais complicadas é referente a configuração da rede pois, como Almeida(2001) afirmou, a definição dos itens de configuração de redes neurais é ainda considerada uma arte, que requer grande experiência dos projetistas.

Todos os valores de parâmetros (dentro do universo de valores permitido para cada parâmetro) são válidos porém suas escolhas possuem resultados diferentes para cada rede e padrões. Pequenas diferenças nestes parâmetros podem levar a grandes diferenças tanto no tempo de treinamento como na generalização obtida.

Abaixo está a lista de parâmetros utilizados para configurar uma rede MLP. Não existe um método universal para defini-los porém podem-se seguir caminhos. É importante ressaltar que o caminho que irá produzir o melhor resultado é ainda é o da tentativa e erro.

a) Número de Camadas Escondidas

Um número indeterminado de camadas ocultas podem ser utilizadas em uma RNA porém não é recomendado. Se houver uma grande quantidade de camadas, a RNA terá um grande trabalho para atualizar os pesos durante o treinamento.

A camada de saída é a única que tem “noção” exata do erro cometido pela rede, as outras só tem estimativas desse erro (Braga, 2007). Quanto mais distante da camada de saída, mais impreciso se tornará o erro.

Se a RNA tiver por objetivo aproximar função, Cybenko (1988) e (1989) recomenda que se utilize uma camada intermediária para aproximar qualquer função contínua e duas camadas intermediárias para aproximar qualquer função matemática.

b) Número de neurônios da camada escondida

De todos os parâmetros, esse é o mais empírico de todos pois ele depende muito do número e qualidade dos exemplos de treinamento, e da complexidade da função a ser aprendida

Muitos autores estudaram a melhor forma de se chegar ao número de neurônios funcional a RNA. Hecht-Nielsen(1990) indica que se utilize um total de $2N+1$ (para N neurônios na camada de entrada) neurônios na camada oculta. Lippmann(1987) indica que se utilize duas camadas ocultas, a segunda deve ter o dobro de neurônios que a primeira.

Até o momento, o mais recomendado é verificar empiricamente. Recomenda-se que não se utilize neurônios em excesso, o que pode levar a RNA a memorizar os padrões de treinamento impossibilitando a generalização, e nem utilize neurônios em pouco número, o que pode levar a RNA a não criar um modelo amplo o suficiente fazendo com que gaste mais tempo tentando encontrar uma representação ótima (Braga, 2007).

c) Normalizando dados

A normalização é um procedimento necessário para que os dados possam suportados pela RNA, mais especificamente pelas funções de transferência.

As funções de transferência utilizadas pela MLP suportam valores de $[0,1]$ e $[-1,1]$. Para x^n o valor normalizado, seguem abaixo as opções de normalização:

- Para intervalo $[0,1]$:

$$x^n = \frac{x - x^{min}}{x^{max} - x^{min}} \quad (4.14)$$

onde x é um valor, x^{min} é o menor valor do grupo de dados e x^{max} o maior valor. Nesta situação, x^{min} será 0 e x^{max} será 1.

- Para intervalo $[-1,1]$:

$$x^n = \frac{x - x^{med}}{\max[x^{max} - x^{med}, x^{med} - x^{min}]} \quad (4.15)$$

onde x^{med} é o número médio de um grupo de dados de entrada. Nesta situação, x^{min} será -1, x^{max} será 1 e x^{med} será 0. Assim se deverá saber qual o valor médio do seu grupo de treinamento.

d) Inicialização dos pesos das sinapses

Após a definição das camadas e do número de neurônios, deve-se dar valores iniciais aos pesos das sinapses. Essa definição inicial é importante pois ela poderá ser a diferença entre um treinamento longo ou curto.

Se os valores dos pesos iniciais forem muito bons, a rede não necessitará de muitas rotinas para correção dos pesos, assim como se os valores forem muito ruins, a rede necessitará de mais tempo para correção dos pesos.

A inicialização dos pesos ainda é muito empírica por isso sua forma tradicional de inicialização é com valores aleatórios de distribuição uniforme. Outra forma é estipulando um limiar para os pesos que varie randômicamente em um intervalo.

Boers & Kuiper(1992) propõem utilizar o intervalo $\left[\frac{+3}{\sqrt{n}}, \frac{-3}{\sqrt{n}}\right]$ sendo n o número de nós de entrada. Fahlman (1988) propõe utilizar o intervalo $[-1.0,$

+1.0] embora seus melhores resultados foram com os intervalos [-5.0, +5.0] e [-4.0, +4.0].

e) Taxa de aprendizado

Quando se utiliza o treinamento BP e o BP com momentum, ela é um importante parâmetro pois controla a velocidade com que os pesos são ajustados. Ela é um número positivo que varia entre 0 e 1.

Ela pode ser constante no treinamento, no caso seu valor seria de forma empírica, ou variar no treinamento, iniciando com um valor pequeno e ir aumentando(Haykin,2001).

f) Fator de momento

Este fator é utilizado no treinamento BP com momentum para aumentar a velocidade de treinamento da rede neural. Ele é um número positivo que varia entre 0 e 1. Vale lembrar que um valor grande diminui o numero de interações e aumento a ascilação de aprendizagem.

g) Dinâmica de treinamento

A dinâmica de treinamento diz respeito ao tempo que os pesos devem ser atualizados na rede. Para isso, pode seguir duas opções:

- Por padrão: A cada apresentação de um padrão, os pesos são atualizados.
- Por ciclo (época): Após a apresentação de todos os padrões, os pesos são atualizados. O periodo de um conjunto de padrões é chamado de época.

h) Critério de parada

Após o início do treinamento, toda a rede já foi configurada, porém é necessário uma ultima configuração: quando a rede deve parar de treinar. Para isso, existem vários critérios e os mais utilizados são:

- **Número de Épocas:** a rede irá treinar durante um número determinado de épocas. O problema é não há garantia de que o erro tenha atingido um patamar aceitável e que a RNA terá aprendido.

- **Erro Total:** a rede irá treinar até que o erro total diminua até um valor x determinado. Esta situação é fácil de implementar porém erro pode indicar especialização da rede aos dados de treinamento.
- **Validação Cruzada:** esta técnica tenta parar o treinamento quando a rede começa a se especializar nos dados de treinamento e diminuir a generalização. Para isso, utiliza-se um conjunto de dados que não está no treinamento da rede, o conjunto de teste.

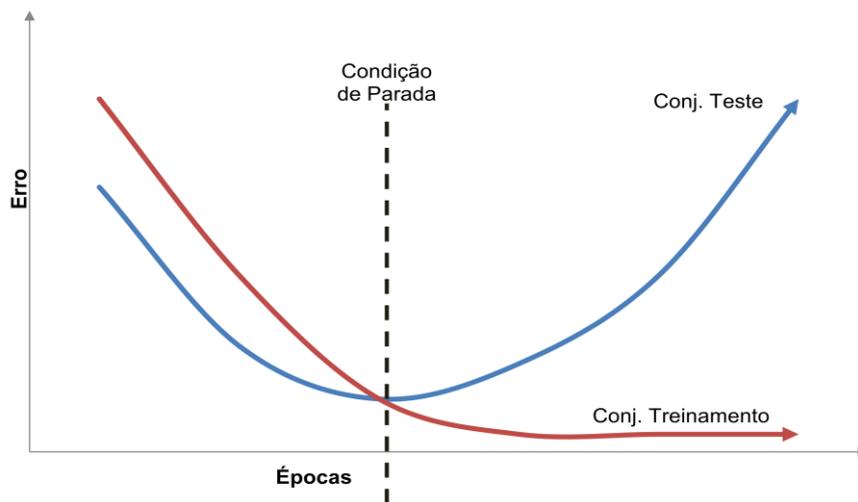


Figura 4.5- Gráfico representando a condição de parada

A cada x épocas, é realizada uma estimacão de erro da rede sobre o conjunto de dados de teste. Se o erro do conjunto de testes começar a aumentar, conforme na figura 4.5, o treino é paralisado.

4.4. Considerações finais

Este capítulo apresentou o modelo de Redes Neurais Artificiais mais utilizado, o Perceptron de Múltiplas Camadas. Acompanhando este modelo, foi apresentado primeiro e mais conhecido algoritmo de treinamento, o BackPropagation, acompanhado de três variações sua. Ao final, foi apresentada a melhor forma de se configurar uma MLP baseada no treinamento BP.

Realizar esta apresentação sobre RNA foi e suma importância para que se pudesse selecionar os conceitos que serão necessários para implementar a RNA que será utilizada para prever as propriedades do biodiesel. Por isso, o próximo capítulo dará ênfase a descrição do software que conterá a RNA para resolver o problema proposto.

5. SISTEMA PROPOSTO

O Laboratório de Análises e Pesquisa em Química Analítica de Petróleo e Biocombustíveis da UFMA (LAPQAP / UFMA) e o Laboratório de Pesquisa em Química Analítica (LPQA / UFMA) realizam pesquisas aplicadas ao desenvolvimento de metodologias analíticas voltadas para análise de qualidade de qualidades do biodiesel

Esses laboratórios produzem biodiesel para que possam realizar o desenvolvimento dessas metodologias. A seleção das melhores oleaginosas representativas do cenário de produção do estado do Maranhão para produção desse biodiesel é feita por meio de triagens desenvolvidas pelos laboratórios (LAPQAP/LPQA).

Após produzirem o biodiesel, o LAPQAP e LPQA realizam as análises laboratoriais para verificar se a amostra de biodiesel produzido está dentro das normas estipuladas pela ANP. Cada parâmetro especificado pela ANP possui um método específico de análise laboratorial para encontrar o seu resultado a partir de uma determinada amostra de biodiesel como pode ser visualizada no anexo I.

Com o objetivo de buscar melhores características dessa amostra, realiza-se também a análise de cromatografia gasosa do biodiesel para identificar os ésteres e suas concentrações. Como informado no capítulo 2, os perfis dos ésteres influenciam alguns dos parâmetros de qualidade.

Por isso, o sistema que será proposto visa acelerar o procedimento de análise de alguns parâmetros, utilizando a análise dos perfis de ésteres apresentados pela cromatografia gasosa da amostra para prever os resultados desses parâmetros. As análises laboratoriais que podem se beneficiar são:

- Massa específica,
- Viscosidade cinemática,
- Índice de iodo,
- Fração de destilados,
- Número de cetano,

- Ponto de névoa,
- Ponto de entupimento de filtro a frio e
- Ponto de fluidez.

O sistema se baseia na ideia de que, após a produção da amostra de biodiesel, o laboratório deve realizar a cromatografia gasosa da amostra para descobrir a composição química. Tendo a composição, um software poderá realizar uma busca de padrões dos ésteres e encontrar os resultados que seriam das análises laboratoriais sem a necessidade de realizá-las, ocasionando a redução do tempo e do custo das pesquisas de biodiesel.

O software a se desenvolvido é o Sys-PANN (Prediction by Artificial Neural Networks) que visa prever, através de técnicas de redes neurais artificiais, parâmetros de qualidade a partir da análise dos perfis de ésteres da amostra.

Como o Sys-PANN deverá funcionar baseado em RNA, ele deverá ser constituído de dois módulos para poder corresponder às funções básicas da RNA:

- Módulo de Treinamento: conforme visualizado na figura 5.1, será responsável pelo treinamento da rede.

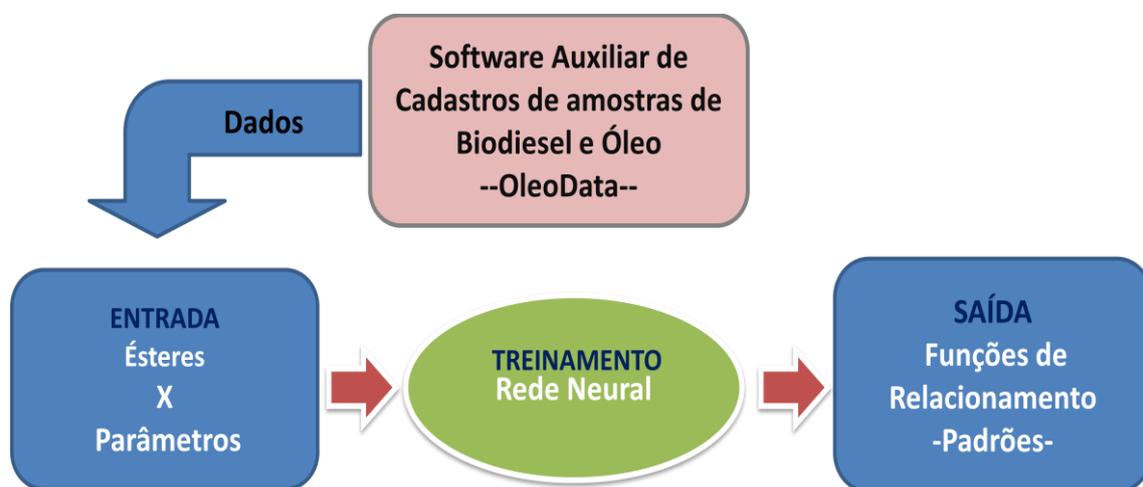


Figura 5.1- Arquitetura geral do Sistema

- Módulo de Predição: conforme visualizado na figura 5.2, será responsável por prever os parâmetros utilizando a RNA treinada.

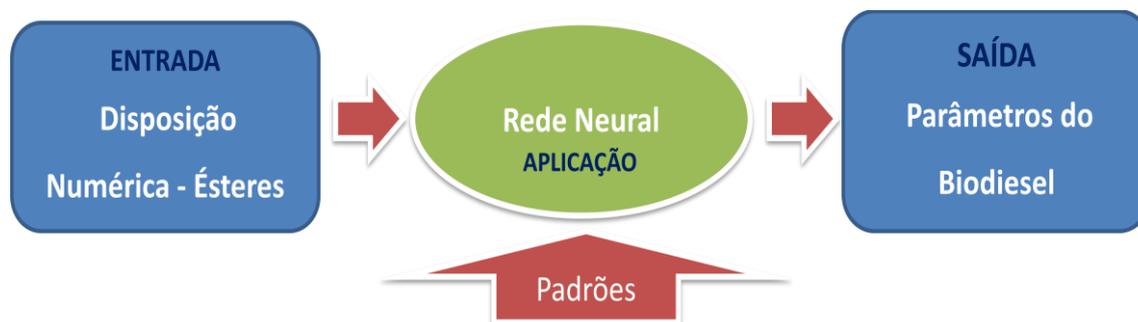


Figura 5.2- Arquitetura do módulo de predição do Sys-PANN

O módulo de treinamento será alimentado por dados em uma tabela contendo a relação ésteres x parâmetros de qualidade. Esta tabela será gerada por um software auxiliar, o OleoData, que irá cadastrar amostras de óleo e biodiesel, conforme na figura 5.1

Após uma explicação geral do software proposto, os dois módulos serão detalhados mais a frente, focando-se no uso da RNA nos módulos. Contudo, será explicado anteriormente o desenvolvimento do OleoData.

5.1. Software auxiliar OleoData

Trabalhando em conjunto com os pesquisadores do LAPQAP e LPQA, o software auxiliar de controle de amostras de óleo e biodiesel foi desenvolvido para facilitar a coleta de dados relacionados às amostras de biodiesel além de permitir a exportação dos dados numa disposição adequada para alimentar o Sys-PANN.

O OleoData será responsável pelos cadastros de amostras de óleos vegetais e de biodiesel pela equipe do laboratório de química. As amostras de óleo serão produzidas em laboratório e receberão análises físico-químicas e posteriormente serão transformadas em biodiesel. Estas amostras de biodiesel também sofrerão análises físico-químicas, no qual tanto as amostras de óleo quanto as de biodiesel serão cadastradas.

O sistema permite armazenar, organizar, padronizar e compartilhar os dados e resultados obtidos no processo de produção de biodiesel a partir de matrizes vegetais.

5.1.1. Modelagem do OleoData

Para a melhor compreensão das funcionalidades deste software, será utilizado um modelo de caso de uso que poderá ser visualizado na figura 5.3. O modelo demonstra apenas as ações principais, por isso apenas um ator³ é utilizado, o analista de laboratório, embora haja também o administrador do sistema.

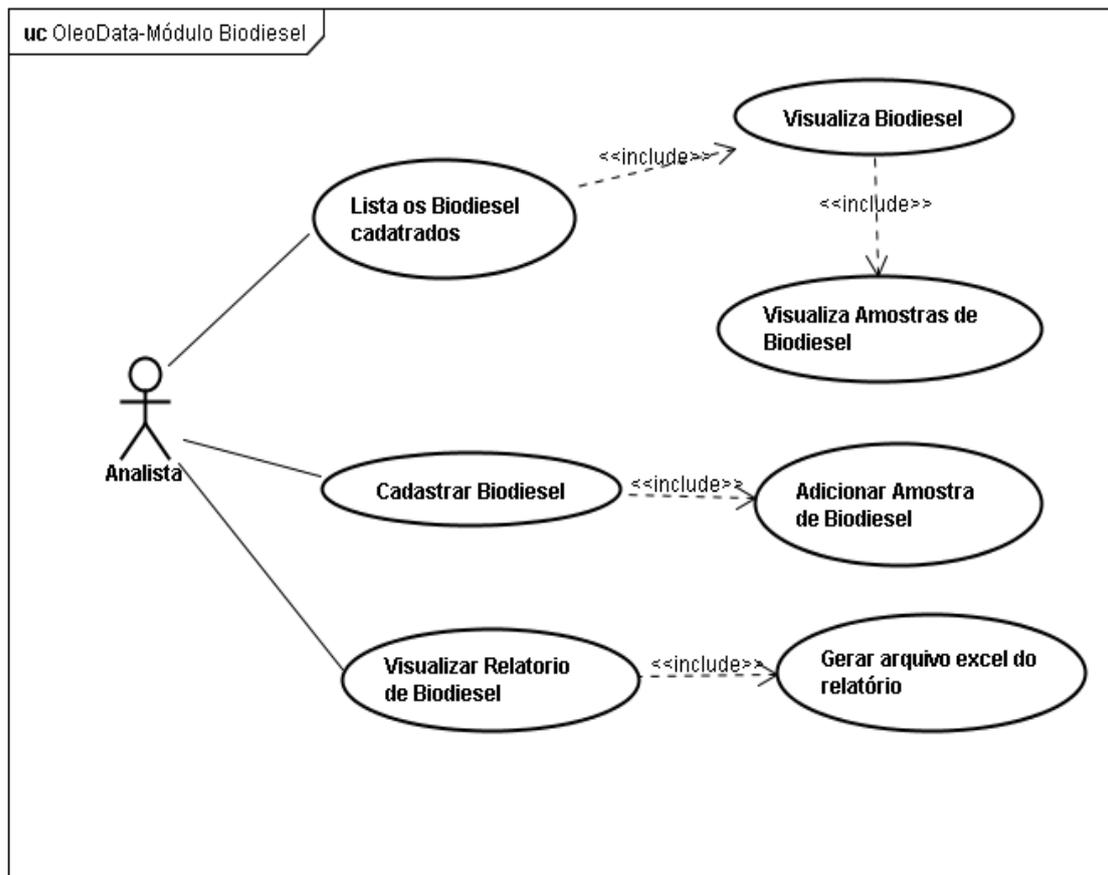


Figura 5.3-Diagrama de Caso de uso do OleoData na visão do analista

O diagrama na figura 5.3 está voltado para as ações do analista nos casos de uso que envolve Biodiesel, o qual é foco do trabalho. O analista é o

³ Ator representa uma ou mais funções de usuário no sistema

principal usuário do sistema, pois a ele fica designados as funções matrizes do OleoData. O administrador possui as mesmas funções do analista somado ao de administrar os usuários.

O analista terá a função de, após produção do biodiesel, cadastrar este biodiesel e posteriormente suas amostras, que serão analisadas físico-quimicamente e terão seus resultados também cadastrados. Após este cadastro, será possível visualizar o que ele e outros analistas cadastraram. Posteriormente, ele poderá acessar o relatório de todos os cadastros em uma tabela, que poderá ser exportada em Excel. Esta tabela é o principal produto do sistema para este trabalho, pois ela será utilizada pelo Sys-PANN.

5.1.2. Implementação do OleoData

Pensando no compartilhamento de dados entre instituições de pesquisa e na portabilidade para se acessar o sistema de qualquer computador, o OleoData foi implementado para o ambiente Web, utilizando a linguagem de programação PHP e JavaScript, e o banco de dados (BD) MySQL. Toda a aplicação e o BD serão alocados em servidor no LPQA e disponibilizados na web para ser acessado de qualquer computador com acesso a rede mundial de computadores.

O PHP foi utilizado por ser de fácil implementação e ser uma linguagem interpretada pelo lado do servidor voltada para Web. Como ele é um sistema que será utilizado por diversos analistas sem a necessidade de um grande domínio de informática, a linguagem JavaScript é utilizada para facilitar o entendimento e utilização do sistema através de seus scripts voltados a interface do sistema. As informações cadastradas e geradas pelo aplicativo são armazenadas num BD MySql.

A apresentação do sistema será concentrada nas funções relacionadas com Biodiesel. A pagina inicial do Sistema pode ser visualizada na figura 5.4



Login	Oleodata
Email: <input type="text"/>	<p>O Sistema Oleodata é um trabalho desenvolvido pelo laboratório LSI (Laboratório de Sistemas Inteligentes) com parceria entre os laboratórios LAPQAP/LPQA (Laboratório de Análises e Pesquisa em Química Analítica de Petróleo / Laboratório de Pesquisa em Química Analítica). O sistema permite armazenar, organizar, padronizar e compartilhar os dados e resultados obtidos no processo de produção de biodiesel a partir de matrizes vegetais.</p> <p>A principal ideia do Oleodata é fornecer dados sobre o processo de produção de biodiesel para facilitar a produção científica a pesquisadores, professores, alunos e outros interessados.</p> <p>Clique aqui e faça parte do projeto.</p>
Senha: <input type="text"/>	
<input type="button" value="Logar"/> <input type="button" value="Cancelar"/>	
Criar um Usuário	

©2012 LAPQAP/LPQA - Todos os direitos reservados.
Desenvolvido pelo LSI.

Figura 5.4-Tela Inicial do OleoData

A pagina inicial do OleoData é responsável por realizar o login de usuário, no qual ele apresenta seu nome de usuário e senha no formulário. O aplicativo identifica o tipo de usuário e oferece os recursos permitidos ao perfil em questão. Nesta página, existe a possibilidade de um usuário não cadastrado solicitar a criação de cadastro, ficando a cargo do administrador, comunicar se o usuário será analista ou administrador.



Componentes Relatórios Avançados Usuário

Lista de Biodiesel

... Adicionar Biodiesel ...

Nº	Data de Cadastro	Identificador	Óleo de origem	Ação
0	04/02/2013	BD0DE0A0	DE0 - Amostra 0	EXCLUIR
1	03/01/2013	BD0BA1A1	BA1 - Amostra 1	EXCLUIR
2	09/12/2012	BD-BA0	BA1 - Amostra 3	EXCLUIR
3	09/12/2012	BD-MA1	MA1 - Amostra 0	EXCLUIR

©2012 LAPQAP/LPQA - Todos os direitos reservados.
Desenvolvido pelo LSI.

Figura 5. 5- Lista de Biodiesel cadastrados no OleoData

A figura 5.5 apresenta a tela inicial do sistema após o login. No menu horizontal, há a opção *Componentes* que possui duas opções: Oleo ou Biodiesel. A escolha de uma delas gera a lista de seus respectivos cadastrados, a exemplo nesta figura, aparece uma lista de biodiesel. Para se visualizar informações de um item da lista, basta clicar no biodiesel listado.

Caso o analista deseje cadastrar um novo biodiesel produzido ele deverá clicar na opção <<Adicionar Biodiesel>> e então será redirecionado para a pagina de cadastro de Biodiesel.

Cadastro de Biodiesel

Óleo:
Selecione um Óleo... ?

Amostra do Óleo:
Carregando amostra... ?

Volume (mL):
?

Massa (g):
?

Data Inicial:
?

Data Final:
?

<< Voltar Cadastrar

Identificador de Biodiesel

O Identificador do Biodiesel foi gerado Automaticamente: **BD0DE0A0** ?

... Sair ...

Figura 5.6- Telas de cadastro de Biodiesel e geração de ID automático ao Biodiesel no OleoData

A figura 5.6 apresenta o formulário presente na tela de cadastro de biodiesel. O analista deverá informar o óleo vegetal e de qual amostra desse óleo que se originou esse biodiesel. Essas duas informações devem ser previamente cadastradas no setor de Óleo do OleoData.

Além de sua origem, deverá ser informado também o volume e massa de biodiesel produzido, assim como sua data de produção(incluindo o início e fim). Após o cadastro, o sistema gera um código identificador automaticamente, como visualizado na figura 5.6.

Depois de criar o biodiesel, o analista deve pesquisar em cima de amostras desse biodiesel que sofrerão análises físico-químicas e de composição, para assim cadastrar no sistema.

Adicionar Percentual de Massa de Éster...

Perc. de massa: Seleccione um ester... :

Perc. de massa: Seleccione um ester... :

Análise Físico-Química:

Propriedade cetano

Resultado:

Data do Ensaio:

Propriedade viscosidade

Resultado:

Data do Ensaio:

Propriedade acidez

Resultado:

Data do Ensaio:

Figura 5.7- Tela de Cadastro de Amostras de Biodiesel no OleoData

A figura 5.7 apresenta os formulários presentes na tela de cadastro de amostra de biodiesel. Além desses, outros dados são inseridos como volume e data das análises da amostra. Porém, os itens mais importantes estão nos formulários de cadastro de análise físico-química e de composição.

Nesta janela, o analista deverá informar o ésteres que compõem a amostra assim como suas respectivas quantidades. Depois, deverá apresentar os valores das análises físico-químicas.

Tendo todos os dados cadastrados, os usuários poderão ver as informações de cada biodiesel e amostra em janelas similares. Se o analista desejar ver todos os biodiesel e suas amostras com seus respectivos dados, poderá acessá-lo através da opção <<Relatório>> no menu principal.

Relatorio de Biodiesel

Gerar Excel ...

Data de Cadastro	Identificador	Óleo de origem	Volume
26/09/2013	BD0DE0A0	DE0 - Amostra 0	
	+	Amostra 0	20
04/02/2013	BD0DE0A0	DE0 - Amostra 0	23
	+	Amostra 0	2
03/01/2013	BD0BA1A1	BA1 - Amostra 1	40
	+	Amostra 1	100
09/12/2012	BD-BA0	BA1 - Amostra 3	
	+	Amostra 0	50
	+	Amostra 1	30
	-	Amostra 2	10
		Analise de Composição	
		Ester	Volume
		acido 2	5
		acido 3	3
		acido 6	7
		Analise de Propriedades Quimica	
		Ester	Volume
		acidez	
		cetano	33
		urano	56
		viscosidade	24
09/12/2012	BD-MA1	MA1 - Amostra 0	

Figura 5.8- Tela de Relatório de Biodiesel no Oleodata

A figura 5.8 apresenta a tela responsável por emitir, em uma só janela, todas as informações básicas cadastradas de biodiesel e suas amostras. Como se pode visualizar, o biodiesel é sinalizado em verde e suas amostras cadastradas em laranja. Para visualizar a composição de ésteres e os parâmetros de qualidade (rosa), basta clicar no [+] presente em cada amostra.

Esta janela também permite ao usuário gerar um arquivo com os dados que estão no relatório e realizar o download dele. Esta é a função mais importante para o trabalho atual do OleoData, pois são os dados desse arquivo que serão utilizados pelo Sys-PANN para treinar a RNA.

O arquivo gerado deverá ser um arquivo de excel na extensão xls (excel 2003). Não se pode utilizar a extensãoxlsx (excel 2007) pois o Sys-PANN não lê excel desta extensão.

Tendo finalizado o OleoData, os dados poderão ser disponibilizados para análise pelo Sys-PANN. Antes de implementar o Sys-PANN, foi necessário realizar uma pesquisa detalhada para descobrir a melhor RNA e como implementa-la.

5.2. Sys-PANN – Módulo de treinamento

O sistema proposto SysPANN deverá prever os parâmetros de qualidade de Biodiesel a partir da análise de perfis de ésteres da amostra. Para isso, o sistema irá utilizar Redes Neurais Artificiais para encontrar a relação entre o quantitativo de ésteres e parâmetros das propriedades de biodiesel. Para encontrar essa relação, a RNA tentará realizar uma de suas tarefas, a *Apróximação de Funções*.

Por isso, o sistema foi dividido em dois módulos: de treinamento, como foi visualizado na figura 5.1, e o de predição, visualizado na figura 5.2. Esta seção tratará do módulo de treinamento da RNA.

Para se entender melhor o que o módulo de treinamento deve fazer, uma modelagem do sistema é recomendada. Diferente do OleoData que possui múltiplas funções e se utilizou um diagrama de caso de uso, neste caso, como o módulo possui apenas uma função – a de retornar uma RNA treinada – será utilizado um diagrama de sequência⁴ para explicar os passos dessa operação.

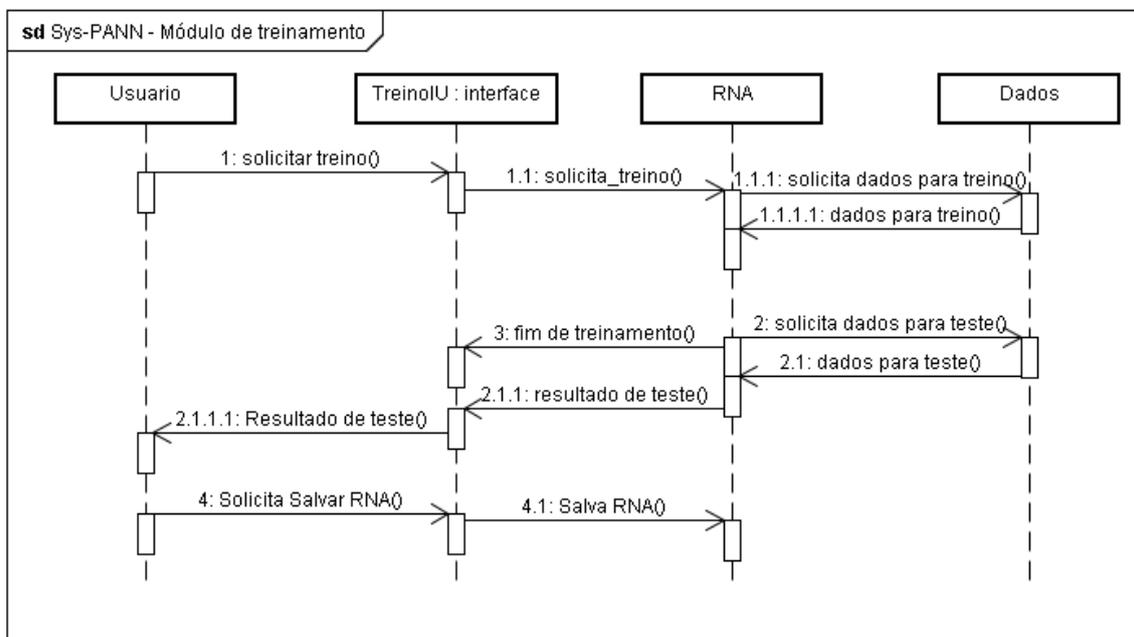


Figura 5.9-Diagrama de Sequência do módulo de treinamento do Sys-PANN

⁴ Demonstra as interações entre diferentes objetos na execução de uma operação, destacando a sequência em que as diversas operações são executadas.

A figura 5.9 apresenta um diagrama de sequência do que se deseja do módulo de treinamento do Sys-PANN. Tendo como autor, um usuário qualquer do sistema, ele irá visualizar o sistema através da interface de treinamento representado no diagrama por TreinoIU. O usuário solicitará via interface (TreinoIU) o treinamento da RNA esperada. Todo processamento da rede neural é representada pelo objeto RNA.

Com a solicitação de treinamento ao objeto RNA, esse objeto irá solicitar os dados necessários para o treinamento, o conjunto de padrões presentes no arquivo que é representado pelo elemento Dados. Com os padrões de treinamento enviados a RNA, o treinamento finalmente se inicia.

Finalizado o treinamento, o objeto RNA solicita ao objeto Dados o conjunto de padrões de teste para verificar se a rede foi treinada. Tendo esses padrões, o RNA processa e apresenta o resultado ao TreinoIU para que possa ser visualizado pelo usuário.

Se o usuário aprovar o que foi lhe apresentado, deverá solicitar a função de salvar a rede ao objeto TreinoIU, que então a salva para que esta rede possa ser posteriormente utilizada no módulo de predição.

Como se pode observar, mesmo que o módulo de treinamento desempenhe apenas uma função principal, ele terá diversas operações para realizar com a RNA. Compreendido as operações que este módulo irá desempenhar, pode-se concentrar agora na implementação da RNA.

Porém, antes de se implementar a RNA que será empregada no sistema deve-se planeja-la. Para realizar o melhor desenvolvimento, foram utilizados os procedimentos de desenvolvimento de projeto de RNA, conforme apresentados no capítulo três. A evolução do desenvolvimento da RNA deste trabalho será descrita conforme os procedimentos a seguir.

5.2.1. Definição e Configuração da RNA

Este 1º passo é um momento crucial no desenvolvimento da rede, no qual se definirá a rede neural mais adequada baseando-se no problema

proposto. Para isso, será seguida a lista de escolhas para se chegar na RNA desejada.

- I. Escolher o paradigma neural apropriado;
- II. Escolher o modelo apropriado;
- III. Escolher o algoritmo de treinamento e suas configurações ;e
- IV. Escolher as funções de ativação

Inicialmente, deve-se escolher o paradigma neural e o capítulo dois apresentou 6 modelos de RNA. Considerando-se que o problema necessita prever uma propriedade a partir dos níveis de ésteres no biodiesel, verifica-se que se deseja encontrar uma função matemática cujas variáveis da equação são os ésteres e o resultado é o valor propriedade.

Encontrar uma função matemática corresponde ao problema de se aproximar função. O modelo mais adequado e que também é conhecido como aproximador universal de funções, é o modelo Perceptron de multi-camadas.

Dessa forma, as opções I e II foram respondidas com a definição de utilização do paradigma de treinamento Supervisionado e o do modelo Perceptron de Multi-Camadas.

Por se ter escolhido o modelo MLP, agora se tem que definir também o número de camadas escondidas necessárias e o número de neurônios que se deseja. A última seção do capítulo 4 trata sobre as melhores maneiras de se encontrar esses valores.

Para a definição do número de camadas, será levado em consideração as pesquisas de Cybenko (1988) e (1989) que define o número de camadas ocultas. Como ele define que duas camadas intermediárias são o necessário para aproximar qualquer função matemática e se desconhece qual o tipo de função que se deseja encontrar, então a RNA esperada terá 2 camadas ocultas.

Após se definir o número de camadas ocultas da RNA, pode-se definir o número de neurônio de cada uma das 4 camadas. Vale ressaltar que a camada de entrada será tratada como se tivesse neurônios, esses porém, sem processamento.

A camada de entrada deverá receber os dados externos que no caso correspondem ao quantitativo de ésteres. Cada neurônio corresponderá a um éster constituindo um percentual de sua quantidade na amostra de

combustível. Para o teste do protótipo serão utilizados 13 ésteres que serão apresentados posteriormente. Dessa forma, a camada de entrada será composta por 13 neurônios.

A camada de saída deverá devolver os resultados esperados, que no caso são os parâmetros. Em busca de uma melhor exatidão nos resultados e simplificação do problema, será utilizado um parâmetro por rede. Assim, a camada de saída terá um único neurônio.

Para a definição das camadas ocultas, há certa dificuldade de se encontrar os números, pois, como Almeida(2001) e Braga(2007) afirmaram, a sua definição é uma arte de tentativa e erro. E apesar de técnicas para definir este número, nenhuma tem sua eficácia comprovada.

Devido a grande dificuldade de se encontrar a quantidade de neurônios para a camada oculta, optou-se então para uma opção mais automática: testar todas as opções de combinações de números de neurônios das camadas ocultas, com intuito de se encontrar o melhor resultado.

Dessa forma, o que se propõe é que diversos treinamentos sejam realizados automaticamente e se verifique qual combinação de neurônios teve o menor erro. Assim, ainda visualizando o diagrama de sequência da figura 5.9, o usuário antes de solicitar o treino, iria informar a variação do número de neurônios que se deveria ter para que cada combinação represente uma rede e assim se descubra a melhor configuração.

A RNA discutida assumirá a seguinte configuração:

- Camada de entrada: 13 neurônios;
- Camada oculta 1: x neurônios;
- Camada oculta 2: y neurônios;
- Camada de saída: 1 neurônio.

Finalizado a definição do modelo e sua configuração, agora é necessário escolher o algoritmo de treinamento. Para isso, decidiu-se escolher entre uma das 3 variações de back-propagation descritas no capítulo anterior.

- BackPropagation com momentum
- Resilient Propagation
- Levenberg-Marquardt

Elas serão implementadas, testadas e comparadas para que se verifique qual possui o melhor desempenho. A melhor será adotada como algoritmo padrão.

O ultimo passo é escolher a função de ativação adequada para os neurônios MCP. Das funções citadas, as mais adequadas para trabalhar com o modelo escolhido são as funções diferenciáveis, que são:

- Logística-sigmóide – utiliza valores entre [0,1];
- Hiperbólica-tangente – utiliza valores entre [-1, 1]

Para se escolher uma dessas duas, deve-se verificar como os dados estão dispostos. Se o conjunto de dados utilizado possuir mediana, deve se adotar a hiperbólica tangente, senão a Logística-sigmóide.

Após definir a arquitetura, pode-se trabalhar com os dados que serão utilizados para treinamento.

5.2.2. Preparação dos Dados

Neste passo, os dados relativos ao problema, no caso, os padrões são coletados em quantidade e qualidade suficientes para se realizar um bom treinamento da rede neural. De posse dos dados, deve-se dividir o grupo em dois: dados de treinamento, representando 80% do grupo total e os dados de teste representando 20% do total.

Para o teste deste protótipo, o LPQA concedeu um conjunto de dados referentes a composição de amostras de biodiesel e suas análises físico-químicas. Para este caso, cada amostra com sua análise físico-química representa um padrão.

O conjunto de dados cedido é constituído de 98 padrões (amostras), sendo 13 ésteres e 3 parâmetros. Os ésteres que serão utilizados serão:

- C8:0
- C10:0
- C12:0
- C14:0
- C16:0
- C18:0

- C18:1
- C18:2
- C18:3
- C18:1OH
- C20:0
- C20:1
- C22:1

Os parâmetros que serão utilizados:

- Índice de lodo,
- Massa específica e
- Viscosidade.

Para padronizar os dados e facilitar o uso no sistema, os dados foram cadastrados no OleoData e posteriormente foi gerado um arquivo de Excel contendo essa relação para que pudesse ser utilizada no treinamento.

Após a coleta dos padrões, é necessário agora pré-processar esses dados para deixá-los no formato utilizado pela rede, em especial pelas funções de ativação. Isso acontece pois a função de ativação depende da variação dos dados, como foi explicado na seção anterior.

Levando em consideração os ésteres, que representam porcentagem de composição, então seus valores devem variar de 0 a 100, com sua mediana fixa em 50. Já os parâmetros, não possuem valores fixos de variação.

Tomando como exemplo o parâmetro de massa específica, as amostras, por determinação da ANP, devem ter valores entre 850 e 900 para serem consideradas dentro dos padrões. Porém, a massa específica fora dos padrões não possuem limite nem inferior e nem superior, assim podendo ter valor indeterminado.

Sem valores superiores e inferiores, é difícil de dizer a mediana que esses valores podem assumir. Assim, se for para normalizar em $[-1, 1]$, fica impossível se determinar a mediana para 0. Pode-se determinar o valor máximo e mínimo do grupo, porém se for inserido o valor fora do grupo, a normalização ocasiona falha.

Por isso, o mais adequado é normalizar entre [0, 1] e utilizar a equação 5.1 de normalização:

$$x^n = \frac{x - x^{min}}{x^{max} - x^{min}} \quad (5.1)$$

onde x é um valor a ser normalizado, x^{min} é o menor valor do grupo de dados e x^{max} o maior valor. Nesta situação, x^{min} será 0 e x^{max} será 100.

Para devolver a resposta da RNA, basta desnormalizar realizando a ação inversa, a equação 5.2:

$$x = x^n(x^{max} - x^{min}) + x^{min} \quad (5.2)$$

Como os dados serão utilizados entre 0 e 1, a função mais adequada é a Logística-sigmóide. Com a RNA definida e os dados preparados para serem treinados, agora se pode implementar a rede e treinar os dados.

5.1.1.Implementação e Treinamento da RNA

Neste passo do desenvolvimento, a RNA pode ser implementada seguindo as configurações determinadas nos passos anteriores. Após a implementação a rede já poderá ser treinada.

Considerando que o Sys-PANN conterà uma RNA e em conjunto de diversas operações matemáticas, optou-se por implementar o sistema na linguagem de programação Java, devido as diversas bibliotecas que dão suporte as necessidades do sistema. Para facilitar o desenvolvimento da RNA, decidiu-se utilizar o framework Encog.

Encog é um Framework de IA para Java e .NET com foco em RNA (Heaton, 2010). Ele foi criado por Jeff Heaton a fim de ter uma biblioteca mais robusta que a JOONE, até então, a única para IA. Este framework recebeu melhor avaliação da comunidade acadêmica que os frameworks Neuroph e JOONE.

Este framework é muito útil, pois ele implementa boa parte da rede, deixando ao usuário a configuração configurar a quantidade de camadas,

números de neurônios, a função desejada, o algoritmo de treinamento desejado e implementação o ponto de parada.

O Encog implementa sua rede utilizando o classe *BasicNetwork()*. A partir daí, a rede pode ser configurada com seus métodos. O trecho de código na figura 5.10 demonstra a implementação da rede e das camadas no Encog a partir do objeto *network*

```
network = new BasicNetwork();
// camada de entrada
network.addLayer(new BasicLayer(neuronio_entrada));
// camadas ocultas
network.addLayer(new BasicLayer(new ActivationSigmoid(), true, neuronio_oculta1));
network.addLayer(new BasicLayer(new ActivationSigmoid(), true, neuronio_oculta2));
// camada de saida
network.addLayer(new BasicLayer(new ActivationSigmoid(), true, neuronio_saida));
```

Figura 5.10- Código de implementação no encog das RNA

No código presente na figura 5.10 se pode notar que, como se havia projetado, a rede *network* foi implementada com quatro camadas (*new BasicLayer()*). As duas ocultas e a camada de saída, como são neurônios do tipo MCP, foram instanciadas com a função de ativação Sigmoid (*new ActivationSigmoid()*).

Para melhor compreender como foi implementado o sistema com RNA, a figura 5.11 apresenta o diagrama de classes. Este diagrama apresenta uma classe em destaque, a classe **Rna**. É nela que todo processamento de uma rede neural acontece e a biblioteca do Encog é utilizada. Apesar dela ser a principal, quem gerencia todo o procedimento é a classe **PrincipalRna**.

Para cada rede neural configurada, a classe **PrincipalRna** instancia um objeto de **Rna**. Após instanciar, a RNA precisa de dados para o treinamento que são derivados da classe **DadoXls**. É ele que extrai os dados do arquivo xls e posteriormente passa a classe que normaliza os dados, a classe **Normalizacao**.

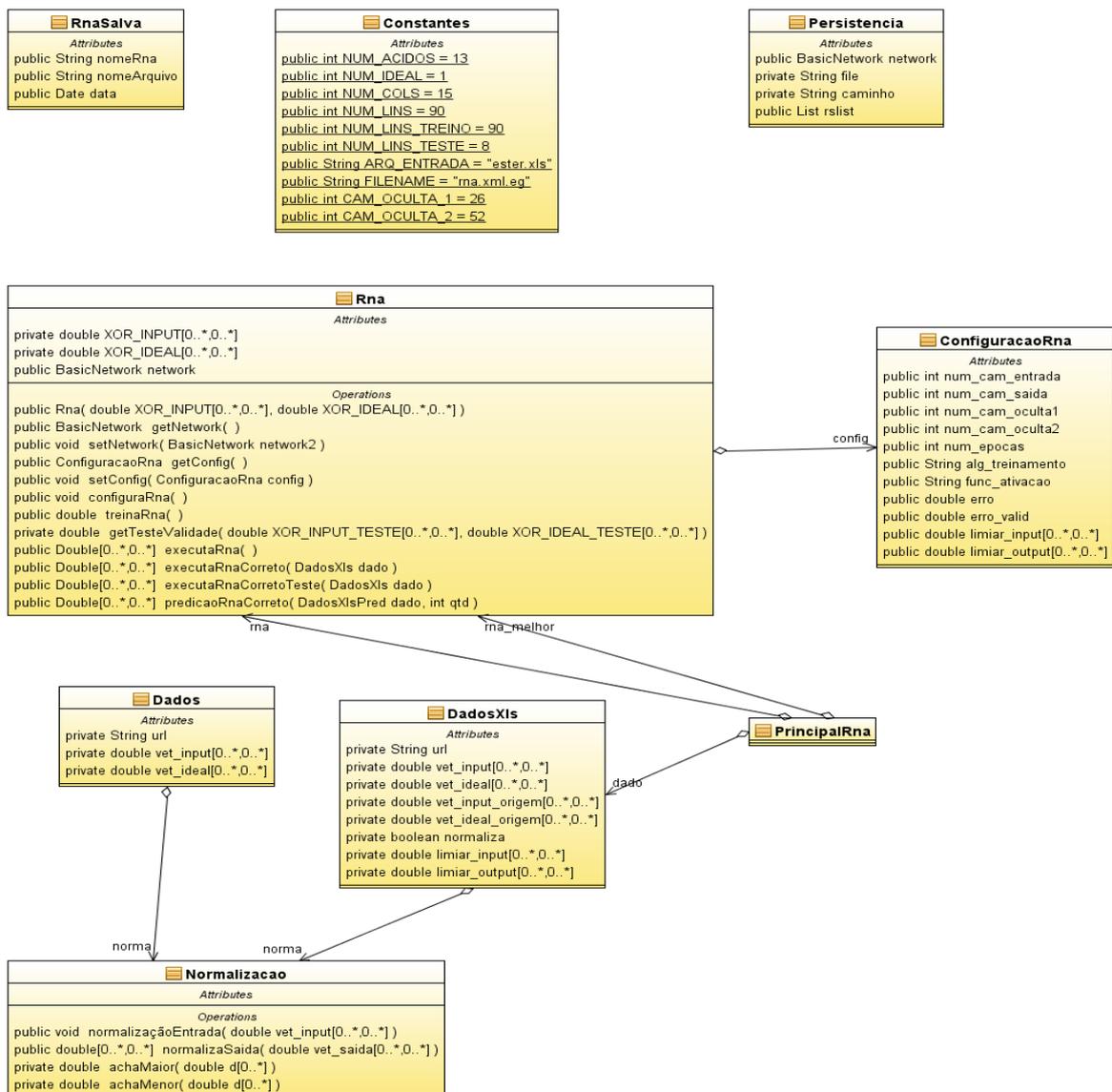


Figura 5.11- Diagrama de Classes da RNA

A **Normalizacao** utiliza a equação 5.1 e 5.2 para normalizar e desnormalizar os dados provenientes de **DadoXls**. Posteriormente, **PrincipalRna** repassa os dados a **Rna** para que possa realizar o treino.

Cada RNA treinada, fica com sua configuração gravada nos objetos de classe **ConfiguracaoRna**. Com esta classe fica facil de salvar posteriormente a configuração da RNA desejada.

Quando se deseja salvar a rede, classe **Persistencia** é a responsável por persistir a configuração da RNA e a RNA treinada. Ambos são salvos em estruturas de dados XML. Esse formato foi escolhido pela facilidade

de se trabalhar com XML no Java e de não haver a necessidade utilizar um SGBD⁵, tornando o sistema mais leve.

A figura 5.12 apresenta como os dados da classe **ConfiguracaoRna** ficam arranjados no arquivo XML. É possível verificar no XML ficam salvos o número de neurônios da camada de entrada, saída e ocultas, o erro gerado pela rede e quantas épocas foram necessárias.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<simulacao.ConfiguracaoRna>
  <num_cam_entrada>13</num_cam_entrada>
  <num_cam_saida>1</num_cam_saida>
  <num_cam_oculta1>26</num_cam_oculta1>
  <num_cam_oculta2>26</num_cam_oculta2>
  <num_epocas>391</num_epocas>
  <erro>0.03681971447674632</erro>
  <erro_valid>0.025348028006046963</erro_valid>
</simulacao.ConfiguracaoRna>
```

Figura 5.12-XML da configuração de uma RNA

Inicialmente, foram testados três algoritmos de treinamento: BackPropagation com momentum, Resilient Propagation e Levenberg-Marquardt. Os três são implementados pelo Encog ficando a necessidade apenas de determinar valor do momentum necessário para o BackPropagation.

Os testes foram realizados a fim de descobrir qual dos três possuía melhor desempenho para os dados utilizados. Para isso, os três algoritmos foram testados em condições iguais de configuração de rede que consistia em:

- Número de Neurônios oculto-1: 15
- Número de Neurônios de entrada-2: 20

Os testes foram realizados em três seções para se descobrir quanto tempo e épocas eram utilizadas para se atingir um determinado nível de erro de treinamento. Cada nível de erro correspondeu a uma seção. Os níveis de erro utilizados foram: 0,1; 0.09 e 0.008. Os resultados dos testes podem ser visualizados na tabela 5.1.

⁵ Sistema Gerenciador de Banco de Dados

Tabela 5.1 - Comparativo de Algoritmos: BP, LMA e RProp

Erro de treino	0,1			0,09			0,08		
	BP	LMA	RProp	BP	LMA	RProp	BP	LMA	RProp
Tempo	3.8 seg	84 seg	0.09seg	4.5 seg	2.34min	0.120	10.0 seg	4.04min	0.170
Epocas	1938	355	30	2457	480	40	4000	780	45

Como esperado, o Back Propagation sempre apresentou pouco tempo, porem necessitando de muitas épocas para compensar as poucas alterações por épocas que ela apresenta por ser um algoritmo simples. O Levenberg-Marquardt é um algoritmo muito recomendado e demonstra que possui uma melhora em relação ao BP pelo fato de necessitar de um número menor de épocas mas como esperado, seu processamento por época é muito custoso devido a necessidade de sempre calcular a matriz jacobiana para cada neurônio.

Dos três, o que teve resultados mais satisfatórios foi o algoritmo de treinamento Resilient Propagation, apresentando sempre o menor tempo e o menor número de épocas por isso ele que será utilizado no Sys-PANN.

Com relação a dinâmica de treinamento e inicialização dos pesos, o Encog é inflexível nestes dois itens. Na dinâmica, ele trabalha com Épocas, no qual os pesos são atualizados a cada ciclo. E os pesos, ele inicializa de forma randômica.

Após ter se definido todos os itens de configuração da RNA, falta apenas a condição de parada. O Encog não possui um método próprio para isso, sendo necessário implementa-lo.

Considerando os estudos e dicas apresentados na ultima seção do capítulo 4, é preferível trabalhar com validação cruzada. Para isso foi necessário acrescentar mais um grupo aos dados, o grupo de validação, Assim, a distribuição do quantitativo de dados por grupo ficou:

- Grupo de treinamento – 70% dos dados totais,
- Grupo de teste – 10% dos dados totais,
- Grupo de validação – 20% dos dados totais.

Para esta forma de condição de parada, o treinamento acontece apenas com o grupo de treinamento, e a cada época, a RNA é testada com o grupo de validação, gerando-se assim o erro de treinamento e de validação. Vale lembrar que o grupo de teste e de validação não pode ser utilizado no treinamento.

Teoricamente, quando se inicia o treinamento de uma rede, ela começa primeiro generalizando para qualquer dado, apresentando assim um erro de validação baixo e de treinamento alto. Com o decorrer do treinamento, a rede começa a se especializar nos dados de treinamento e a não mais generalizar, então o erro de treinamento diminui e de validação aumenta.

A escolha é paralisar o treinamento antes da rede se especializar, ou seja, quando o erro começar a aumentar. Tem-se que se descobrir quando ocorre o erro mínimo global, que corresponde ao menor erro que a validação vai apresentar antes do erro aumentar.

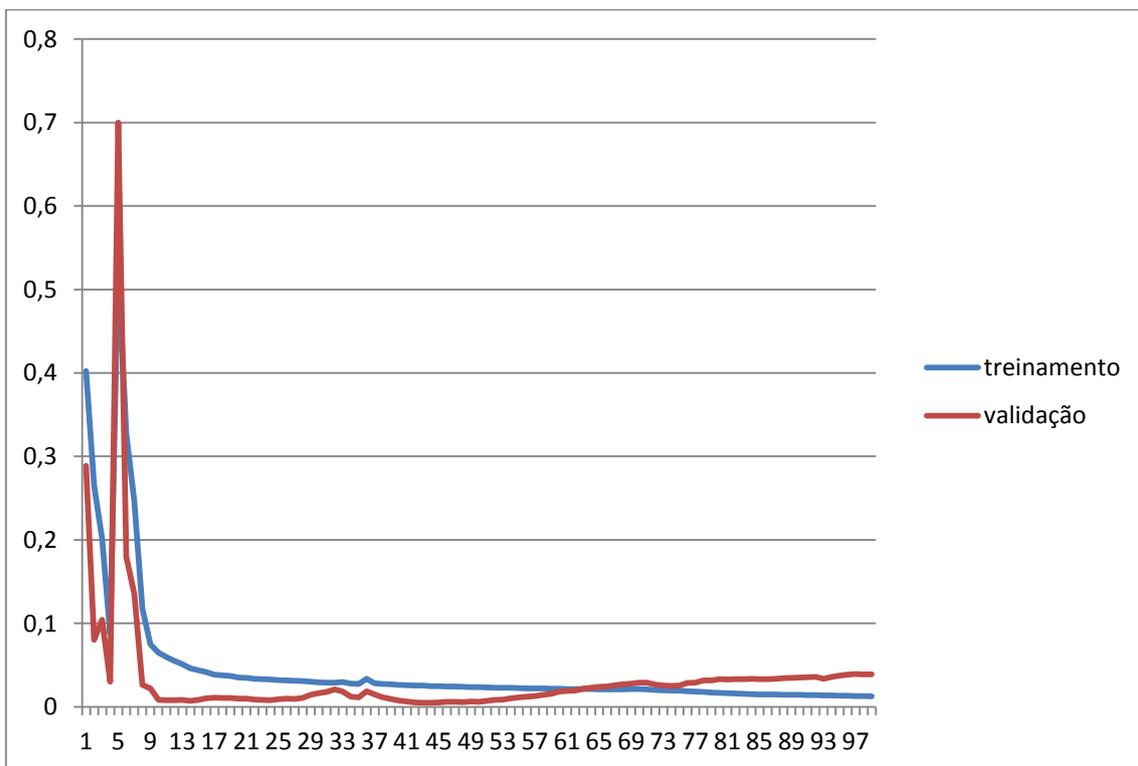


Figura 5.13- Gráfico de teste de validação cruzada

Porém, através de testes, a curva dos erros não são suaves o suficiente para se detectar quando acontece um mínimo global. O gráfico 5.1

representa um desses testes e demonstra que tanto o treinamento quanto a validação possuem um início ruidoso e posteriormente eles se normalizam.

Seguindo o exemplo do teste, a intenção seria parar o treinamento no menor erro da validação, sendo no gráfico indicado na época 25. Devido a esse início ruidoso decidiu-se que a validação iria ocorrer apenas a partir da época 20. Depois disso, seria verificado a cada época, os três últimos erros de cada época para se constatar que os erros estão aumentando. Assim, o treinamento só vai parar quando último erro for maior que o penúltimo e o penúltimo maior que o antepenúltimo.

Tendo se implementado o motor do Sys-PANN, a RNA, ele pode ser implementado por completo agora. A figura 5.14 apresenta a janela inicial do sistema. Nesta janela inicial que se escolhe qual módulo que se deseja utilizar, se é o de treinamento ou o de predição.



Figura 5.14- Tela inicial do Sys-PANN

Após escolher a opção de gerar uma nova rede Neural, a janela correspondente ao módulo de treinamento aparece. Esta janela pode ser visualizada na figura 5.15. Ela contém um formulário para configuração da rede que se deseja encontrar.

Inicialmente, se informa o número de neurônios da camada de entrada e da de saída. Na definição do número de neurônios das camadas ocultas, é necessário informar o espaço numérico da variação de neurônios.

Assim, se for informado de 25 a 27 nos campos das camadas ocultas, então serão geradas 9 redes nas seguintes configurações:

- 13-25-25-1
- 13-25-26-1
- 13-25-27-1
- 13-26-25-1
- 13-26-26-1
- 13-26-27-1
- 13-27-25-1
- 13-27-26-1
- 13-27-27-1

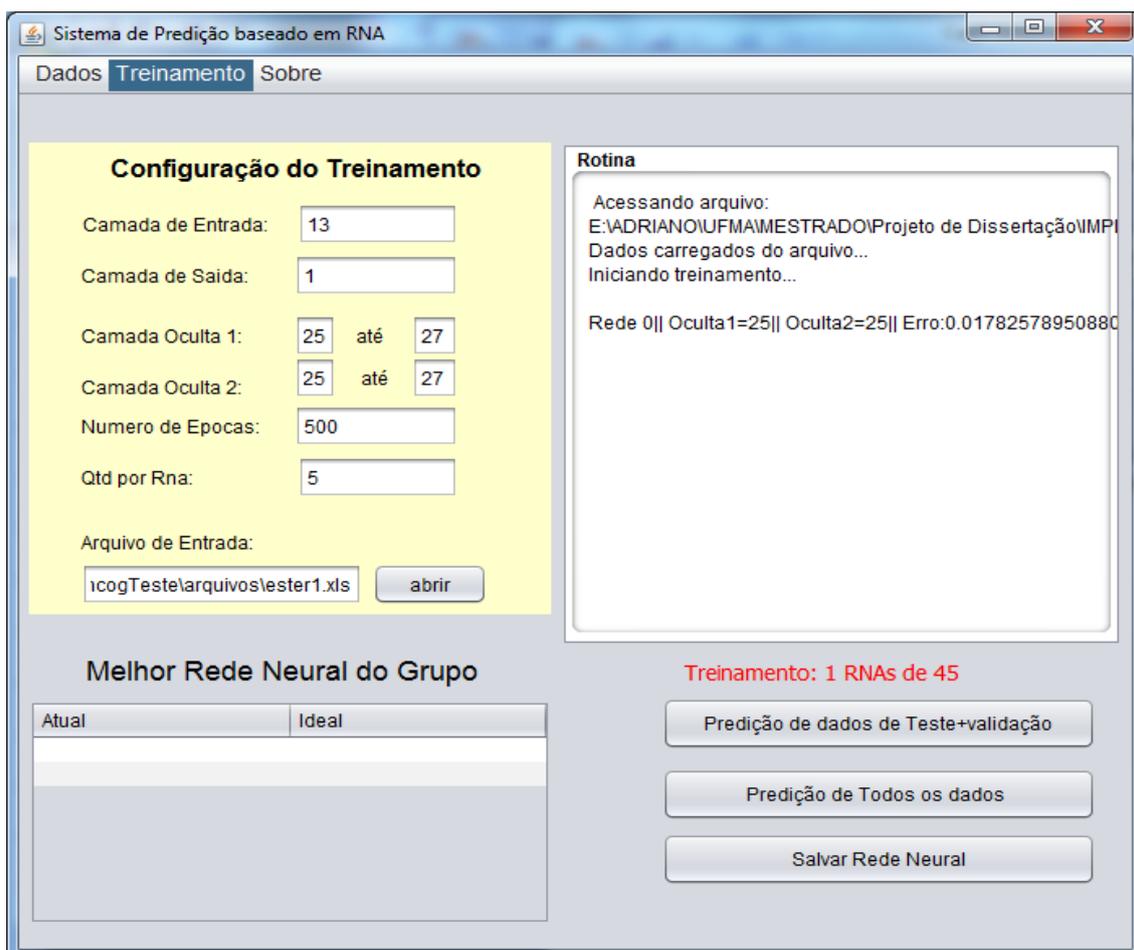


Figura 5.15-Modulo de treinamento do Sys-PANN

Após se configurar, deve-se informar o máximo de épocas que se pode treinar. Posteriormente, deve-se informar o número de vezes que uma dada RNA de configuração x deve ser treinada. Isso foi necessário pois, como os pesos são iniciados randomicamente, duas redes com mesma configuração terão iniciação de pesos diferentes, levando as RNAs produzirem resultados diferentes.

Por ultimo, se informa o arquivo excel que os dados estão guardados para ser utilizado no treinamento. Após o preenchimento de todas as informações, clica-se em <<Treinamento>> para se iniciar o treinamento das redes.

A janela de rotina apresenta a evolução dos treinamentos e abaixo dela, se informa em vermelho quantas RNA foram treinadas.

Melhor Rede Neural do Grupo

Configuração	valor
Camada de entrada	13
Camada de saída	1
Camada Oculta -1	26
Camada Oculta -2	27
Epocas utilizada	150
Erro de treinamento	0.038767234345748414
Erro de Validação	0.0027104970754671803

Treinamento: FINALIZADO!!!

Predição de dados de Teste+validação

Predição de Todos os dados

Salvar Rede Neural

Figura 5.16-Treino Finalizado no Sys-PANN

A figura 5.16 apresenta a situação da janela quando se finaliza o treino. Quando o treino é finalizado, é apresentada a rede que teve o menor erro e seus dados de configuração são apresentados no quadro da Melhor Rede Neural. O seu desempenho pode ser visualizado clicando no botão <<Predição de dados de Teste + Validação>>. Essa opção abre uma janela que apresenta os parâmetros ideais utilizados no teste e validação com suas respectivas predições.

Se a rede apresentou um bom desempenho, pode-se então salvar a rede para que ela possa ser utilizada pelo módulo de predição, que será apresentado a seguir. Para isso, basta clicar em << Salvar Rede Neural>>. Uma janelinha se abrirá e terá que se dá um nome para cadastrar essa RNA salva.

5.3. Sys-PANN – Módulo de Predição

O módulo de Predição do Sys-PANN é a parte principal do trabalho, pois é nela que vai se realizar a predição dos parâmetros desconhecidos a partir da apresentação da composição química. Para seu funcionamento, é necessário o treinamento antes de alguma RNA no módulo de treinamento.

Este módulo é mais simples, pois sua função é predizer os parâmetros das amostras de biodiesel que forem inseridos nele. A tela do módulo é apresentada na figura 5.17.

Para poder predizer algum parâmetro, o usuário deverá selecionar uma das RNAs salvas do módulo anterior. Essa seleção acontece na lista de RNAs presente no quadro direito da figura 5.17. Depois de selecionada, os dados de configuração da RNA são carregados no formulário de rede Neural Atual. Assim poderá se visualizar os detalhes da RNA e se é ela a desejada.

Importante dizer que se deve carregar a RNA correspondente a determinado parâmetro pois cada rede só pode predizer um tipo de parâmetro.

Após a confirmação de que é a RNA desejada, a predição pode acontecer. Ela é realizada no quadro azul abaixo para que aconteça, os ésteres da amostra devem estar num arquivo excel, assim como foi utilizado no

treinamento, só que sem os parâmetros e então se selecionar o arquivo desejado com os ésteres. Com o arquivo carregado, pode-se clicar em <<PREDIÇÃO>>.

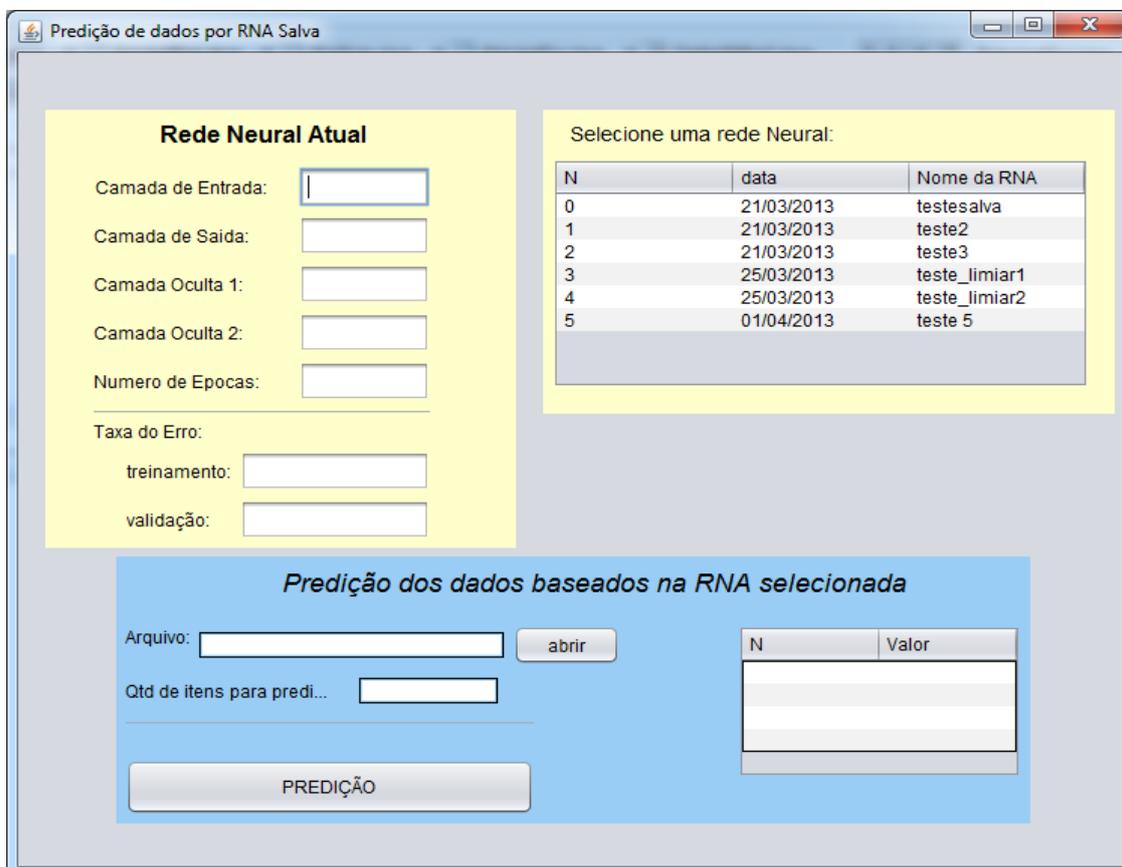


Figura 5.17- módulo de predição do Sys-PANN

Depois de solicitada a predição, o resultado solicitado aparecerá no quadro a direita. E assim, a função de predição se finaliza. Se desejar, é possível prever várias amostras bastando informar no campo Qtd de itens, a quantidade de amostras que necessita prever.

O módulo de predição é simples, pois o trabalho desgastante fica por conta do módulo de treinamento que posteriormente só passa a RNA pronta para ser utilizada para predição.

5.4. Considerações Finais

O Sys-PANN é um software desenvolvido em JAVA utiliza o Framework Encog para facilitar a implementação da RNA que ele utiliza para prever os parâmetros de qualidade do Biodiesel.

Após análises nos capítulos anteriores, ficou definido a configuração que seria utilizada no sistema: uma rede MLP utilizando o algoritmo de treinamento Resilient Propagation. A função de ativação definida foi Logística-Sigmóide.

Devido a dificuldade de se encontrar o número ideal de neurônios, decidiu-se por automatizar essa busca: o sistema irá buscar a melhor configuração numa busca exaustiva. Após definido e implementado, poderá ser testado e os seus resultados serão apresentados no próximo capítulo.

6. RESULTADOS E DISCUSSÕES

Recentemente, o LPQA começou a utilizar o software Statística 10, desenvolvido pela StatSoft®, para realizar seus estudos estatísticos e preditivos. Este software é composto de um módulo com ferramentas estatísticas e outro com mineração de dados. Dentro deste último módulo, encontram-se as RNAs.

O Statística 10 é um software amplamente divulgado e utilizado por diversas empresas e instituições, e por isso seus resultados serão utilizados para validar o funcionamento do Sys-PANN.

Semelhantemente ao Sys-PANN, o Statística 10 testa diversas configurações de RNAs para encontrar a mais adequada. Porém, o seu uso é limitado a apenas uma camada oculta, em oposição ao Sys-PANN que utiliza duas camadas ocultas.

Para validar o Sys-PANN, decidiu-se comparar seu desempenho ao Statística 10, o qual recebeu o mesmo grupo de dados que foi utilizado pelo Sys-PANN focando no teste de desempenho ao grupo de dados do parâmetro de qualidade Índice de Iodo. A tabela 6.1 apresenta as melhores configurações que cada software encontrou.

Tabela 6.1- Configurações dos Softwares Sys-PANN e Statística

	Sys-PANN	Statística
Configuração	13-25-25-1	13-10-1
Função de Transferência	Logística - Sigmoide	Tangente-Hiperbolica
Erro de Treinamento	0,044	0,0847

Para comparar o desempenho dos dois softwares, será utilizado o grupo de teste do Sys-PANN, que totaliza 10 amostras. Como o Statística 10 apresenta, posteriormente ao treino, a predição de todos os valores do treinamento, então será separado desse grupo as mesmas 10 amostras para facilitar a comparação.

As figuras 6.1 e 6.2 apresentam os gráficos de dispersão do Sys-PANN e do Statistica 10, respectivamente. O gráfico de dispersão é utilizado para comparar pares de valores que no presente caso, são o valor ideal e o valor aproximado (ou atual).

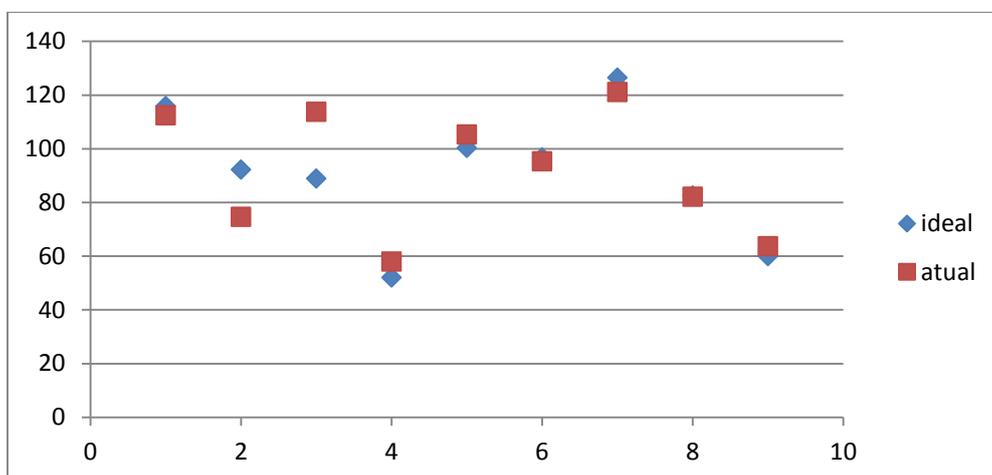


Figura 6.1- Gráfico de dispersão do Sys-PANN

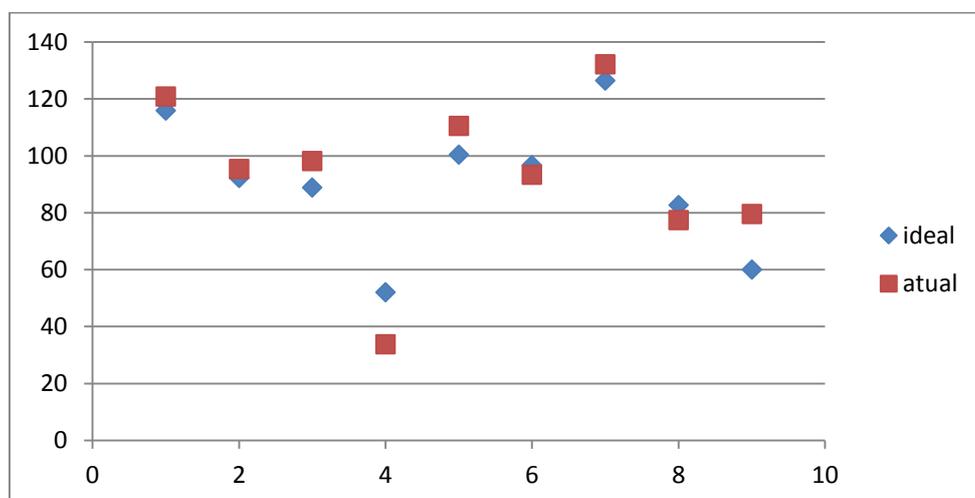


Figura 6.2- Gráfico de dispersão do Statistica

Nota-se que, no gráfico do Sys-PANN (figura 6.1), os pontos do valores atuais estão muito mais próximos aos pontos dos valores ideais, em comparação ao Statistica 10. E ainda, a maior parte desse pontos atuais sobrepõe os ideais, demonstrando o quão próximos estão.

A tabela 6.2 exibe os valores preditos por cada software em relação ao valor ideal do parâmetro Índice de Iodo. Observa-se que a predição pelo

Sys-PANN, na maioria das amostras, se aproxima mais do valor ideal do que a predição pelo Statística 10.

Tabela 6.2- Valores de predição de Índice de Iodo

Ideal	Sys-PANN	Statística
115,9	112,47	120,87
92,2	74,66	95,35
88,9	113,82	98,12
52	57,98	33,74
100,39	105,31	110,47
96,74	95,33	93,37
126,5	121,13	132,12
82,66	82,17	77,46
60	63,75	79,5

Pode-se notar assim o bom desempenho do Sys-PANN em relação ao Statística 10 com relação aos valores de teste. O Sys-PANN apresenta um desempenho ligeiramente melhor, considerando que este é um protótipo.

Essa ligeira superioridade pode ser atribuída ao fato de que o número de camadas ocultas do Statística 10 é insuficiente para o problema de aproximação de funções. A utilização de duas camadas é mais recomendado se a função alvo é de tipo desconhecido.

Diante do bom resultado comprovado do Sys-PANN, pode-se realizar a busca pela melhor configuração de RNA para cada parâmetro utilizando o algoritmo de treinamento Resilient Propagation com a função de ativação Logística-Sigmóide.

Para encontrar as melhores configurações, foram adotadas, para as três propriedades de biodiesel (Índice de Iodo, Massa específica e Viscosidade), os mesmos parâmetros de busca:

- 1º camada Oculta: 10 a 50 neurônios;
- 2º camada Oculta: 10 a 50 neurônios;
- Quantidade de treino em uma mesma configuração: 5;

- Máximo de épocas: 1000.

Diante da definição dos parâmetros, as três buscas foram realizadas e selecionaram-se as redes que possuíram os menores erros de validação para cada propriedade. A tabela 6.3 apresenta as melhores configurações encontradas.

Tabela 6.3 - Melhores configurações de RNA encontrada para os 3 parâmetros de qualidade

Nº de Neurônios	Índice de Iodo	Massa específica	Viscosidade
C. Entrada	13	13	13
1º C. Oculta	16	25	21
2º C. Oculta	48	34	12
C. Saída	1	1	1
Erro de Treino	0,0140	0,0312	0,0089
Erro de Validação	0,0328	0,0110	0,0277
Épocas	510	185	425

Observa-se que o número de neurônios variou bastante entre uma propriedade e outra, demonstrando o quanto é difícil encontrar manualmente uma configuração, o que é facilitado pela busca automática.

Vale notar também a interferência do número de épocas no treinamento. A massa específica, que teve poucas épocas, teve o erro de validação menor que o erro de treinamento, e o índice de iodo, que utilizou mais épocas, teve o erro de treinamento menor que o de validação. Isso demonstra e comprova que uma RNA inicia seu treino generalizando e, depois, no decorrer das épocas, se especializa.

A seguir, serão apresentados os resultados dos treinos para demonstrar o quanto as RNAs se aproximaram dos valores ideais, tanto no treinamento quanto no grupo de validação. Para isso serão apresentados os gráficos de dispersão. É importante ressaltar que, nestes gráficos, quanto menos pontos atuais (azul) aparecerem, melhor será o resultado, ou seja, mais os valores atuais estão sobrepostos pelos valores ideais (vermelho).

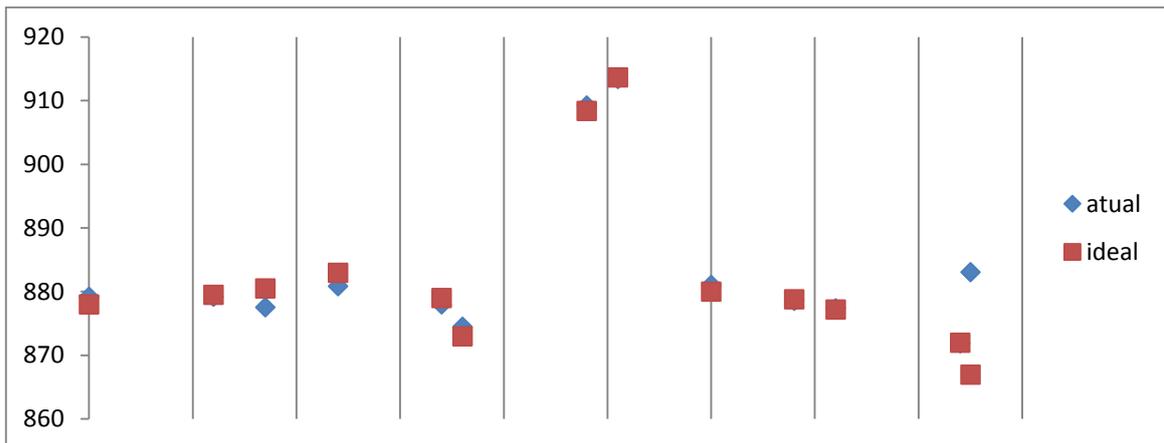
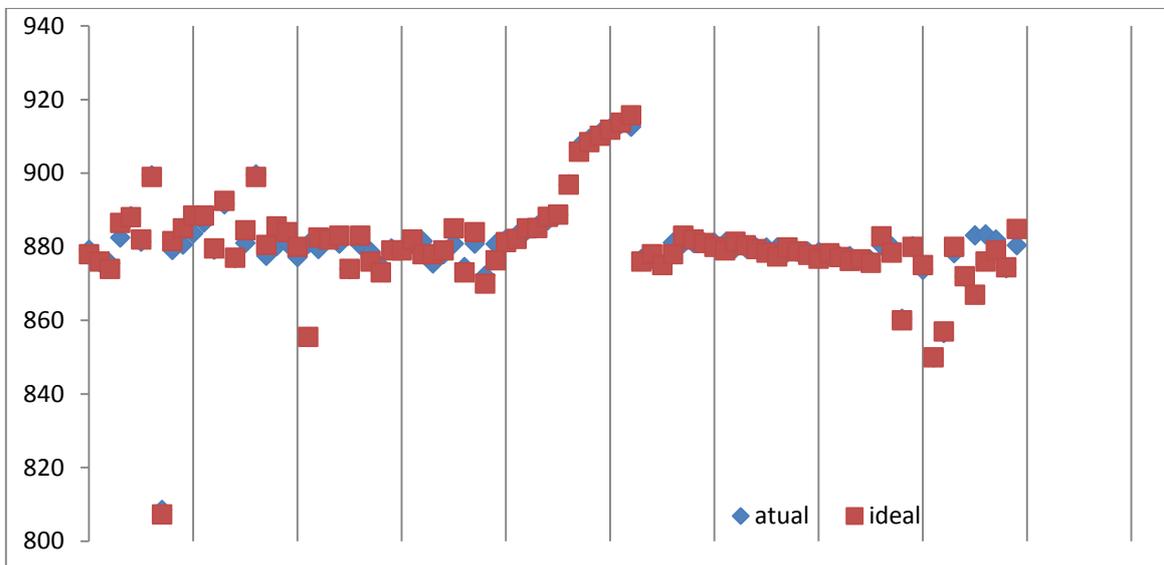


Figura 6.5- Gráfico de dispersão do grupo de validação da Massa Específica



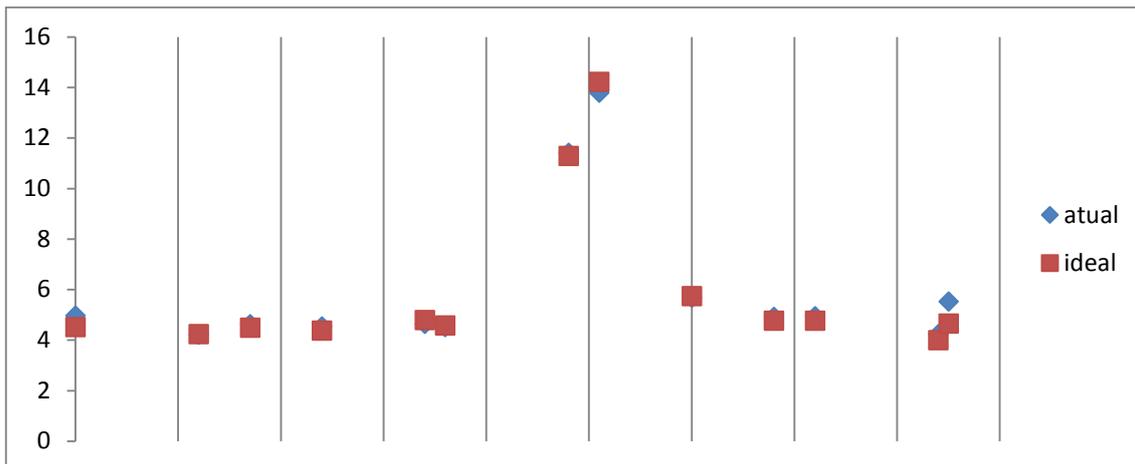


Figura 6.7 - Gráfico de dispersão do grupo de validação da Viscosidade

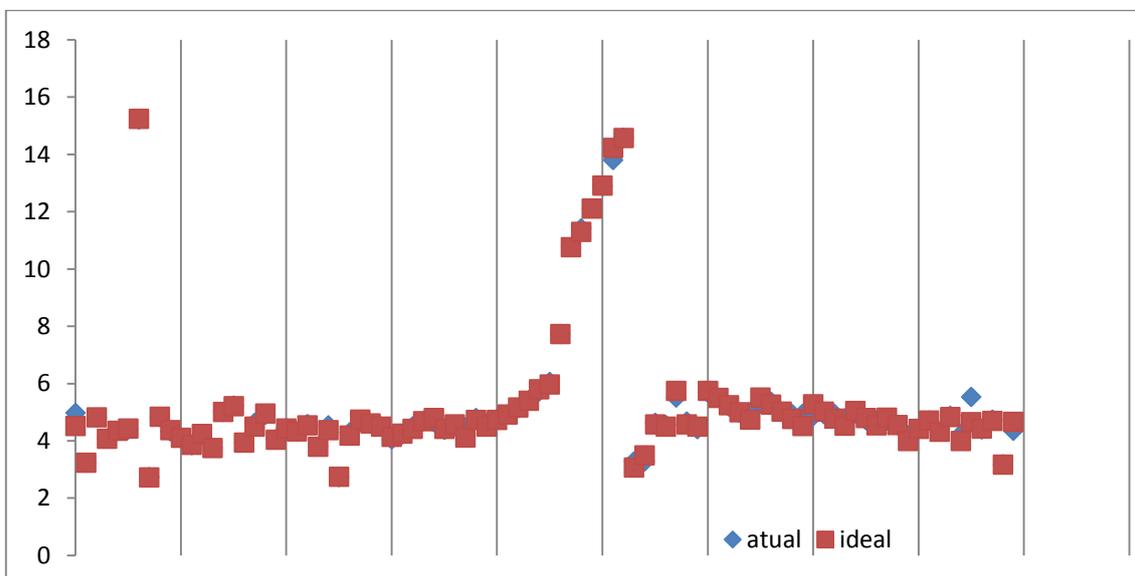


Figura 6.8 - Gráfico de dispersão do grupo de treinamento da Viscosidade

A melhor RNA treinada com o grupo da Viscosidade apresentou um erro de validação de 2,7% e de treinamento de 0,89%. Esses baixos valores condizem com o que aparece nos gráficos 6.7 e 6.8. Com um erro de treinamento muito baixo, menor de 1%, a RNA acerta quase todos os valores ideais, com apenas alguns erros se sobressaindo no gráfico 6.8.

Um fato em comum em todos os gráficos é que os pontos atuais (azul) aparecem mais nas áreas de menor concentração de pontos ideais (vermelho) dos gráficos, demonstrando que esses valores não estavam em quantidade suficiente para apresentar essas características para a rede

aprender. Com isso, os valores que se desviaram mais da média correspondem às características pouco treinadas, apontando assim uma deficiência no grupo de treinamento pela falta de homogeneidade de características.

Dessa forma, Sys-PANN utilizando o algoritmo de treinamento Resilient Propagation, função de ativação Logística-Sigmóide, validação cruzada como ponto de parada e a busca pela melhor configuração, se mostra como uma boa ferramenta para predição de parâmetros de qualidade de Biodiesel.

7. CONSIDERAÇÕES FINAIS

O biodiesel é um combustível produzido a partir do óleo de vegetais, que pode substituir, parcial ou totalmente, o diesel fóssil. Ele tem se tornado a cada ano mais importante a medida que seus benefícios vão sendo descobertos. Desta forma, a produção, o consumo e a pesquisa estão sendo estimulados pelo governo federal através do PNPB. No Maranhão, o LPQA é o laboratório responsável pelas pesquisas no estado e por isso decidiu-se ajudá-los com algo que os ajuda-se a acelerar suas pesquisas.

Para isso, foi realizado um estudo sobre o biodiesel, como ele é produzido, a sua composição química e seus critérios de qualidade. Descobriu-se então que a composição química do biodiesel (composição química dos ésteres) tem influência direta sobre alguns tipos de parâmetros de qualidade. Por isso, objetivou-se descobrir qual a relação matemática entre os ésteres e os parâmetros de qualidade, para que se pudesse desenvolver uma ferramenta que realizasse a predição desses parâmetros a partir dos percentuais de composição dos ésteres no Biodiesel.

Para realizar essa predição, decidiu-se utilizar um método de sucesso na área da IA que pode ser utilizada para este fim: as Redes Neurais Artificiais. Realizou-se um estudo detalhado das RNAs para descobrir qual a mais adequada já a RNA tem vários tipos, modelos e configurações.

A partir dos estudos, verificou-se que o problema de predição que busca uma relação se enquadra no grupo de problema de Aproximação de função. O tipo de RNA mais adequada para este problema é uma Rede de modelo Perceptron Multi-Camadas (MLP), conhecido como Aproximador Universal de Funções.

Após a escolha do tipo de rede, foi realizado um estudo sobre MLP para descobrir qual a melhor maneira de configurar a rede para o problema desejado. Para tal, concluiu-se que seria utilizada uma rede de duas camadas ocultas, com neurônios que teriam funções de ativação Sigmóide.

Porém, se descobriu que a definição da quantidade de neurônios de cada camada oculta é uma busca de tentativa e erro, por isso a ferramenta que

seria desenvolvida deveria realizar uma busca pela melhor quantidade de neurônios.

Após a definição da RNA, a ferramenta desejada para auxiliar na pesquisa do Biodiesel utilizando RNA foi desenvolvida, o Sys-PANN. Ela foi desenvolvida em Java. Para facilitar e padronizar os dados a serem utilizados, foi desenvolvido um aplicativo web em PHP para cadastro de biodiesel e sua composição química com os parâmetros de qualidade.

A RNA do Sys-PANN foi testado utiliza-se uma arquivo de dados reais de biodiesel de contendo 98 amostras com suas composições químicas e 3 parâmetros. Foi utilizado inicialmente 3 algoritmos de treinamento: BackPropagation com momentum, Resilient Propagation e Levenberg-Marquardt. Após alguns testes, verificou-se o desempenho melhor do Resilient Propagation.

Para um melhor resultado, o Sys-PANN faz apenas uma predição de parâmetro de amostra por vez pois a RNA foi desenvolvida teve melhores resultados tendo apenas um neurônio de saída.

Para verificar a qualidade dos resultados da RNA, foi utilizado o software Statistica, um renomado software do mercado que pode também gerar RNA. Nos testes de desempenho, os resultados do Sys-PANN foram um pouco superiores ao Statistica, demonstrando a qualidade dos resultados.

Posteriormente, foram geradas três redes, uma para cada parâmetro disponível, e através da fase de validação e teste, o Sys-PANN aproximou muito dos resultados esperados, chegando em alguns casos a igualar os resultados. Assim, se em algumas situações ele não chegar muito próximo do resultado esperado, pelo menos consegue tendenciar, pois o erro não é tão grande e é o suficiente para indicar se o valor está dentro do padrão de qualidade.

Vale ressaltar que a rede poderá ter um melhor desempenho se o grupo de treinamento por maior e, principalmente, mais homogêneo, o que produziria um resultado melhor ainda.

Assim, os resultados do Sys-PANN demonstram que a utilização de redes neurais artificiais para predição é um método eficaz. Quando a rede é bem treinada resulta numa boa predição ou, na pior das hipóteses, consegue tendenciar uma decisão.

Apesar dos bons resultados, o sistema pode ser melhorado num trabalho futuro, por isso pretende-se:

- I. Adicionar mais neurônios na camada de saída sem depreciar os resultados.
- II. Testar outros algoritmos de treinamento e propor alterações a eles para se ter um melhor resultado.
- III. Estudar e melhorar a condição de parada da RNA.
- IV. Utilizar o método de otimização de colônia de formigas para otimizar o fluxo de dados dentro de uma RNA

REFERÊNCIA

ANP -. **Resolução nº. 14**, de 11 de maio de 2012 (DOU 18.5.2012).

Arbib, M. A. **The Handbook Of Brain Theory and Neural Networks**. 2. ed. London, England: The MIT Press, 2002

Murugesan,A. Umarani,C. Chinnusamy,T.R. Krishnan,M. Subramanian,R. Neduzchezchain,N. **Production and analysis of bio-diesel from non-edible oils—A review**. Renewable and Sustainable Energy Reviews 13 (2009) 825–834

Almeida, A. D. **Comparação entre métodos para roteamento de redes de dados usando redes neurais artificiais**. Dissertação - UFSC.. Florianópolis, 2001

Basheer,I.A., Hajmeer, M.; **Artificial neural networks: fundamentals, computing, design, and application**. Journal of Microbiological Methods. ed 43 (2000) p 3 –31.

Berkeley Biodiesel - **History Of Biodiesel Fuel**. Acessado em 20 de janeiro de 2013, Disponível em: <http://www.berkeleybiodiesel.org/history-biodiesel-fuel-traced.html>

BIODIESELBR, “**Biodiesel no Brasil**”, Acessado em 15 de dezembro de 2012, Disponível em <http://www.biodieselbr.com>

Braga, A. P.; **Redes neurais artificiais: Teoria e aplicações**. 2ed - LTC, 2007.

Boers, E. J. W. e Kuiper,H. **Biological metaphors and the design of modular artificial neural networks**. Master's thesis, Leiden Univ., The Netherlands, Aug. 1992

Broomhead, S. and Lowe, D . **Multivariable function interpolation and adaptive networks** . Complex Systems, 2:321-355, 1988.

Cartilha_biodiesel -Biodiesel, **O Novo Combustível do Brasil. Programa Nacional de Produção e Uso Do Biodiesel** Acessado em 10 de fevereiro de 2013. Disponível em: http://www.mme.gov.br/programas/biodiesel/galerias/arquivos/biodiesel/cartilha_biodiesel_portugues.pdf

Coelho, B. F. M. **Modelo de previsão da evolução do biodiesel no Brasil utilizando lógica.** Dissertação - Rio de Janeiro: Faculdades Ibmecc. 2007.

Cybenko, G. **Continuous valued neural networks with two hidden layers are sufficient.** Technical report, Department of Computer Science, Tufts University, 1988.

Cybenko, G. **Approximation by superposition of a sigmoidal function.** Mathematics of Control, Signals and Systems, 2 , pp. 303–314. (1989).

Demirbas, A. **Relationships derived from physical properties of vegetable oil and biodiesel fuels.** Fuel 87 p 1743–1748. (2008)

DFBIO - **Processo de Extração de Óleos e Produção de Biodiesel em Miniusinas,** Acessado em 20 de dezembro de 2012, Disponível em: <http://www.dfbio.com.br/>

Fahlman, S. E. **An empirical study of learning speed in backpropagation networks,** School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-88-162, Sept. 1988.

Ferneda, E. **Redes neurais e sua aplicação em sistemas de recuperação de informação.** E. Ci. Inf., Brasília, v. 35, n. 1, p. 25-30, jan./abr. 2006

Geris, Regina; SANTOS, Nádia; AMARAL, Bruno; MAIA, Isabelle; CASTRO, Vinícius; CARVALHO, **Reação de Transesterificação para Aulas Práticas de Química Orgânica.** Química Nova, Salvador – BA, vol. 30, Nº05. 1369-1373, 2007.

Gianola D, Okut H., Weigel KA, Rosa GJM. **Predicting complex quantitative traits with Bayesian neural networks: a case study with Jersey cows and wheat.** BMC Genet. 2011

Grossberg, S. **Adaptive pattern classification and universal recoding, I: parallel development and coding of neural feature detectors.** Biological Cybernetics, New York, v.23, n.3, p. 121-134, 1976.

Guarieiro, L. L. N.; Vasconcellos, P. C.; Solci, M. C. **Poluentes Atmosféricos Provenientes da Queima de Combustíveis Fósseis e Biocombustíveis: Uma Breve Revisão.** Rev. Virtual Quim., 2011, 3 (5), 434-445. Data de publicação na Web: 16 de novembro de 2011

Haykin, S. S. - **Redes Neurais - Principios e Pratica.** Bookman, 2001

Heaton, J.; **Programming Neural Networks with Encog 2 in Java.** Heaton Research, Inc. March, 2010.

Hebb, D.O. **The Organization of Behavior.** Wiley, 1949.

Hecht-Nielsen,R. **Neurocomputing.** HNC, Inc. and University of California, San Diego. 1990.

Hoekman, S.K. , Broch, A., Robbins, C., Cenicerros, E., Natarajan, M. **Review of biodiesel composition, properties, and specifications.** Renewable and Sustainable Energy Reviews, 16 (2012) 143-169.

Hopfield, J. J., **Neural Networks and Physical Systems with Emergent Collective Computational Abilities,** Proc. Nat. Acad. Science, USA, vol. 79, pgs. 2554-2558. (1982).

Hornik, K., Stinchcombe,M.B., White, H. **Multilayer feedforward networks are universal approximators.** Neural Networks, Vol. 2, pp. 359-366, 1989.

Kalogirou, S.A., Mathioulakis, E., Belessiotis, V. **Artificial neural networks for the performance prediction of large solar systems.** Renewable Energy, Volume 63, March 2014.

Kratzer, K. P., **Neuronale Netze – Grundlagen und Anwendungen**, Carl Hanser Verlag, 1991.

Knothe, G. **Dependence of biodiesel fuel properties on the structure of fatty acid alkyl esters**. Fuel Proc Technol 2005; 86:1059–70.

Kinnebrock, W., **Neuronale Netze – Grundlagen, Anwendungen, Beispiele**, R. Oldenbourg Verlag, GmbH, 1992.

Kohonen, T., **Clustering Taxonomy and Topological Maps of Patterns**. Proc. of the Sixth Intern. Conference on Pattern Recognition, pgs. 114-128.(1982).

Lippmann, R.P. **An introduction to computing with neural nets**. IEEE ASSP Magazine, [S.I.], v. 4, n. 2, p. 4-22. 1987.

Lobo, P.I.; Ferreira, S.L.C.; Cruz, R.S. - **Biodiesel: parâmetros de qualidade e métodos analíticos**. Quim. Nova, Vol. 32, No. 6, 1596-1608, 2009

Secretaria de Tecnologia Industrial. **Produção de combustíveis líquidos a partir de óleos vegetais**. Brasília: STI/CIT, 1985.

Mcculloch, W. S.; Pitts, W. H. **A logical calculus of the ideas immanent in nervous activity**. **Bulletin of Mathematical Biophysics**, v. 5, p. 115-133, 1943.

Melo, M.L.S. **Caracterização, estabilidade oxidativa e determinação do potencial enérgico do biodiesel derivado do mamão**. Tese de Doutorado, UFPB, 2010

Minsky and S. Papert. **Perceptrons: an introduction to computational geometry**. MIT Press, Massachusetts. 1969.

Najjar, Y.M.; Basheer, I.A., Hajmeer, M.; **Computational neural networks for predictive microbiology: I. methodology**. International Journal of Food Microbiology 34 p.27-49. 1997.

Novakovic, B.M.; **General models of artificial neural networks**. Industrial Electronics, Control and Instrumentation, Kobe. vol.2, 1991

Okut H, Gianola D, Rosa GJM, Weigel KA: **Prediction of body mass index in mice using dense molecular markers and a regularized neural network.** Genet Res Camb 2011.

Parente, E.J.S., **Biodiesel: Uma Aventura Tecnológica num País Engraçado**, "Tecbio", Fortaleza, 2003

Patnaik, A.; Misra, R.K. **ANN Techniques in Microwave Engineering.** Microwave Magazine, IEEE, [S.l.], v. 1, n. 1, p. 55-60, mar. 2000.

PETROBRAS, **Produção de Biocombustível**, Acessado em 20 de dezembro de 2012, Disponível em <http://www.petrobras.com.br/pt/quem-somos/perfil/atividades/producao-biocombustiveis/>

PNPB - PROGRAMA NACIONAL DE PRODUÇÃO E USO DE BIODIESEL, **Cartilha Biodiesel SEBRAE**, Acessado em 02 de agosto de 2011, Disponível em http://www.biodiesel.gov.br/docs/Cartilha_Sebrae.pdf

Powell, M.J.D. **Radial basis function approximation to polynomials.** Numerical Analysis 1987 Proceedings, p. 223–241, 1988.

Rathmann, R.; Benedetti, O.; Plá, J.A.; Padula, A.D. (2007). **Biodiesel: uma alternativa estratégica na matriz energética brasileira?**. Disponível em: <http://www.biodiesel.gov.br/docs/ArtigobiodieselGINCOB-UFRGS.pdf>.

Ramadhasa, A.S., Jayaraja, S., Muraleedharana, C., Padmakumarib, K. - **Artificial neural networks used for the prediction of the cetane number of biodiesel.** Renewable Energy 31 (2006) 2524–2533.

Riedmiller, M.; Braun, H. **A direct adaptive method for faster backpropagation learning: The rprop algorithm.** Proceedings of the IEEE Int. Conf. on Neural Networks, San Francisco, USA, p. 586-591, 1993

Rosenblatt, F. The perceptron: **A probabilistic model for information storage and organization in the brain.** Psychol. Rev.. 65:386- 408, 1958.

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. **Learning representations by back-propagating errors".** Nature 323 (6088): 533–536. 1986

Santos dos, R. B., Lacerda Jr, V., Castro, E. V. R., Souza , T. S. **Síntese e estudo do biodiesel do óleo da cutieira (Joannesia princeps)**.30a Reunião Anual da Sociedade Brasileira de Química. 2007

Tolmasquim, Mauricio Tiomno (Coord. Geral); SZKLO, Alexandre Salem (Coord.Executivo). **A Matriz Energética Brasileira na Virada do Milênio**. Rio de Janeiro:COPPE/UFRJ; ENERGE, 2000.

Yabanova, I.; Keçebaş, A., **Development of ANN model for geothermal district heating system and a novel PID-based control strategy**. Applied Thermal Engineering, 51 (2013) p 908-916. doi:10.1016

Widrow, B. and Hoff, M.E. **Adaptative switching circuits**. Institute of Radio Engineers, Western Electronic Show and Convention, part 4, pp.96-140, 1960

ANEXO I - Resolução nº. 14 da ANP - Especificação do Biodiesel

CARACTERÍSTICA	UNIDADE	LIMITE	MÉTODO		
			ABNT NBR	ASTM D	EN/ISO
Aspecto	-	LII (1)	-	-	-
Massa específica a 20º C	kg/m ³	850 a 900	7148 14065	1298 4052	EN ISO 3675 - EN ISO 12185
Viscosidade Cinemática a 40ºC	mm ² /s	3,0 a 6,0	10441	445	EN ISO 3104
Teor de água, máx.	mg/kg	(2)	-	6304	EN ISO 12937
Contaminação Total, máx.	mg/kg	24	-	-	EN ISO 12662 NBR 15995
Ponto de fulgor, mín. (3)	ºC	100,0	14598	93	EN ISO 3679
Teor de éster, mín	% massa	96,5	15764	-	EN 14103
Resíduo de carbono, máx. (4)	% massa	0,050	15586	4530	-
Cinzas sulfatadas, máx.	% massa	0,020	6294	874	EN ISO 3987
Enxofre total, máx.	mg/kg	10	15867	5453	EN ISO 20846 EN ISO 20884
Sódio + Potássio, máx.	mg/kg	5	15554 15555 15553 15556	-	EN 14108 EN 14109 EN 14538
Cálcio + Magnésio, máx.	mg/kg	5	15553 15556	-	EN 14538
Fósforo, máx.	mg/kg	10	15553	4951	EN 14107

Corrosividade ao cobre, 3h a 50 °C, máx.	-	1	14359	130	EN ISO 2160
Número Cetano (5)	-	Anotar	-	613 6890 (6)	EN ISO 5165
Ponto de entupimento de filtro a frio, máx.	°C	(7)	14747	6371	EN 116
Índice de acidez, máx.	mg KOH/g	0,50	14448 -	664 -	EN 14104 (8)
Glicerol livre, máx.	% massa	0,02	15341 (8) 15771	6584 (8) -	EN 14105 (8) EN 14106 (8)
Glicerol total, máx. (9)	% massa	0,25	15344 15908	6584 (8) -	EN 14105 (8)
Monoacilglicerol, máx.	% massa	0,80	15342 (8) 15344 15908	6584 (8)	EN 14105 (8)
Diacilglicerol, max.	% massa	0,20	15342 (8) 15344 15908	6584 (8)	EN 14105 (8)
Triacilglicerol, máx.	% massa	0,20	15342 (8) 15344 15908	6584 (8)	EN 14105 (8)
Metanol e/ou Etanol, máx.	% massa	0,20	15343	-	EN 14110 (8)
Índice de Iodo	g/100g	Anotar	-	-	EN 14111 (8)
Estabilidade à oxidação a 110°C, mín. (10)	h	6	-	-	EN 14112 EN 15751 (8)

Nota:

- (1) Límpido e isento de impurezas, com anotação da temperatura de ensaio.
- (2) Será admitido o limite de 380 mg/kg 60 dias após a publicação da Resolução. A partir de 1º de janeiro de 2013 até 31 de dezembro de 2013 será admitido o limite máximo de 350 mg/kg e a partir de 1º de janeiro de 2014, o limite máximo será de 200 mg/kg.
- (3) Quando a análise de ponto de fulgor resultar em valor superior a 130º C, fica dispensada a análise de teor de metanol ou etanol.
- (4) O resíduo deve ser avaliado em 100% da amostra.
- (5) Estas características devem ser analisadas em conjunto com as demais constantes da tabela de especificação a cada trimestre civil. Os resultados devem ser enviados à ANP pelo Produtor de biodiesel, tomando uma amostra do biodiesel comercializado no trimestre e, em caso de neste período haver mudança de tipo de material graxo, o Produtor deverá analisar número de amostras correspondente ao número de tipos de materiais graxos utilizados.
- (6) O método ASTM D6890 poderá ser utilizado como método alternativo para determinação do número de cetano.
- (7) Limites conforme Tabela II. Para os estados não contemplados na tabela o ponto de entupimento a frio permanecerá 19ºC.
- (8) Os métodos referenciados demandam validação para os materiais graxos não previstos no método e rota de produção etílica.
- (9) Poderá ser determinado pelos métodos ABNT NBR 15908, ABNT NBR 15344, ASTM D6584 ou EN14105, sendo aplicável o limite de 0,25% em massa. Para biodiesel oriundo de material graxo predominantemente láurico, deve ser utilizado método ABNT NBR 15908 ou ABNT NBR 15344, sendo aplicável o limite de 0,30% em massa.
- (10) O limite estabelecido deverá ser atendido em toda a cadeia de abastecimento do combustível.