

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL**

**MATEUS DE MATOS
SYMON OLENIK**

**IMPLEMENTAÇÃO DE REDES NEURAIS PERCEPTRON E ADALINE
EM AMBIENTE LABVIEW**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2018

MATEUS DE MATOS

SYMON OLENIK

**IMPLEMENTAÇÃO DE REDES NEURAIS PERCEPTRON E ADALINE
EM AMBIENTE LABVIEW**

Trabalho de Conclusão de Curso apresentado como requisito para a obtenção do título de Tecnólogo em Automação Industrial, do Departamento de Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. José Jair Alves Mendes Júnior

PONTA GROSSA

2018



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa

Departamento Acadêmico de Eletrônica
Tecnologia em Automação Industrial



TERMO DE APROVAÇÃO

IMPLEMENTAÇÃO DE REDES NEURAIS PERCEPTRON E ADALINE EM AMBIENTE LABVIEW

por

SYMON OLENIK
MATEUS DE MATOS

Este Trabalho de Conclusão de Curso foi apresentado em 02 de Julho de 2018 como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial. Os candidatos foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Me. José Jair Alves Mendes Jr.
Prof. Orientador

Dra. Marcella Scoczynski Ribeiro Martins
Membro titular

Dr. Sérgio Luiz Stevan Junior
Membro titular

Dedico este trabalho à minha família,
pelos momentos de ausência.

AGRADECIMENTOS

Agradecemos primeiramente a Deus, pela força na caminhada e pela luz que nos guiou. Agradecemos às nossas famílias pelos momentos de apoio e por entender as ausências. Agradecemos aos amigos, pelo companheirismo na caminhada acadêmica. Agradecemos aos professores pelo conhecimento em especial ao orientador deste trabalho. Agradecemos um ao outro por sempre acreditar e incentivar um ao outro.

A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho original.
(EINSTEIN, Albert)

RESUMO

MATOS, Mateus, OLENIK, Symon. **Implementação de redes neurais Perceptron e Adaline em ambiente LabVIEW™**. 2018. 59f. Trabalho de Conclusão de Curso Tecnologia em Automação Industrial - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

As Redes Neurais Artificiais (RNAs) são sistemas especialistas que possuem estruturas inspiradas em neurônios biológicos, que devido a sua capacidade de processar dados complexos e possuir aprendizado de forma constante, são amplamente utilizadas para resolução de problemas. Essa capacidade de resolver problemas possui diversas áreas de atuação como finanças, economia, controle de plantas fabris e reconhecimento de padrões. Com a evolução tecnológica a programação de uma RNA pode ser feita em diversos *softwares* e plataformas microcontroladas diferentes. Uma técnica muito utilizada recentemente para prototipagem, testes e para estudos é a instrumentação virtual, definida na utilização de *softwares* que simulam a montagem de um circuito ou de uma cadeia de operações lógicas sem a necessidade da montagem física do mesmo, economizando tempo e recursos; e, tendo ainda a vantagem de poder ser editada, copiada e simulada sem nenhum tipo de recurso físico, a não ser um computador e o programa desejado de instrumentação virtual. O estudo deste trabalho busca implementar e comparar RNAs do tipo Perceptron e Adaline nos *softwares* MATLAB® e LabVIEW™, como forma de observar o processo de construção e as formas de apresentação de ambos os programas. Para que tais resultados fossem alcançados, foram utilizadas bases de dados com saídas e entradas conhecidas. Foram testadas em ambos os *softwares* todas as amostras propostas e adquiridos dados para estudo. Com a obtenção dos resultados dos testes, foram comparados e, pode-se observar que, o ambiente MATLAB® foi mais eficaz em função de tempo de processamento, porém, o *software* LabVIEW™ apresenta os resultados esperados e ainda fornece uma plataforma de aquisição de dados externos mais prática e versátil.

Palavras-chave: RNAs. MATLAB®. LabVIEW™. Instrumentação Virtual.

ABSTRACT

Matos, Mateus, OLENIK, Symon. **Implementation of Perceptron and Adaline neural networks in LabVIEW™ environment.** 2018. 59p. Work of Conclusion Course Technology in Industrial Automation - Federal Technology University - Paraná. Ponta Grossa, 2018.

Artificial Neural Networks (ANN) are specialist systems which have structures inspired in biological neurons, which due their capacity to process complex data and constant learning are widely used for solving problems. This capacity for problem solving has several areas of operation such as finances, economics, factory plant control, and pattern recognition. With the technological evolution the programming of an ANN can be made in different microcontrolled plataforms. A very used technique for prototyping, test, and for studies is the virtual instrumentation. Virtual Instrumentation is defined in the use of *softwares* that simulates the assembly of a circuit or a chain of logical operations without the need of the physical assembly. This technique allows to saving time and resources, whilst still have the advantage of being edited, copied, simulated without any kind of physical resource other than the computer and the program for the virtual instrumentation. The study of this work intent to implement and compare Perceptron and Adaline ANNs in the *softwares* MATLAB® and LabVIEW™, as a way to observing the construction process and the form of presentation in both programs. For such results to be achieved, databases witch known outputs and inputs ware used. Both samples were tested in both programs. With the results of the tests, they were compared and it can be observed that the MATLAB® environment will be more effective in processing time, however, the LabVIEW™ program presents the expected results and still provides an external data acquisition platform more practical and versatile.

Keywords: ANN. MATLAB®. LabVIEW™. Virtual Instrumentation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Partes de um neurônio	12
Figura 2 – Modelo matemático de um neurônio artificial	13
Figura 3– Demonstração gráfica de uma rede do tipo Perceptron.	18
Figura 4- Demonstração gráfica de uma rede do tipo Adaline.....	20
Figura 5 – Etapas realização do trabalho.	22
Figura 6 - Etapa de apresentação de dados para treinamento da rede Perceptron de um neurônio onde a) Apresentação de dados, b) reconhecimento do tamanho da matriz, c) geração de pesos sinápticos e d) inserção do limiar de ativação.....	24
Figura 7- Etapa de treinamento dentro do loop <i>while</i> da rede Perceptron de uma camada onde a) Loop for de multiplicação das entradas pelos pesos, b) saída do resultado <i>u</i> e geração da função de ativação e c) comparação da saída desejada com a função.	25
Figura 8- Etapa de correção de pesos sinápticos da rede Perceptron de um neurônio onde a) Loop <i>while</i> para correção dos pesos e b) registro da atualização dos pesos.	26
Figura 9 - Apresentação dos resultados para a rede Perceptron de uma neurônio onde a) Saídas finais do treinamento, b) pesos finais do treinamento e c) número de épocas totais.....	26
Figura 10 - Apresentação dos dados para o treinamento da rede Adaline onde a) Apresentação dos dados, b) reconhecimento do tamanho da amostra, c) inserção do limiar de ativação e d) geração de pesos sinápticos aleatórios.	27
Figura 11 – Diagrama de Blocos do cálculo do erro quadrático médio onde a) apresentação dos dados da sub rotina e b) loop for para cálculo do EQM.....	28
Figura 12 - Processo de correção dos pesos sinápticos da rede Adaline onde a) sub rotina do EQM, b) loop for para cálculo de saídas e c) comparação de EQM anterior e atual para critério de parada.	29
Figura 13 – Diagrama de blocos da fase de operação da rede neural.	30
Figura 14 – Segmento da fase de operação, sendo a) apresentação dos dados de entrada e em b) inserção dos pesos obtidos no treinamento.	30
Figura 15 – Segmento da fase de operação, sendo a) loop for para multiplicação dos dados e em b) geração da função de saída e registro das saídas obtidas.	31
Gráfico 1 – Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Perceptron LabVIEW™.....	41
Gráfico 2 – Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Adaline LabVIEW™.	42
Gráfico 3 - Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Perceptron MATLAB®.....	42
Gráfico 4- Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Adaline MATLAB®.....	43

Quadro 1 – Sequência de passos para execução de treinamento de uma Perceptron.	15
Quadro 2– Pseudocódigo do treinamento de uma rede Perceptron de um neurônio.	17
Quadro 3 – Pseudocódigo do treinamento de uma rede Adaline.	19

LISTA DE TABELAS

Tabela 1– Base binária AND.....	31
Tabela 2 – Base binária NAND.....	32
Tabela 3 – Base binária OR.....	32
Tabela 4 – Base binária NOR.....	32
Tabela 5 - Conjunto de treinamento referente as amostras de óleo.....	33
Tabela 6 – Conjunto de treinamento referente ao acionamento de duas válvulas....	34
Tabela 7 – Resultados porta AND 3 entradas.....	35
Tabela 8 – Resultados porta NAND 3 entradas.....	35
Tabela 9 – Resultados porta OR 3 entradas.....	35
Tabela 10 – Resultados porta NOR 3 entradas.....	35
Tabela 11 – Resultados do conjunto de treinamento referente as amostras de óleo.	36
Tabela 12 – Resultados do conjunto de treinamento referente ao acionamento de duas válvulas.....	36
Tabela 13 – Resultados porta AND 3 entradas.....	36
Tabela 14 – Resultados porta NAND 3 entradas.....	36
Tabela 15 – Resultados porta OR 3 entradas.....	37
Tabela 16 – Resultados porta NOR 3 entradas.....	37
Tabela 17 – Resultados do conjunto de treinamento referente as amostras de óleo.	37
Tabela 18 – Resultados do conjunto de treinamento referente ao acionamento de duas válvulas.....	37
Tabela 19 – Resultados base AND 3 entradas.....	38
Tabela 20 – Resultados base NAND 3 entradas.....	38
Tabela 21 – Resultados base OR 3 entradas.....	38
Tabela 22 – Resultados base NOR 3 entradas.....	38
Tabela 23 – Resultados do conjunto de treinamento referente as amostras de óleo.	39
Tabela 24 – Resultados do conjunto de treinamento referente as aberturas de válvulas.....	39
Tabela 25 – Resultados porta AND 3 entradas.....	39
Tabela 26 – Resultados porta NAND 3 entradas.....	39
Tabela 27 – Resultados porta OR 3 entradas.....	39
Tabela 28 – Resultados porta NOR 3 entradas.....	40
Tabela 29 – Resultados do conjunto de treinamento referente as amostras de óleo	40
Tabela 30 – Resultados do conjunto de treinamento referente as aberturas de válvulas.....	40

SUMÁRIO

1 INTRODUÇÃO	8
1.1 TEMA DA PESQUISA	8
1.2 DELIMITAÇÃO DO TEMA	9
1.3 PROBLEMA	9
1.4 OBJETIVOS	9
1.4.1 Objetivo geral	9
1.4.2 Objetivos específicos	9
1.5 JUSTIFICATIVA	10
1.6 METODOLOGIA	10
2 REVISÃO BIBLIOGRÁFICA	11
2.1 REDES NEURAIS ARTIFICIAIS	11
2.1.1 Definição de Neurônio Biológico	12
2.1.2 Neurônio Artificial e Rede Perceptron	13
2.1.3 Treinamento da Rede Neural Perceptron	14
2.2 ADALINE	17
2.3 SOFTWARE LABVIEW™	20
3 METODOLOGIA	22
4 DESENVOLVIMENTO	24
4.1 TREINAMENTO DA REDE PERCEPTRON NO LABVIEW™	24
4.2 TREINAMENTO DA REDE ADALINE NO LABVIEW™	26
4.3 OPERAÇÃO DA REDE PERCEPTRON E ADALINE NO LABVIEW™	29
4.4 BASES PARA TESTE DE FUNCIONAMENTO DAS REDES	31
4.4.1 Bases Binárias	31
4.4.2 Bases de Teste Não-binárias	33
5 RESULTADOS	35
5.1 RESULTADOS USANDO O LABVIEW™ EM REDE PERCEPTRON	35
5.1.1 Rede Adaline	36
5.2 TABELAS DOS RESULTADOS MATLAB®	37
5.2.1 Tabelas Perceptron	38
5.2.2 Tabelas Adaline	39
5.3 GRÁFICOS DA EVOLUÇÃO DOS PESOS	41
6 CONSIDERAÇÕES FINAIS	44
REFERÊNCIAS	45
APÊNDICE A - Diagrama da rede Perceptron em LabVIEW™	47
APÊNDICE B - Diagrama da rede Adaline em LabVIEW™	49
APÊNDICE C - Código da rede Perceptron em MATLAB®	51
APÊNDICE D - Código da rede Adaline em MATLAB®	53

1 INTRODUÇÃO

As Redes Neurais Artificiais (RNAs) são sistemas especialistas que possuem estruturas inspiradas em neurônios biológicos. Devido a sua capacidade de processar dados complexos e possuir aprendizado de forma constante, as RNAs são amplamente utilizadas para resolução de problemas. Essa capacidade de resolver problemas possui inúmeras áreas de atuações como economia, finanças, controle de plantas fabris, reconhecimento de padrões, entre outros (SILVA; SPATTI; FLAUZINO, 2010)

Essa grande flexibilidade de trabalhar em diversas áreas e sua fácil programação é um ponto que diferencia as RNAs de métodos convencionais, fazendo com que sejam usadas em diversas áreas de atuação. Com a evolução tecnológica a programação de uma RNA pode ser feita em diversos *softwares* e plataformas microcontroladas diferentes. Isso é das vantagens que as tornam cada vez mais populares para resolução de problemas, tendendo a custo computacionais menores que outros métodos de programação.

Para a programação de redes neurais, são usualmente utilizados os *softwares* como MATLAB®, R® e STATISTICA™. Pela sua utilização na área de instrumentação e de automação industrial, para este trabalho foi escolhido a plataforma LabVIEW™.

Uma técnica muito utilizada recentemente para prototipagem, testes e para estudos é a instrumentação virtual, que simula a montagem de um circuito ou de uma cadeia de operações lógicas sem a necessidade da montagem física do mesmo, desta forma economizando tempo, e recursos, e, tendo ainda a vantagem de poder ser editada, copiada, e simulada sem nenhum tipo de recurso físico, a não ser um computador e o programa desejado de instrumentação virtual. Por sua vez, a instrumentação virtual pode ser auxiliada no desenvolvimento de sistemas de aprendizado, como as RNAs e em instrumentos e sistemas inteligentes.

1.1 TEMA DA PESQUISA

Desenvolvimento de Redes Neurais Artificiais em LabVIEW™.

1.2 DELIMITAÇÃO DO TEMA

Programação de Redes Neurais Artificiais Perceptron e Adaline implementadas em ambiente de instrumentação virtual (*software* LabVIEW™).

1.3 PROBLEMA

Como a instrumentação virtual pode ser melhorada com o uso de Redes Neurais Artificiais do tipo Perceptron e Adaline.

1.4 OBJETIVOS

1.4.1 Objetivo geral

Implementar um sistema de Redes Neurais Artificiais na plataforma de instrumentação virtual LabVIEW™ utilizando as topologias Perceptron e Adaline para que possam ser utilizadas em diversas aplicações sendo apresentadas de forma didática.

1.4.2 Objetivos específicos

- Realizar uma revisão bibliográfica contendo os princípios de RNAs, especialmente da Perceptron e da Adaline.
- Implementar o algoritmo das RNAs utilizando o *software* de processamento MATLAB®;
- Desenvolver as RNAs com as rotinas em LabVIEW™ em forma de subVI;
- Comparar o desempenho das RNAs com os mesmos conjuntos de amostras para as duas rotinas;
- Analisar os resultados do desempenho das RNA;
- Esclarecer os pontos observados no decorrer do processo de utilização do LabVIEW™ para esta aplicação.

1.5 JUSTIFICATIVA

Com a ampla utilização da instrumentação virtual, observa-se que o tratamento e processamento de dados é facilitado por meio destas plataformas. A exemplo disso, um dos processamentos de dados é a Rede Neural Artificial, que possui como base, a tentativa de representação do funcionamento do cérebro humano (SILVA; SPATTI; FLAUZINO, 2010).

A solução de problemas lineares e não lineares com a utilização das RNAs (CUNHA et al., 2007) vem ganhando espaço com este tipo de programação, como nas áreas médicas, agrícolas, industriais, entre outras (CAVALCANTE et al., 2014;SALLE CTP et al., 2012).

Propõe-se então, neste trabalho, o desenvolvimento de RNAs para a solução de problemas, analisar as suas vantagens no ramo da automação industrial, mais precisamente aplicada a instrumentação virtual, caracterizada pelo LabVIEW™. O LabVIEW™ foi a ferramenta escolhida devido as suas características de poder englobar um sistema de supervisão de dados claro e intuitivo, oferecendo à disposição do programador suas capacidades de processamento, aquisição e comunicação com outros dispositivos de forma facilitada.

1.6 METODOLOGIA

O método de pesquisa a ser aplicado para chegar aos objetivos estabelecidos é de caráter experimental, pois há a necessidades de coletas de dados a partir de um experimento para comprovar a sua eficácia. As pesquisas experimentais seguem por um planejamento rigoroso, onde as etapas de pesquisas iniciam pela formulação de problemas e hipóteses. Ela pode ser desenvolvida em laboratórios ou campos aonde são criadas condições de manipulações. Sobre o tratamento dos dados, estes se enquadram com o quesito quantitativo, pela necessidade de operações de caráter numérico (GERHARDT; SILVEIRA, 2009).

Para a comparação de resultados, foram coletados os dados referentes ao tempo de processamento, número de épocas para convergência e comportamento das correções dos pesos.

2 REVISÃO BIBLIOGRÁFICA

A seguinte seção apresenta os principais conceitos sobre Redes Neurais Artificiais, sobre o *software* utilizado para a sua construção (LabVIEW™).

2.1 REDES NEURAS ARTIFICIAIS

As RNAs são definidas como modelos computacionais, as quais foram inspiradas no sistema nervoso dos seres vivos. Elas possuem como fator principal a capacidade adquirir e manter conhecimentos, sendo elucidadas como um conjunto de unidades de processamento (SILVA; SPATTI; FLAUZINO, 2010). São caracterizadas por possuírem Neurônios Artificiais, interligados por um grande número de interconexões (chamadas de sinapses artificiais), representadas numericamente por vetores ou matrizes. As RNAs possuem um modelo matemático inspirado na estrutura neural de organismos inteligentes, os quais permitem a aquisição de conhecimento por meio de experiência, conseguindo se assemelhar a um cérebro humano (HAYKIN, 2001).

Pela sua construção, a semelhança com o cérebro humano dá-se pelas características de poder adquirir conhecimento por meio de processos de aprendizagem e pelas forças de conexão que existem entre os neurônios artificiais (chamadas de pesos sinápticos) (HAYKIN, 2001). É nos pesos sinápticos que o conhecimento é armazenado. Normalmente, as RNAs são implementadas com componentes eletrônicos ou simuladas em computadores digitais. Dentre as principais características das RNA estão a capacidade de aprendizado, adaptação por experiência, habilidade de generalização e facilidade de prototipagem (SILVA; SPATTI; FLAUZINO, 2010).

O primeiro trabalho sobre o sistema de Redes Neurais Artificiais (RNA) foi realizado no ano de 1943, por McCulloch e Pitts, que desenvolveram uma pesquisa sobre o comportamento do neurônio biológico. Esta pesquisa teve como principal objetivo criar um modelo matemático do neurônio biológico. McCulloch e Pitts concluíram, por meio desse aprendizado, a importância fundamental para a futura implementação computacional do neurônio, tendo como fatores predominantes a atividade do neurônio obedecendo o princípio de tudo ou nada. McCulloch e Pitts

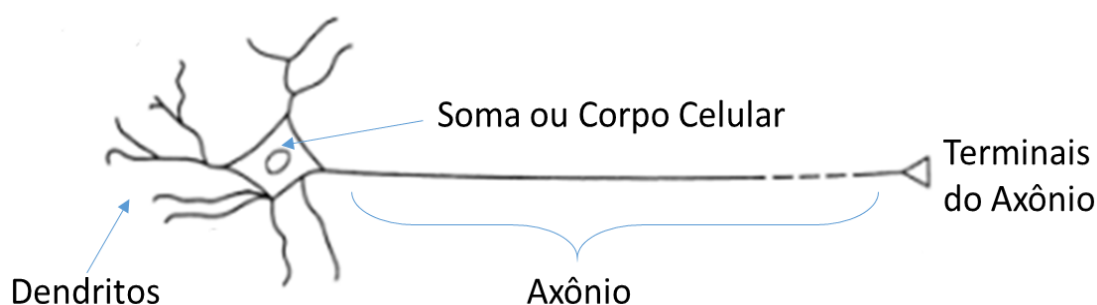
definiram que o neurônio possui e seu desempenho se assemelha ao comportamento de “tudo ou nada”, portanto é necessário estabelecer suas funções de forma linear na qual seja definida. (CARDON; MÜLLER, 1994).

Contudo, apesar de todos os estudos realizados até a década de 1950, o primeiro modelo de RNA foi o Perceptron de um neurônio desenvolvido por Frank Rosenblatt em 1958. O Perceptron é uma RNA simples, pois constitui-se de uma camada de entrada e uma outra camada de saída (SILVA; SPATTI; FLAUZINO, 2010). A cada entrada existe um peso relacionado, os quais o valor de saída será a soma dos produtos de cada entrada pelo seu respectivo peso.

2.1.1 Definição de Neurônio Biológico

Os neurônios são células nervosas, as quais conduzem impulsos nervosos. Como unidades básicas do sistema nervoso, carregam consigo informações e estímulo para todo o corpo. Estes são formados, como apresentado na Figura 1, por dendritos (que recebem as informações e estímulos); corpo celular (que integra as informações recebidas); e axônios (que enviam o impulso nervoso). Este impulso do axônio poderá ser tanto para outro neurônio, fibra muscular ou glândula (GOMIDE, 2012) (MONTANAR, 2016).

Figura 1 – Partes de um neurônio



Fonte: Levitan, Kaczmarek (2002), adaptado.

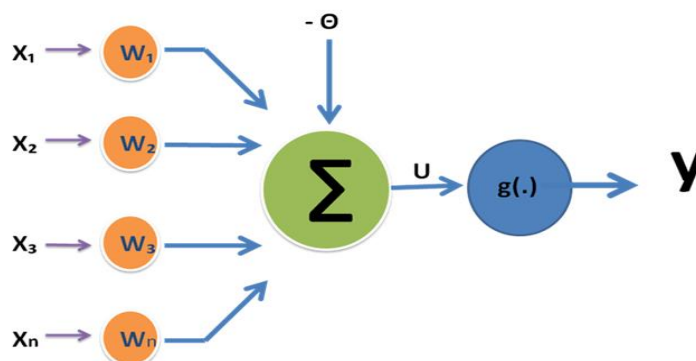
A transmissão que ocorre entre os neurônios depende de reações físico-químicas que ocorrem nas ramificações dos dendritos. Dá-se o nome de sinapse a comunicação entre os neurônios, transmitindo os impulsos nervosos (GUYTON; HALL, 2011).

2.1.2 Neurônio Artificial e Rede Perceptron

A primeira RNA, a Perceptron, foi criada por Rosenblatt no ano de 1958, inspirada no funcionamento de um neurônio artificial por McCulloch e Pitts. A concepção inicial do Perceptron era identificar os sinais elétricos que vinham e se originavam nas fotocélulas, visando mapear os padrões geométricos. Estes padrões eram sintonizados por resistores por meio de um processo de treinamento (GOMIDE, 2012).

A Figura 2 apresenta os componentes básicos do neurônio e da rede Perceptron simples. As entradas são representadas por x_1, x_2, x_3 e x_n , sendo x_n a n -ésima entrada da rede. Cada entrada se conecta com um peso sináptico correspondente (w_1, w_2, w_3 e w_n). A multiplicação entre cada entrada com seu respectivo peso é somada pelo corpo somatório (Σ). O resultante dessa soma é diminuído de um determinado valor chamado limiar de ativação (θ), ou seja, se comparando com limiar sináptico biológico. Ela é responsável por fazer um disparo à saída do neurônio. Este valor é enviado a uma função de ativação $g(u)$, que possui como função limitar a saída do neurônio em um determinado intervalo de valores. Com isso, tem-se a saída da rede, representada por (SILVA; SPATTI; FLAUZINO, 2010).

Figura 2 – Modelo matemático de um neurônio artificial



Fonte: Silva, Spatti, Flauzino (2010), adaptado

Matematicamente, as equações (1) e (2) apresentam o modelo, sendo:

$$u = \sum_{i=1}^n w_n x_n - \theta \quad (1)$$

$$y = g(u) \quad (2)$$

em que:

w_n é o peso sináptico n correspondente a entrada i ;

x_n é a entrada n -ésima da Perceptron;

θ é o limiar de ativação;

u é a saída do neurônio;

g é a função de ativação e;

y é a saída geral da rede.

2.1.3 Treinamento da Rede Neural Perceptron

O princípio básico de uma RNA dá-se pela sua capacidade de receber dados de entrada e processar esses dados. Por meio dos pesos sinápticos, a rede pode classificar ou separar os dados de acordo com o proposto pela sua programação.

Como apresentado por Gomide (2012), o Perceptron usa o algoritmo de treinamento relacionando o erro com a saída desejada para ajustar os pesos sinápticos. Possui, assim, a capacidade de diminuir o erro e recordar as tentativas feitas de aproximar a função a saída desejada (MOLETTA, 2015).

O treinamento da rede tem sua base nos estudos de Donald Hebb. Caracteriza-se pela comparação entre duas saídas de neurônios treinados para que caso estas saídas estejam convergindo para o resultado desejado, o peso sináptico associado com aquele processo é implementado (PAGANI, 2012).

Ainda, Hebb escreve que, os neurônios que tem sua ação em conjunto, ou seja, são estimulados ou exercem estímulo, fortalecem a ligação sináptica, no caso das RNAs, os pesos sinápticos. Observou-se no estudo da programação destes sistemas especialistas que o acréscimo dos pesos ocorre de maneira proporcional, e simétrica (CARDON; MÜLLER, 1994).

Partindo deste estudo de Hebb, foram desenvolvidos vários sistemas de treinamento para as RNAs, sendo o relevante para este trabalho o treinamento supervisionado.

O treinamento da Perceptron é supervisionado, que tem em suas características fundamentais a necessidade de um tutor ou supervisor, para fornecer a rede os exemplos de resposta e de conjuntos de entrada, fazendo com relação a

essas a correção dos pesos de acordo com as épocas de treinamento (BATISTA, 2012).

No processo de treinamento da rede Perceptron, precisa-se seguir certos passos para que o resultado seja o desejado. Tendo em mente as regras utilizadas para tal, dar-se à, a sequência descrita no Quadro 1.

Quadro 1 – Sequência de passos para execução de treinamento de uma Perceptron.

Passo 1	Definir um conjunto de dados conhecido disposto em forma de matriz. Entrar com um conjunto de saída compatível com o de entrada.
Passo 2	Iniciar de forma aleatória um conjunto de pesos, que deve ser iniciado com valores pequenos e disposto em forma de matriz. Definir um número máximo de épocas a ser utilizado para o treinamento.
Passo 3	Efetuar uma multiplicação ponto a ponto entre pesos e entradas a fim de que valor desejado seja igual ao da saída.
Passo 4	Repetir o procedimento do passo 3 até que o número de épocas predefinido seja atingido ou o erro seja inexistente. Este procedimento irá gerar uma resposta (u).
Passo 5	Gerar a saída da função que será $g(u)$. Comparar y e a saída desejada.
Passo 6	Caso a função seja diferente da saída desejada os pesos serão recalculados. O novo valor de pesos se dará pela diferença entre a função y e a saída desejada, multiplicado pelo entrada proposta e pela taxa de aprendizagem. Isso será somado ao valor anterior de peso para definir qual será o peso utilizado no próximo treinamento da rede.
Passo 7	A cada cálculo de pesos, será incrementado no valor de 1 o contador de épocas.
Passo 8	Verificar o limite de épocas e caso possível, reiniciar o treinamento.

Utiliza-se uma base de dados conhecida, e com valores de saída conhecidos se tratando de resultados obtidos pela rede.

Com a base de dados, utiliza-se os pesos sinápticos, que no início do processo são definidos de forma aleatória. A partir do desenvolvimento dos cálculos da rede, estes são excitados ou inibidos (incrementados ou decrementados, respectivamente). Juntamente com os pesos, define-se o limiar de ativação da rede, valor que é adicionado ao cálculo da rede. Estes valores, podem ser calculados

utilizando a mesma regra matemática, pois o valor do limiar de ativação pode ser inserido dentro dos pesos na etapa de cálculo da rede. As equações representam, respectivamente as formas de recálculo de pesos (3) e limiar de ativação (4), respectivamente (SILVA, SPATTI, FLAUZINO, 2010).

$$w_{atual} = w_{anterior} + \eta \cdot (d_n - y) \cdot x_n \quad (3)$$

$$\theta_{atual} = \theta_{anterior} + \eta \cdot (d_n - y) \cdot x_n \quad (4)$$

Nas equações (3) e (4) define-se:

w_{atual} é o peso sináptico que será calculado;

$w_{anterior}$ é o peso sináptico correspondente a última época;

θ_{atual} é o limiar de ativação que será calculado;

$\theta_{anterior}$ é o limiar de ativação correspondente a última época da rede;

η é a taxa de aprendizado da rede;

d_n é a saída n -ésima da Perceptron;

y é a saída obtida pela rede e;

x_n é correspondente à n -ésima entrada da Perceptron.

Em seguida, será definida a taxa de aprendizagem, que define quão rápido o processo irá se desenrolar em direção as respostas desejadas. Uma taxa de aprendizagem muito alta irá proporcionar um funcionamento mais rápido da rede, porém com menor precisão nos resultados e uma taxa de aprendizagem muito baixa será mais confiável, entretanto, irá tomar mais tempo por parte da rede. Este valor é definidor de forma empírica, tendo valores entre 0 e 1.

Por fim, um contador de épocas será responsável por contabilizar quantas vezes o processo ocorreu a fim de que a rede seja capaz de separar as amostras. Este é incrementado a cada vez que a rede completa uma época de treinamento.

O sucesso do treinamento da Perceptron dependerá diretamente da qualidade das amostras apresentadas, da taxa de aprendizagem, da linearidade de separação das amostras e do limite de épocas que a rede foi definida para efetuar seu treinamento (SILVA, SPATTI, FLAUZINO, 2010). O Quadro 2 apresenta o pseudocódigo para o treinamento de uma rede Perceptron de um neurônio.

Quadro 2– Pseudocódigo do treinamento de uma rede Perceptron de um neurônio.

Sendo

- x_n o vetor de entrada com elementos distribuídos por $[-1, x_1, x_2, \dots, x_n]$;
- w_n o vetor de peso com elementos distribuídos por $[\theta_n, w_1, w_2, \dots, w_n]$;
- θ_n o limiar de ativação;
- d_n a resposta desejada;
- $g(u)$ é a função de ativação da saída do neurônio;
- y_n a resposta real após a função de ativação e;
- η a taxa de aprendizagem.

Início do Treinamento.

- 1- Obter o conjunto de amostras de treinamento x_n e sua respectiva saída d_n para cada amostra.
- 2- Inicializar o vetor w_n com valores aleatórios pequenos, a taxa de aprendizagem e o contador de épocas.
- 3- Repetir até o erro ser inexistente ou estoure a quantidade de épocas:
 - 3.1- Cálculo da resposta real, por:
 - 3.1.1- Potencial de Ativação por $u = w \cdot x$
 - 3.1.2- Saída por $y = g(u)$;
 - 3.2- Se $y \neq d$:
 - 3.2.1 Atualização dos pesos por $w \leftarrow w + \eta(d_n - y_n) \cdot x_n$
 - 3.3- Incrementar contador de épocas: época \leftarrow época +1;
 - 3.4- Verificar estouro de épocas;

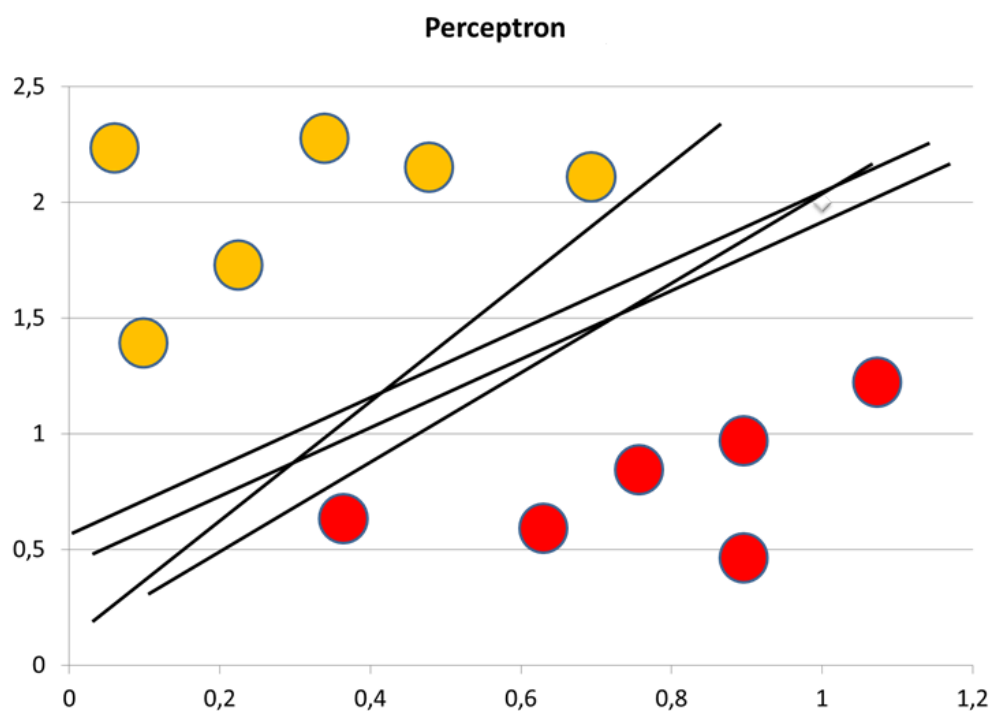
Fim do treinamento.

O Figura 3 apresenta as possibilidades de separações em um hiperplano por uma rede Perceptron utilizando dois grupos de amostras linearmente separáveis e distintas para duas variáveis de entrada. Pode-se perceber a ação da Perceptron, a qual aceita qualquer um dos hiperplanos, desde que estes consigam separar linearmente todas as amostras. As direções do hiperplano são dependentes das condições iniciais, as quais, geralmente, são aleatórias.

2.2 ADALINE

Adaptive Linear Neuron ou Adaline é um modelo de rede neural proposto por Windrow e Hoff nos anos de 1960. Inicialmente sua aplicação era para fazer chaveamento de sistemas eletrônicos de telefonia, sendo esta umas das primeiras utilizações de RNAs no ambiente industrial (SILVA; SPATTI; FLAUZINO, 2010)

Figura 3– Demonstração gráfica de uma rede do tipo Perceptron.



A principal diferença entre um neurônio Adaline e um Perceptron é o algoritmo de aprendizado, ou seja, o seu treinamento. Isso porque a Adaline utiliza o algoritmo de treinamento da regra Delta a fim de minimizar o erro além de utilizar a correção no potencial de ação e não na saída propriamente dita. No algoritmo *Least Mean Square*, ou Erro Quadrático Médio (EQM), que se define pela diferença do quadrado da saída desejada pela saída obtida pela rede, então o erro pode ser expresso como (5) (HAYKIN, 2001)

$$e_n = d_n - x_n w_n \quad (5)$$

em que define-se:

e_n é a função do erro;

d_n é a saída desejada;

x_n é o vetor de erros e;

w_n é o vetor de pesos sinápticos.

Logo, a atualização dos pesos será feita com base na equação (6)

$$w_{(n+1)} = w_n + \eta e_n x_n \quad (6)$$

Na equação (7), está representado o cálculo do erro quadrático médio para o treinamento de redes do tipo Adaline.

$$E_{eqm} \leftarrow E_{eqm} + (d_n - u_n)^2 \quad (7)$$

Na equação (8), está representado o cálculo da precisão da rede Adaline, que é utilizado como critério de parada da mesma.

$$|E_{at} - E_{an}| \leq \varepsilon \quad (8)$$

De forma simplificada, o Quadro 3 – Pseudocódigo do treinamento de uma rede Adaline. apresenta os passos para a realização do algoritmo da Adaline em forma de pseudocódigo.

Quadro 3 – Pseudocódigo do treinamento de uma rede Adaline.

Sendo

- x_n o vetor de entrada com elementos distribuídos por $[-1, x_1, x_2, \dots, x_n]$;
- w_n o vetor de peso com elementos distribuídos por $[\theta_n, w_1, w_2, \dots, w_n]$;
- θ_n o limiar de ativação;
- d_n a resposta desejada;
- $g(u)$ é a função de ativação da saída do neurônio;
- y_n a resposta real após a função de ativação e;
- η a taxa de aprendizagem.

E a função para o erro quadrático médio de p amostras dado por

$$\frac{1}{p} \sum_{k=1}^p (d_n - u_n)^2; \text{ e}$$

ε a precisão do erro quadrático médio.

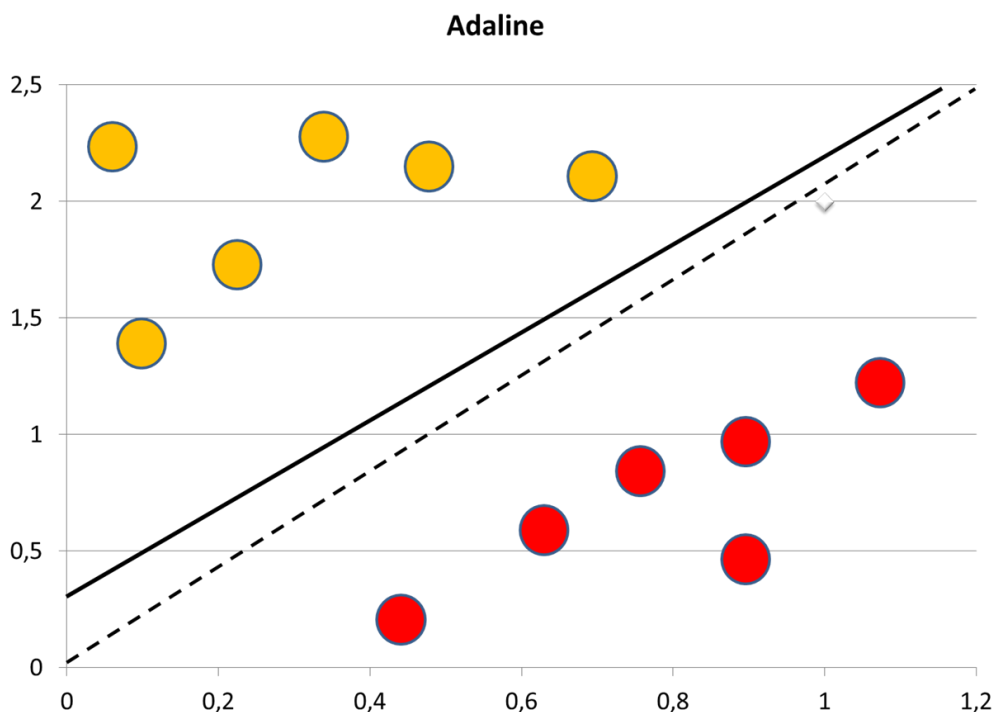
Início do treinamento

- 1- Obter o conjunto de amostras de treinamento x_n e sua respectiva saída d_n para cada amostra.
- 2- Inicializar o vetor w_n com valores aleatórios pequenos, a taxa de aprendizagem e o contador de épocas.
- 3- Repetir até o erro ser menor que a precisão ou estoure a quantidade de épocas:
 - 3.1- Atualizar o erro quadrático médio $E_{an} \leftarrow E_w$
 - 3.2- Para todas as amostras de treinamento:
 - 3.2.1- Calcular o potencial de ativação por $u = w \cdot x$
 - 3.2.2- Atualizar os pesos por $w \leftarrow w + \eta (d_n - u_n) \cdot x_n$
 - 3.3- Incrementar contador de épocas: época \leftarrow época + 1;
 - 3.4- Atualizar o erro quadrático médio $E_{at} \leftarrow E_w$
 - 3.5- Calcular precisão até $|E_{at} - E_{an}| \leq \varepsilon$;

Fim do treinamento.

No Figura 4 apresenta-se como será o hiperplano obtido por meio da precisão de uma rede do tipo Adaline, utilizando dois grupos de amostras distintas. Pode-se perceber que como a Adaline procura um ponto ótimo, qualquer solução para a separação entre as duas classes será próxima do hiperplano ótimo de separação, apresentado na linha tracejada da Figura 4.

Figura 4- Demonstração gráfica de uma rede do tipo Adaline



2.3 SOFTWARE LABVIEW™

O LabVIEW™ é uma plataforma de programação e instrumentação virtual desenvolvida pela National Instruments, que oferece uma vasta gama de opções ao programador utilizando a linguagem gráfica “G” (NATIONAL INSTRUMENTS, 2006). Por sua versatilidade tem aplicações nas mais diversas áreas, como visto em Gonçalves et al. (2011), que faz uso do *software* para análise de dados referente a atletas em atividades aquáticas ou terrestres, e também na área de controle robótico aeronáutico (ANJOS; VILLANI, 2010).

Sua programação se baseia em um diagrama de blocos, oferecendo lógicas, aquisição, controle de dados, e comunicação com diversos dispositivos utilizando

barramentos como Ethernet, USB, Bluetooth, entre outros (NATIONAL INSTRUMENTS, 2006).

No LabVIEW™ tem-se a disposição no VI (Instrumento Virtual), composto por um painel frontal, onde são apresentados os controladores e mostradores do processo em questão, onde o programador pode acompanhar em tempo real os dados, como em um painel físico de instrumentação. No diagrama de blocos é onde ocorre a programação do sistema que será apresentada no Painel Frontal (NATIONAL INSTRUMENTS, 2006).

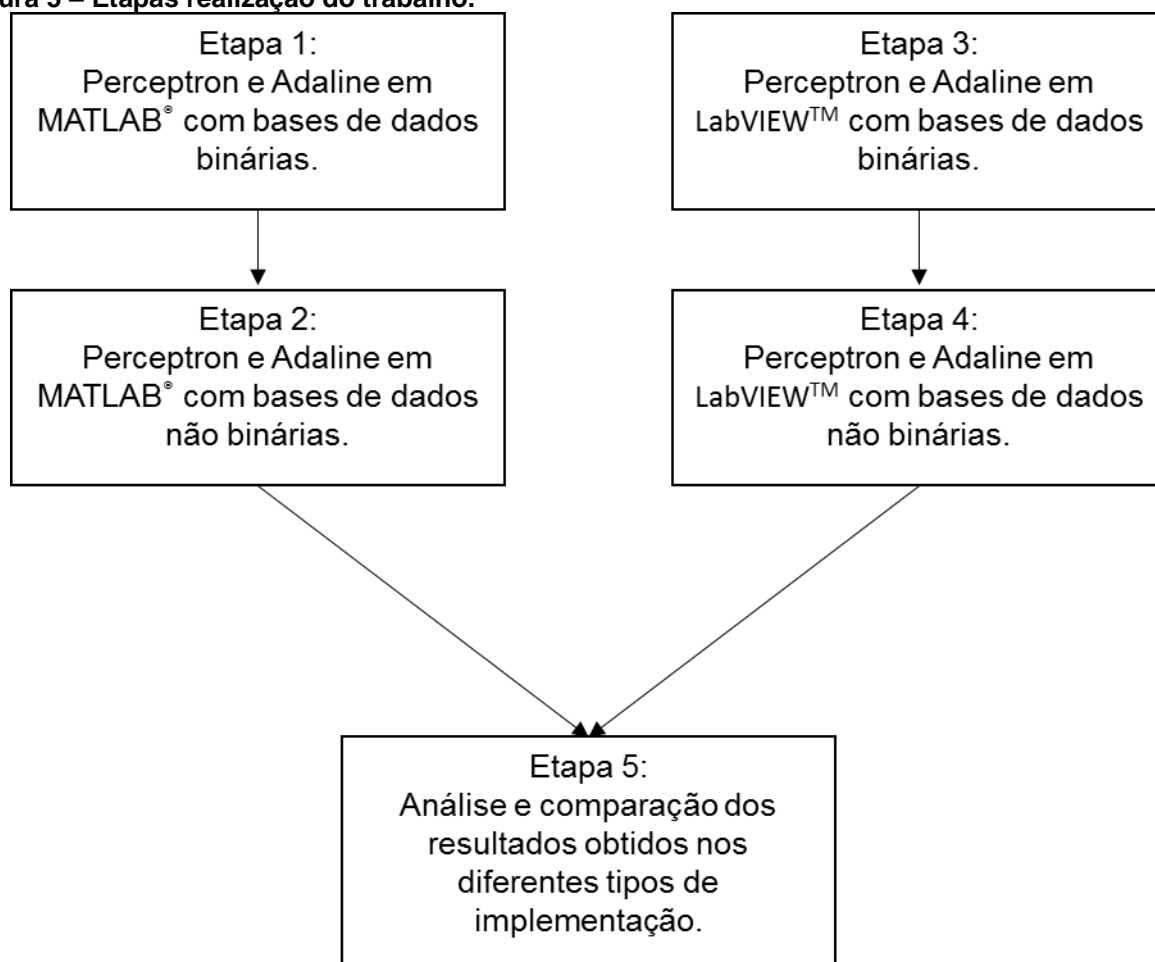
As paletas são janelas onde as ferramentas são apresentadas ao usuário e onde está a função de conectar e posicionar os instrumentos presentes na paleta de controle. Em controle, encontram-se os mostradores, botões, indicadores em geral que são mostrados no painel frontal. Sendo complementar a elas, tem-se a aba de funções, que localiza-se no diagrama de blocos. Tem-se nesta aba a possibilidade de implementar funções booleanas, manipular resultados e sinais, configurar dados de saída, e obter ou simular entradas (NATIONAL INSTRUMENTS, 2006).

As facilidades de manipulação e teste utilizando o LabVIEW™ o tornam um *software* de suma importância nos mais variados campos de desenvolvimento tecnológico.

3 METODOLOGIA

Na Figura 5 são apresentados passos tomados para obtenção dos resultados desejados.

Figura 5 – Etapas realização do trabalho.



No desenvolvimento deste trabalho foram definidas primeiramente quais tipos de treinamentos seriam observados para obtenção do resultado desejado. Após a definição dos treinamentos, foram escolhidas as bases de dados com suas respectivas saídas compatíveis, para que fosse possível o processo de treinamento proposto.

Para o desenvolvimento das RNAs, foram utilizados os *softwares* MATLAB® e LabVIEW™, que, são usualmente programas utilizados para solução de problemas matemáticos ou lógicos e instrumentação virtual, respectivamente.

O objetivo foi comparar a capacidade do LabVIEW™ de desenvolver os mesmos processamentos de dados que o MATLAB, utilizando o mesmo hardware, que foi o processador Intel® Core™ i5-3337U CPU @ 1.80 GHz, com o sistema operacional da Microsoft Windows 8.1®.

As redes foram desenvolvidas em duas linguagens de programação diferentes quanto ao processo de tratamento de dados. Foram coletados os dados referentes às iterações das redes, com um total de 10 amostras de conjuntos de pesos para cada uma das redes. O processo foi efetuado em MATLAB e LabVIEW™.

Para a apresentação dos resultados obtidos, foram geradas tabelas com as informações adquiridas para observação dos dados de pesos, tempo médio de execução dos processamentos e por fim sua quantidade de épocas.

Para a demonstração da evolução dos pesos, foram gerados gráficos com o comportamento da correção de pesos sinápticos. Os dados coletados serviram de base para diferenciar os dois métodos de programação propostos, como também a diferença de processamento de dados.

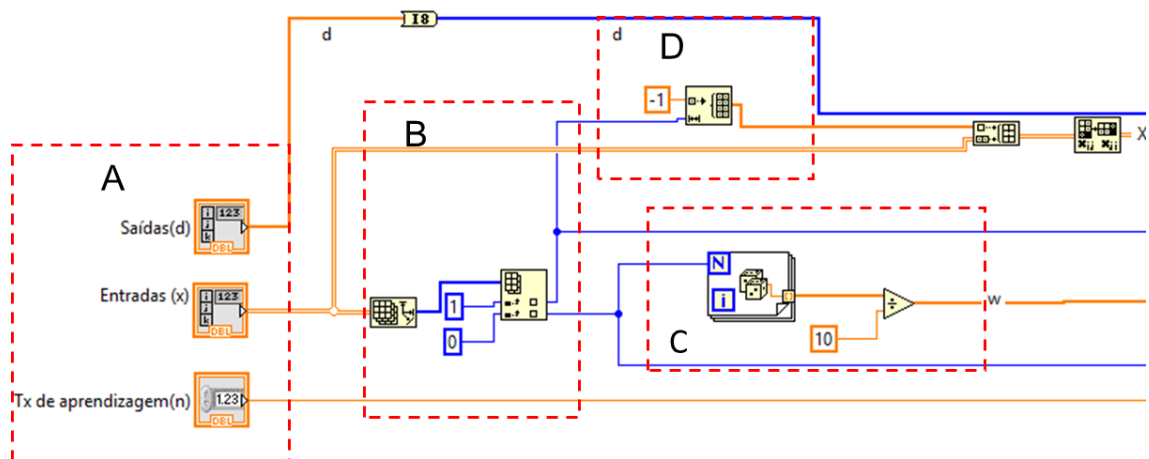
4 DESENVOLVIMENTO

A seguir serão apresentados os códigos desenvolvidos em LabVIEW™ referentes ao treinamento e teste tanto das redes Perceptron e Adaline.

4.1 TREINAMENTO DA REDE PERCEPTRON NO LABVIEW™

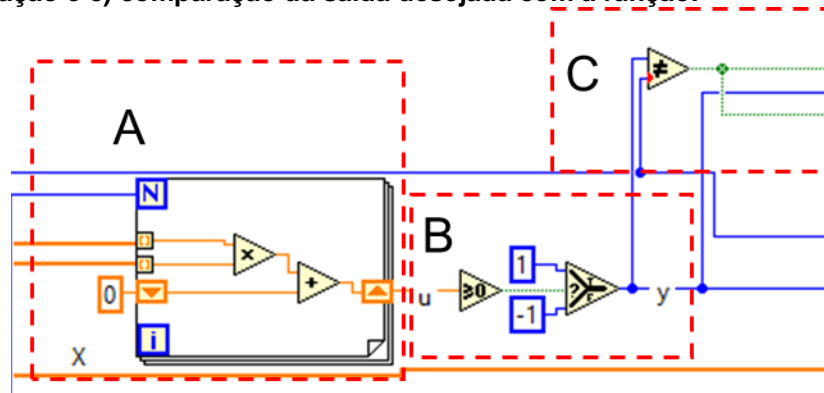
De forma a detalhar todo o processo, a Perceptron é apresentada no Apêndice A para melhor visualização. Na Figura 6 pode-se observar a definição dos dados utilizados para o treinamento da rede. Na **Erro! Fonte de referência não encontrada.6a)**, são apresentados os dados referentes as entradas propostas para treinamento da rede. Para o treinamento da rede, deve-se utilizar um conjunto de dados conhecido. Observa-se que na Figura 6b), será reconhecido o tamanho da entrada, para que sejam gerados pesos equivalentes ao número de entradas, que será efetuado utilizando um bloco do tipo *array size*. Por sua vez, na Figura 6c), tem-se um gerador de números aleatórios, em que será iniciado um valor pequeno, por isso a divisão por 10, ou por um número similar e servirá para a entrada dos pesos sinápticos do treinamento da rede. Já na Figura 6d), tem-se alocado junto com o conjunto de entradas o limiar de ativação, que, para efeito de cálculo, será usualmente alocado com a entrada, que, em forma de matriz, ganhará uma coluna preenchida pelo limiar.

Figura 6 - Etapa de apresentação de dados para treinamento da rede Perceptron de um neurônio onde a) Apresentação de dados, b) reconhecimento do tamanho da matriz, c) geração de pesos sinápticos e d) inserção do limiar de ativação.



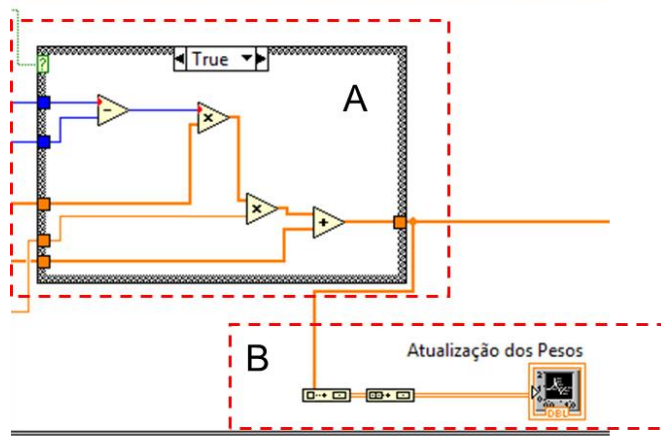
Na Figura 7 apresenta-se o processo de treinamento da rede, seguindo os passos descritos no quadro 2. Na Figura 7a) será feita a etapa de multiplicação dos pesos sinápticos por todos os elementos da entrada, gerando uma resposta (correspondente a u). Esta resposta, por sua vez, terá a saída em uma função, que será gerada na Figura 7b). Na Figura 7c) se fará a comparação da saída desejada com a saída gerada da rede. Este é uma das condições para que a rede pare seu treinamento. Utiliza-se um loop do tipo for para que sejam multiplicadas cada uma das amostras de entrada, por cada um dos pesos sinápticos gerados aleatoriamente. Caso o resultado na Figura 7b), seja de que a saída coincide com o da função gerada, o treinamento irá parar.

Figura 7- Etapa de treinamento dentro do loop *while* da rede Perceptron de uma camada onde a) Loop for de multiplicação das entradas pelos pesos, b) saída do resultado u e geração da função de ativação e c) comparação da saída desejada com a função.



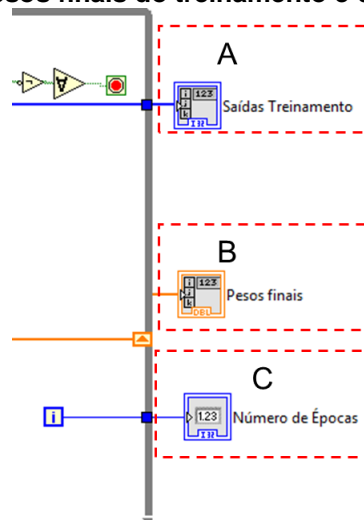
Caso isso não ocorra, em Figura 8a), observa-se o processo de ajuste dos pesos da rede, que, serão reajustados utilizando os valores antigos os valores da diferença observada entre a função de saída da rede e a saída pretendida, e a taxa de aprendizagem. Para tal, multiplica-se a diferença do resultado e da saída desejada, pela taxa de aprendizagem, e pelo valor da entrada. Será então, registrado em um gráfico a correção realizada nos pesos, como visto na Figura 8b).

Figura 8- Etapa de correção de pesos sinápticos da rede Perceptron de um neurônio onde a) Loop while para correção dos pesos e b) registro da atualização dos pesos.



Por sua vez, na Figura 9a), tem-se a tabela de resultados obtidos pela rede, e onde o sinal de diferença atua para a parada da rede. Na Figura 9b), são apresentados os pesos finais obtidos na parada da rede, sendo por número de épocas máximas atingidas, ou pela diferença entre saídas desejadas e saídas geradas pela rede inexistentes. Na Figura 9c) o contador de épocas será incrementado a cada iteração gerada.

Figura 9 - Apresentação dos resultados para a rede Perceptron de uma neurônio onde a) Saidas finais do treinamento, b) pesos finais do treinamento e c) número de épocas totais.



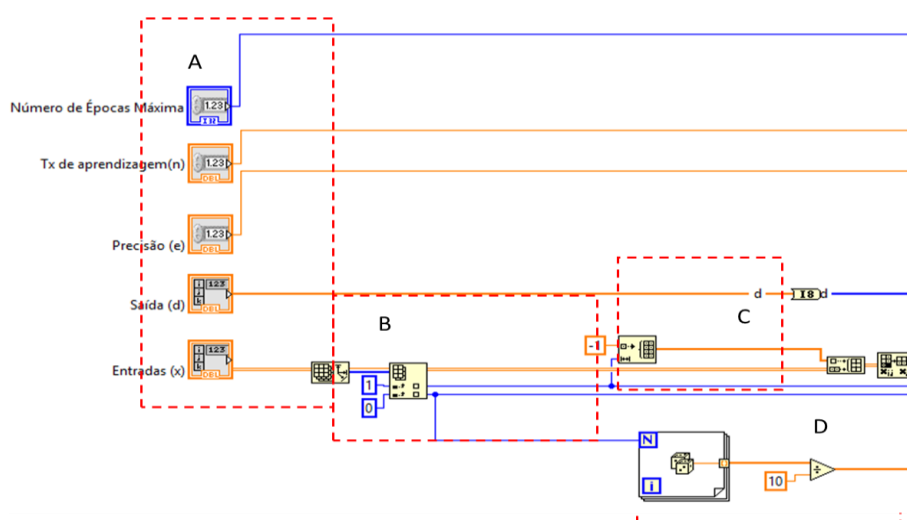
4.2 TREINAMENTO DA REDE ADALINE NO LABVIEW™

Posteriormente a apresentação da rede Perceptron, o treinamento da rede Adaline por meio do programa LabVIEW™ será apresentado no Apêndice B visão do

diagrama de blocos do processo de treinamento de uma rede Adaline implementada em LabVIEW™. Este processo será descrito em partes, tal como foi realizado para o treinamento da rede Perceptron.

Na Figura 10a), tem-se a definição dos dados necessários para o cálculos da rede, sendo eles a taxa de aprendizado, precisão desejada, saídas, entradas e o número de épocas máximo pretendido para a rede. Na Figura 10b), é indicado onde será identificado, utilizando o bloco *array size*, o tamanho da matriz de entrada para que sejam gerados pesos sinápticos em quantidade equivalente ao número de entradas. Seguindo o fluxo de dados da rede, na Figura 10c), tem-se a inserção do limiar de ativação na matriz de entradas, para quesitos de cálculo se faz usual tal processo. Na Figura 10d), são gerados pesos sinápticos de valor pequeno e aleatório, para utilização em cálculo de respostas da rede.

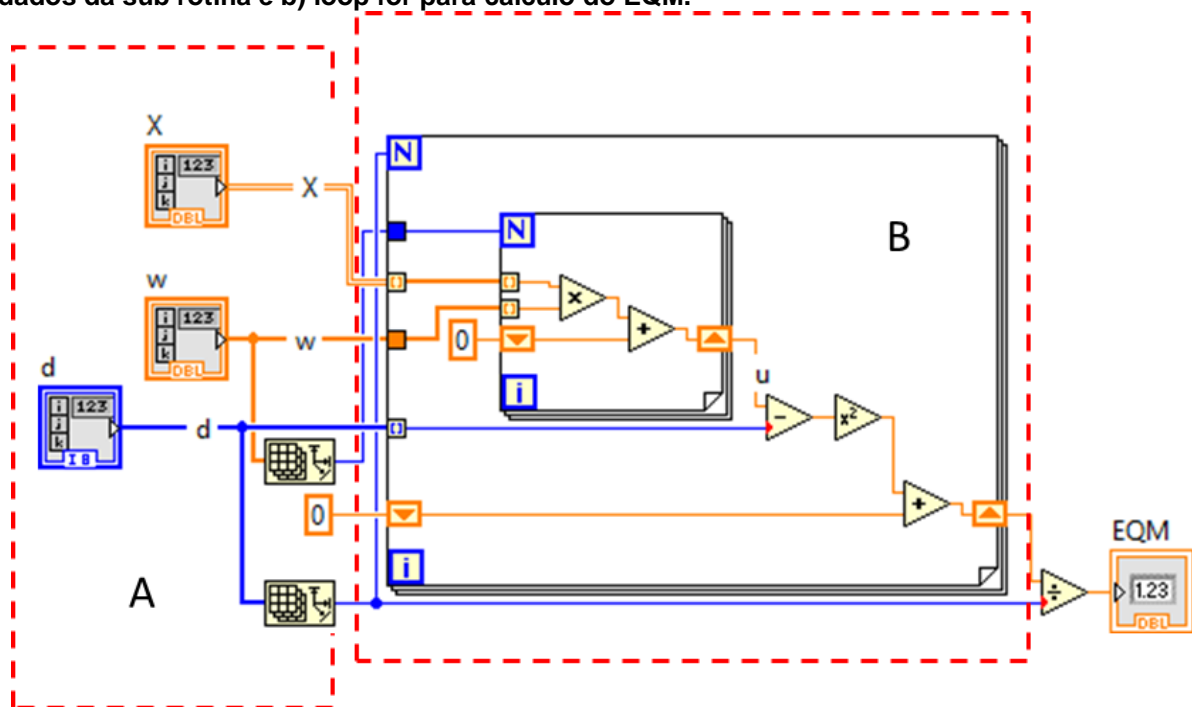
Figura 10 - Apresentação dos dados para o treinamento da rede Adaline onde a) Apresentação dos dados, b) reconhecimento do tamanho da amostra, c) inserção do limiar de ativação e d) geração de pesos sinápticos aleatórios.



Para o processo de treinamento da rede Adaline, é necessário o cálculo do Erro Quadrático Médio (EQM). Tal código é necessário pois este é um dos critérios de parada do treinamento, como apresentado no Quadro 3. Na Figura 11, apresenta-se o processo de cálculo do erro quadrático médio, sendo que na Figura 11a), se apresentam os dados necessários para o cálculo e a definição de como serão apresentados (troca de variáveis). Já na Figura 11b), tem-se o loop for, que será utilizado para as iterações do EQM, em que são utilizados os valores de u e da saída

desejada, efetuando o quadrado da diferença destes valores. Por isso vem a denominação de erro quadrático médio. Esse valor será adicionado ao EQM anterior para obter o EQM a ser usado na próxima iteração. Será então efetuada uma média entre os valores dos erros com relação ao número de iterações, sendo levado em conta o erro gerado na iteração em questão.

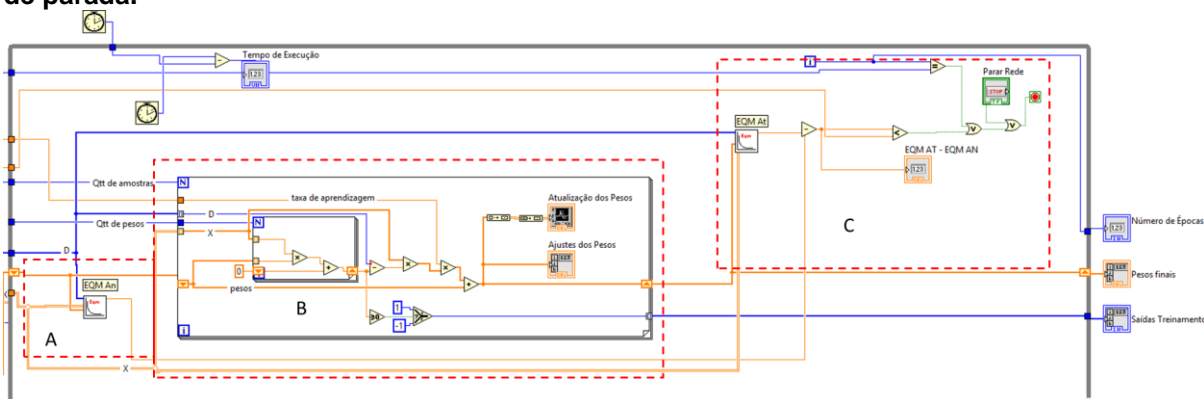
Figura 11 – Diagrama de Blocos do cálculo do erro quadrático médio onde a) apresentação dos dados da sub rotina e b) loop for para cálculo do EQM.



O processo completo de correção dos pesos sinápticos para a rede Adaline é apresentada na Figura 12. Pode-se observar na Figura 12a) uma sub rotina do LabVIEW™, onde está sendo calculado o erro quadrático médio anterior ao resultado gerado pela rede. Na Figura 12b), observa-se o *loop for*, onde serão feitas iterações equivalentes ao número de entradas apresentadas da rede. Nesta etapa serão efetuados os cálculos de multiplicação entre pesos e entradas, a geração da função sinal da rede, porém, antes da geração da função, serão recalculados os pesos sinápticos com base no EQM gerado anteriormente, e, registrados, para geração de bases de dados de resultados. Pode-se ver na Figura 12c) a sub-rotina sendo utilizada novamente para cálculo do novo valor do erro quadrático médio, que será utilizado como critério de parada da rede, sendo efetuada uma diferença entre o EQM anterior e o EQM atual, para obter a precisão. Caso a precisão atinja o valor pré-definido, a rede irá parar. Também será nesta seção onde o número de épocas irá atuar como

critério de parada. Na Figura 12d), pode-se observar os blocos de apresentação de resultados finais da rede, onde estão saídas finais de treinamento, pesos sinápticos finais e o número de épocas onde a rede efetuou a parada.

Figura 12 - Processo de correção dos pesos sinápticos da rede Adaline onde a) sub rotina do EQM, b) loop for para cálculo de saídas e c) comparação de EQM anterior e atual para critério de parada.

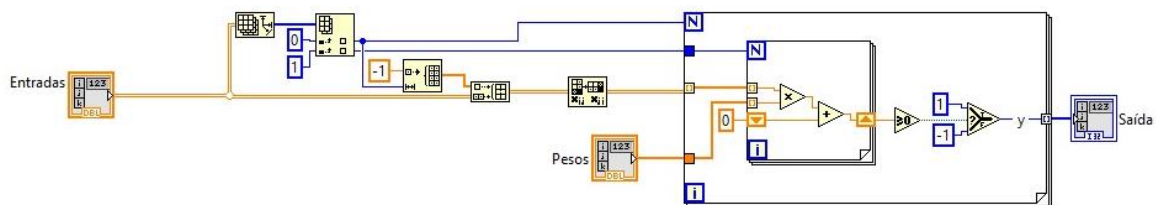


4.3 OPERAÇÃO DA REDE PERCEPTRON E ADALINE NO LABVIEW™

Tem-se na Figura 13, a visão geral da rede neural em sua disposição principal, ou seja, com a formação do diagrama de blocos de modo a utilizar os dados gerados pelos métodos de treinamento citados previamente neste trabalho. Para tal, se faz uso dos métodos de treinamento como uma maneira de obter os pesos sinápticos ótimos para determinado tipo de amostras de entradas. Nesta etapa da rede, as amostras de entrada serão advindas de uma fonte externa, podendo ser uma placa de aquisição de dados, sensores ou uma base de dados já tratados.

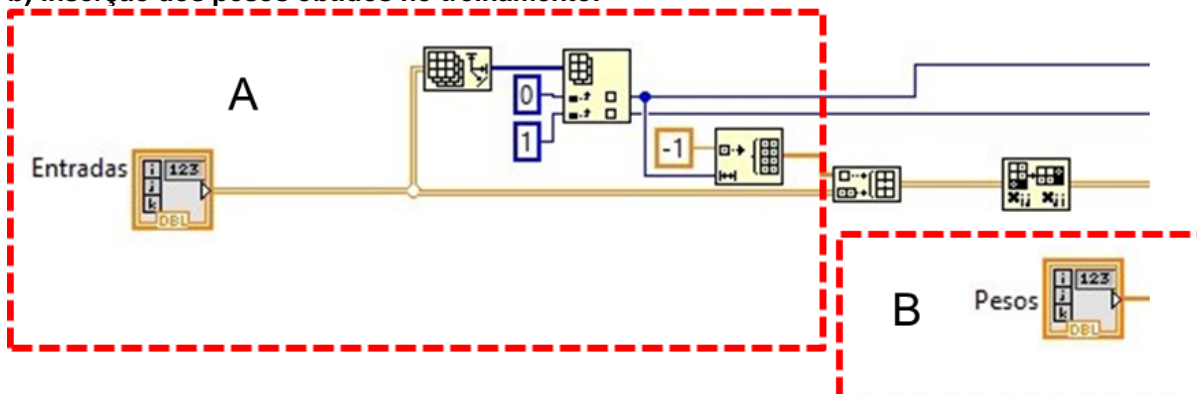
Os métodos de cálculo dos resultados da rede serão semelhantes aos dos processos de treinamento, porém, não haverá etapa para reavaliar o erro, pois se tem o preceito de que isso já foi efetuado no processo de treinamento utilizado. Para a obtenção da saída, se fará o uso de um conjunto de dados desejado, e a utilização dos pesos para o cálculo.

Figura 13 – Diagrama de blocos da fase de operação da rede neural.



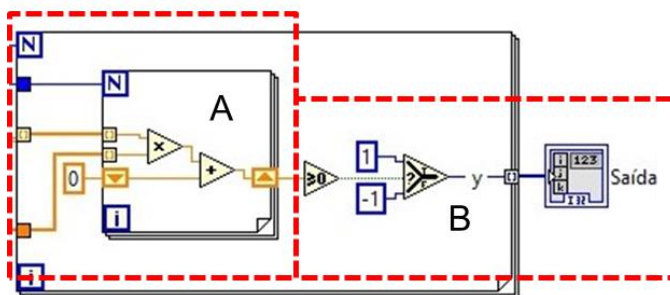
Tal com anteriormente, o diagrama de blocos será explanado. Primeiramente, são definidos os dados do conjunto de entrada que a rede irá utilizar para definir quais serão as saídas do processo. Na Figura 14a), os dados são formatados de modo a serem compatíveis com os métodos de cálculos utilizados nos blocos que seguem. Tendo a formatação dos dados sendo efetuada, será incluso na matriz gerada, o limiar de ativação, pois se faz usual para facilitar as operações posteriores. Na etapa vista, na Figura 14b), os pesos sinápticos finais gerados pelos processos de treinamento da rede serão inseridos pois na etapa de operação da rede, não serão gerados novos pesos.

Figura 14 – Segmento da fase de operação, sendo a) apresentação dos dados de entrada e em b) inserção dos pesos obtidos no treinamento.



Para que o resultado da rede seja tangível, se faz necessária a geração de uma função de ativação para definir se a relação de determinada amostra específica terá afinidade com o resultado desejado. Esta operação, apresentada na Figura 15a), possui o processo de cálculo de u , que foi utilizado no processo de obtenção dos pesos ideais, porém, sem que estes sejam avaliados, apenas utilizados. Figura 15b), será gerada a função de ativação, e, a mesma irá registrar no bloco de saídas, quais serão as amostras que estão e um determinado grupo.

Figura 15 – Segmento da fase de operação, sendo a) loop for para multiplicação dos dados e em b) geração da função de saída e registro das saídas obtidas.



4.4 BASES PARA TESTE DE FUNCIONAMENTO DAS REDES

4.4.1 Bases Binárias

Nas Tabela 1 a Tabela 4, têm-se modelos de bases binárias com os seus respectivos resultados esperados. Os modelos são carregados nos sistemas neurais para que sejam obtidos resultados para estudos de desempenho. Pela necessidade de um conjunto de bases de dados para o treinamento supervisionado da rede, optou-se por bases que apresentam o comportamento de portas lógicas, pois mesmo que as mesmas tenham seu número de entradas alteradas, o comportamento irá se manter além de serem bases conhecidas e linearmente separáveis. Com isso, se fez possível a obtenção de conjuntos de pesos confiáveis.

A Tabela 1 apresenta a base binária relacionada a função (ou porta) AND de três entradas, sendo A, B e C as entradas e S a saída.

Tabela 1– Base binária AND.

A	B	C	S
0	0	0	-1
0	0	1	-1
0	1	0	-1
0	1	1	-1
1	0	0	-1
1	0	1	-1
1	1	0	-1
1	1	1	1

A Tabela 2 é modelo binário do tipo NAND, o resultado esperado é o inverso da AND, pois pode-se dizer que a sua saída é negada, sendo A, B e C as entradas e S a saída.

Tabela 2 – Base binária NAND.

A	B	C	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	-1

A Tabela 3 é o modelo binário do tipo OR, sendo a função (ou porta) OR de três entradas, sendo A, B e C as entradas e S a saída. A Tabela 4 é o modelo do tipo NOR, aonde o seu resultado é OR negado, sendo A, B e C as entradas e S a saída.

Tabela 3 – Base binária OR.

A	B	C	S
0	0	0	-1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Tabela 4 – Base binária NOR.

A	B	C	S
0	0	0	1
0	0	1	-1
0	1	0	-1
0	1	1	-1
1	0	0	-1
1	0	1	-1
1	1	0	-1
1	1	1	-1

4.4.2 Bases de Teste Não-binárias

As bases de teste não-binárias foram obtidas de Silva, Spatti, Flauzino (2010). Essas bases são utilizadas para o teste didático destas redes e por isso foram escolhidas para este trabalho. A Tabela 5 apresenta o conjunto de treinamento da primeira base, sendo referente a classificação de pureza de um tipo de óleo destilado no petróleo.

Tabela 5 - Conjunto de treinamento referente as amostras de óleo.

x ₁	x ₂	x ₃	d
-0,6508	0,1097	4,0009	-1
-1,4492	0,8896	4,4005	-1
2,085	0,6876	12,071	-1
0,2626	1,1476	7,7985	1
0,6418	1,0234	7,0427	1
0,2569	0,673	8,3265	-1
1,1155	0,6043	7,4446	1
0,0914	0,3399	7,0677	-1
0,0121	0,5256	4,6316	1
-0,0429	0,466	5,4323	1
0,434	0,687	8,2287	-1
0,2735	1,0287	7,1934	1
0,4839	0,4851	7,485	-1
0,4089	-0,1267	5,5019	-1
1,4391	0,1614	8,5843	-1
-0,9115	-0,1973	2,1962	-1
0,3654	1,0475	7,4858	1
0,2144	0,7515	7,1699	1
0,2013	1,0014	6,5489	1
0,6483	0,2183	5,8991	1
-0,1147	0,2242	7,2435	-1
-0,797	0,8795	3,8762	1
-1,0625	0,6366	2,4707	1
0,5307	0,1285	5,6883	1
-1,22	0,7777	1,7252	1
0,3957	0,1076	5,6623	-1
-0,1013	0,5989	7,1812	-1
2,4482	0,9455	11,2095	1
2,0149	0,6192	10,9263	-1
0,2012	0,2611	5,4631	1

Fonte: Silva, Spatti, Flauzino, (2010).

A Tabela 6 refere-se ao controle automático de duas válvulas por meio de 4 variáveis de entrada.

Tabela 6 – Conjunto de treinamento referente ao acionamento de duas válvulas.

x_1	x_2	x_3	x_4	d
0,4329	-1,3719	0,7022	-0,8535	1
0,3024	0,2286	0,863	2,7909	-1
0,1349	-0,6445	1,053	0,5687	-1
0,3374	-1,7163	0,367	-0,6283	-1
1,1434	-0,0485	0,6637	1,2606	1
1,3749	-0,5071	0,4464	1,3009	1
0,7221	-0,7587	0,7681	-0,5592	1
0,4403	-0,8072	0,5154	-0,3129	1
-0,5231	0,3548	0,2538	1,5776	-1
0,3255	-2	0,7112	-1,1209	1
0,5824	1,3915	-0,2291	4,1735	-1
0,134	0,6081	0,445	3,223	-1
0,148	-0,2988	0,4778	0,8649	1
0,7359	0,1869	-0,0872	2,3584	1
0,7115	-1,1469	0,3394	0,9573	-1
0,8251	-1,284	0,8452	1,2382	-1
0,1569	0,3712	0,8825	1,7633	1
0,0033	0,6835	0,5389	2,8249	-1
0,4243	0,8313	0,2634	3,5855	-1
1,049	0,1326	0,9138	1,9792	1
1,4276	0,5331	-0,0145	3,7286	1
0,5971	1,4865	0,2904	4,6069	-1
0,8475	2,1479	0,3179	5,8235	-1
1,3967	-0,4171	0,6443	1,3927	1
0,0044	1,5378	0,6099	4,7755	-1
0,2201	-0,5668	0,0515	0,7829	1
0,63	-1,248	0,8591	0,8093	-1
-0,2479	0,896	0,0547	1,7381	1
-0,3088	-0,0929	0,8659	1,5483	-1
-0,518	1,4974	0,5453	2,3993	1
0,6833	0,8266	0,0829	2,8864	1
0,4353	-1,4066	0,4207	-0,4879	1
-0,1069	-3,2329	0,1856	-2,4572	-1
0,4662	0,6261	0,7304	3,437	-1
0,8298	-1,4089	0,3119	1,3235	-1

Fonte: Silva, Spatti, Flauzino, (2010).

5 RESULTADOS

A seguir serão apresentados os valores dos pesos para 10 iterações para cada base de dados, primeiramente para o sistema utilizando o LabVIEW™ e depois para os dados em MATLAB®.

5.1 RESULTADOS USANDO O LABVIEW™ EM REDE PERCEPTRON

Nas Tabela 7, 8, 9 e 10, pode-se observar o conjunto de pesos obtidos para 10 iterações da rede Perceptron, implementada em LabVIEW™, com a base de dados de dados binárias AND, NAND, OR e NOR de três entradas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 7 – Resultados porta AND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	1,828	1,109	1,31	0,733	1,537	1,034	1,801	1,47	1,734	1,467
W1	1,064	0,388	0,793	0,198	0,006	0,208	0,854	1,043	0,63	0,549
W2	0,761	0,719	0,511	0,029	0,816	0,724	0,706	0,065	1,099	0,3521
W3	0,217	0,124	0,0056	0,533	0,715	0,108	0,943	0,423	0,349	0,914
Nº de épocas	159	161	272	95	246	53	160	77	132	99
Tempo (ms)	2	2	3	1	3	1	2	1	2	1
Tempo médio(ms)	1,8									

Tabela 8 – Resultados porta NAND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,009	-0,011	-0,01	-0,008	-0,008	-0,011	-0,001	-0,009	-0,008	-0,01
W1	-0,005	-0,006	-0,004	-0,005	-0,005	-0,006	-0,005	-0,005	-0,004	-0,005
W2	-0,002	-0,004	-0,003	-0,003	-0,003	-0,003	-0,002	-0,002	-0,003	-0,003
W3	-0,002	-0,001	-0,002	-0,001	-0,001	-0,003	-0,002	-0,003	-0,002	-0,002
Nº de épocas	448	755	756	519	631	713	465	770	544	756
Tempo (ms)	5	10	11	6	9	11	7	8	7	12
Tempo médio(ms)	8,6									

Tabela 9 – Resultados porta OR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	0,66	0,692	1,038	0,503	0,825	0,424	1,303	0,332	0,528	0,398
W1	1,442	1,394	1,377	0,507	1,01	0,486	1,31	1,432	1,27	0,399
W2	1,149	1,146	1,38	0,664	0,828	0,508	1,305	0,335	0,531	0,565
W3	0,661	0,693	1,04	1,362	1,565	0,427	1,648	1,147	0,779	1,302
Nº de épocas	196	291	57	150	196	74	76	35	166	148
Tempo (ms)	2	5	1	1	3	2	1	1	1	1
Tempo médio(ms)	1,8									

Tabela 10 – Resultados porta NOR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,001	-0,001	-0,001	-0,001	-0,001	-0,001	-0,001	-0,001	-0,001	-0,001
W1	-0,204	-0,002	-0,002	-0,001	-0,003	-0,002	-0,173	-0,002	-0,001	-0,002
W2	-0,002	-0,002	-0,101	-0,001	-0,037	-0,002	-0,002	-0,099	-0,002	-0,003
W3	-0,002	-0,019	-0,002	-0,002	0,003	-0,002	-0,005	-0,002	-0,002	-0,315
Nº de épocas	59	1158	1015	1234	454	1364	1369	1528	975	917
Tempo (ms)	12	14	15	17	6	18	19	20	12	12
Tempo médio(ms)	14,5									

Por sua vez, nas Tabela 11 e 12, são apresentados o conjunto de pesos obtidos para 10 iterações da rede Perceptron, implementada em LabVIEW™, com a base os dados das amostras de óleo e de acionamento de duas válvulas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 11 – Resultados do conjunto de treinamento referente as amostras de óleo.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,402	-0,642	-0,318	-0,429	-0,564	-0,854	-0,647	-0,347	-0,539	-0,312
W1	0,206	0,332	0,163	0,224	0,297	0,424	0,325	0,177	0,288	0,162
W2	0,396	0,59	0,269	0,401	0,534	0,749	0,58	0,303	0,521	0,266
W3	-0,098	-0,157	-0,077	-0,105	-0,139	-0,206	-0,15	-0,082	-0,133	-0,073
Nº de épocas	338	364	588	276	370	510	562	627	512	499
Tempo (ms)	15	17	25	13	18	23	26	27	25	23
Tempo médio(ms)	21,2									

Tabela 12 – Resultados do conjunto de treinamento referente ao acionamento de duas válvulas

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,325	-0,226	-0,288	-0,248	-0,401	-0,392	-0,132	-0,058	-0,204	-0,423
W1	0,461	0,299	0,396	0,335	0,555	0,543	0,173	0,052	0,271	0,596
W2	0,382	0,259	0,333	0,287	0,467	0,459	0,151	0,058	0,234	0,494
W3	0,006	0,002	0,004	0,003	0,006	0,005	0,001	-0,008	0,002	0,009
W4	-0,269	-0,182	-0,023	-0,202	-0,329	-0,324	-0,107	-0,042	-0,167	-0,35
Nº de épocas	405	303	346	385	501	346	254	174	276	457
Tempo (ms)	33	17	20	20	29	19	14	10	15	21
Tempo médio(ms)	19,8									

5.1.1 Rede Adaline

Nas Tabela 13, 14, 15 e 16, pode-se observar o conjunto de pesos obtidos para 10 iterações da rede Adaline, implementada em LabVIEW™, com a base de dados de dados binárias AND, NAND, OR e NOR de três entradas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 13 – Resultados porta AND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	1,496	1,496	1,496	1,496	1,496	1,496	1,496	1,496	1,496	1,496
W1	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498
W2	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498
W3	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497
Nº de épocas	5021	5014	5015	5040	5030	5014	5003	5007	4988	5024
Tempo (ms)	438	389	420	408	403	419	395	425	396	497
Tempo médio(ms)	419ms									

Tabela 14 – Resultados porta NAND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-1,496	-1,496	-1,496	-1,496	-1,496	-1,496	-1,496	-1,496	-1,496	-1,496
W1	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
W2	-0,0498	-0,0498	-0,0498	-0,0498	-0,0498	-0,0498	-0,0498	-0,0498	-0,0498	-0,0498
W3	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497
Nº de épocas	5151	5162	5107	5134	5134	5138	5102	5120	5101	5157
Tempo (ms)	415	414	414	419	414	429	417	395	418	412
Tempo médio(ms)	414,7									

Tabela 15 – Resultados porta OR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,004	-0,004	-0,004	-0,004	-0,004	-0,004	-0,004	-0,004	-0,004	-0,004
W1	0,496	0,496	0,496	0,496	0,496	0,496	0,496	0,496	0,496	0,496
W2	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497
W3	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497	0,497
Nº de épocas	4285	4357	4304	4280	4413	4317	4324	4349	4390	4316
Tempo (ms)	347	370	360	357	383	363	357	360	355	339
Tempo médio(ms)	359,1									

Tabela 16 – Resultados porta NOR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004
W1	-0,496	-0,496	-0,496	-0,496	-0,496	-0,496	-0,496	-0,496	-0,496	-0,496
W2	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497
W3	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497	-0,497
Nº de épocas	4634	4573	4600	4637	4612	4566	4575	4572	4599	4613
Tempo (ms)	381	395	377	398	383	360	375	408	365	378
Tempo médio(ms)	382									

Posteriormente, nas Tabela 17 e 18, são apresentados o conjunto de pesos obtidos para 10 iterações da rede Adaline, implementada em LabVIEW™, com a base os dados das amostras de óleo e de acionamento de duas válvulas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 17 – Resultados do conjunto de treinamento referente as amostras de óleo.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	2,896	2,896	2,896	2,896	2,896	2,896	2,896	2,896	2,896	2,896
W1	1,542	1,542	1,542	1,542	1,542	1,542	1,542	1,542	1,542	1,542
W2	2,396	2,396	2,396	2,396	2,396	2,396	2,396	2,396	2,396	2,396
W3	-0,703	-0,703	-0,703	-0,703	-0,703	-0,703	-0,703	-0,703	-0,703	-0,703
Nº de épocas	8170	8187	8193	8169	8191	8180	8157	8193	8179	8195
Tempo (ms)	942	971	953	981	1022	971	999	1000	1019	1000
Tempo médio(ms)	985									

Tabela 18 – Resultados do conjunto de treinamento referente ao acionamento de duas válvulas

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837
W1	1,316	1,316	1,316	1,316	1,316	1,316	1,316	1,316	1,316	1,316
W2	1,657	1,657	1,657	1,657	1,657	1,657	1,657	1,657	1,657	1,657
W3	-0,447	-0,447	-0,447	-0,447	-0,447	-0,447	-0,447	-0,447	-0,447	-0,447
W4	-1,192	-1,192	-1,192	-1,192	-1,192	-1,192	-1,192	-1,192	-1,192	-1,192
Nº de épocas	3155	3163	3172	3153	3171	3154	3166	3164	3168	3158
Tempo (ms)	568	492	450	480	462	466	474	490	468	448
Tempo médio(ms)	479,8									

5.2 TABELAS DOS RESULTADOS MATLAB®

Para a comparação dos resultados obtidos com o LabVIEW™, são apresentados os valores obtidos pelas redes com o *software* MATLAB® (apresentadas nos Apêndicas C e D).

5.2.1 Tabelas Perceptron

Nas Tabela 19, 20, 21 e 22, pode-se observar o conjunto de pesos obtidos para 10 iterações da rede Perceptron, implementada em MATLAB®, com a base de dados de dados binárias AND, NAND, OR e NOR de três entradas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 19 – Resultados base AND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	1,153	0,628	0,8	1,04	1,055	1,027	1,125	0,756	0,499	0,561
W1	0,579	0,429	0,429	0,032	0,492	0,348	0,447	0,149	0,194	0,434
W2	0,39	0,151	0,299	0,715	0,551	0,547	0,447	0,138	0,246	0,101
W3	0,57	0,198	0,368	0,322	0,101	0,475	0,664	0,596	0,251	0,118
Nº de épocas	39	27	35	36	47	40	20	2	16	15
Tempo (us)	588	458	554	605	610	483	384	283	203	243
Tempo médio(us)	441,1									

Tabela 20 – Resultados base NAND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,025	-0,012	-0,0139	-0,018	-0,011	-0,019	0,018	-0,0261	-0,019	-0,01
W1	-0,008	-0,004	-0,007	-0,006	-0,002	-0,007	-0,006	-0,014	-0,009	-0,004
W2	-0,011	-0,004	-0,002	-0,007	-0,005	-0,007	0,007	-0,01	-0,008	-0,002
W3	0,012	-0,004	-0,006	-0,011	-0,005	-0,005	-0,009	-0,01	-0,008	-0,004
Nº de épocas	210	194	192	138	172	181	196	139	183	146
Tempo (ms)	36	29	29	26	29	24	36	18	30	19
Tempo médio(ms)	27,6									

Tabela 21 – Resultados base OR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	0,293	0,496	0,39	0,538	0,139	0,602	0,1795	0,154	0,503	0,452
W1	0,309	0,815	0,393	0,991	0,528	0,857	0,18	0,325	0,613	0,957
W2	0,523	0,652	0,511	0,755	0,146	0,988	0,54	0,16	0,513	0,892
W3	0,325	0,503	0,469	0,98	0,756	0,929	0,287	0,747	0,832	0,456
Nº de épocas	1	48	31	1	20	1	31	14	36	21
Tempo (us)	34	594	418	34	386	41	472	242	443	401
Tempo médio(us)	306,5									

Tabela 22 – Resultados base NOR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,004	-0,001	-0,004	-0,003	-0,003	-0,003	-0,003	-0,004	-0,001	-0,004
W1	-0,005	-0,006	-0,007	-0,006	-0,007	-0,004	-0,008	-0,005	-0,096	-0,008
W2	-0,006	-0,002	-0,007	-0,007	-0,006	-0,024	-0,009	-0,009	-0,006	-0,009
W3	-0,009	-0,036	-0,006	-0,008	-0,005	-0,003	-0,013	-0,006	-0,006	-0,004
Nº de épocas	396	288	302	247	288	335	192	337	363	167
Tempo (ms)	49	49	51	41	50	57	22	45	49	20
Tempo médio(ms)	43,3									

A seguir, nas Tabela 23 e 24, são apresentados o conjunto de pesos obtidos para 10 iterações da rede Perceptron, implementada em MATLAB®, com a base os dados das amostras de óleo e de acionamento de duas válvulas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 23 – Resultados do conjunto de treinamento referente as amostras de óleo.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-4,419	-4,436	-4,437	-4,444	-4,43	-4,433	-4,468	-4,453	-4,405	-4,478
W1	2,162	2,158	2,18	2,178	2,167	2,15	2,18	2,177	2,157	2,177
W2	3,495	3,485	3,539	3,527	3,507	3,462	3,529	3,534	3,5	3,521
W3	-1,024	-1,026	-1,031	-1,031	-1,027	-1,001	-1,012	-1,01	-0,999	-1,013
Nº de épocas	1547	1626	1529	1610	1557	1632	1624	1624	1629	1669
Tempo (ms)	225	283	255	156	232	256	307	234	291	271
Tempo médio(ms)	251									

Tabela 24 – Resultados do conjunto de treinamento referente as aberturas de válvulas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-0,347	-0,356	-0,293	-0,331	-0,372	-0,309	-0,344	-0,238	-0,383	-0,476
W1	0,426	0,355	0,324	0,327	0,351	0,355	0,336	0,3	0,328	0,614
W2	0,392	0,375	0,325	0,349	0,383	0,343	0,366	0,27	0,39	0,544
W3	0,001	-0,01	0,001	-0,011	-0,034	-0,002	-0,006	-0,001	-0,022	0,008
W4	-0,274	-0,272	-0,23	-0,251	-0,283	-0,242	-0,264	-0,189	-0,275	-0,386
Nº de épocas	102	131	114	151	175	123	132	97	126	165
Tempo (ms)	21	17	15	28	23	24	23	10	16	30
Tempo médio(ms)	20,7									

5.2.2 Tabelas Adaline

Nas Tabela 25, 26, 27 e 28, pode-se observar o conjunto de pesos obtidos para 10 iterações da rede Adaline, implementada em MATLAB®, com a base de dados de dados binárias AND, NAND, OR e NOR de três entradas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 25 – Resultados porta AND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	1,497	1,497	1,497	1,497	1,497	1,497	1,497	1,497	1,497	1,497
W1	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498
W2	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498
W3	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498	0,498
Nº de épocas	2065	1887	1810	1763	1963	2034	1951	1941	2088	2127
Tempo (ms)	78	61	59	57	63	103	64	63	67	82
Tempo médio(ms)	69,7									

Tabela 26 – Resultados porta NAND 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-1,497	-1,497	-1,497	-1,497	-1,497	-1,497	-1,497	-1,497	-1,497	-1,497
W1	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
W2	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
W3	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
Nº de épocas	2517	2481	2429	2521	2433	2480	2426	2441	2409	2447
Tempo (ms)	86	63	59	62	60	60	59	73	64	65
Tempo médio(ms)	65,1									

Tabela 27 – Resultados porta OR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002
W1	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501
W2	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501
W3	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501	0,501
Nº de épocas	1385	1465	1814	1774	1913	1676	1808	1053	1288	1625
Tempo (ms)	34	36	45	45	50	41	45	26	32	40
Tempo médio(ms)	39,4									

Tabela 28 – Resultados porta NOR 3 entradas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002	0,002
W1	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
W2	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
W3	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498	-0,498
Nº de épocas	2313	2290	2246	2218	2334	2273	2305	2199	2266	2274
Tempo (ms)	84	55	61	55	56	556	55	54	58	55
Tempo médio(ms)	108,9									

Por sua vez, nas Tabela 29 e 30, são apresentados o conjunto de pesos obtidos para 10 iterações da rede Adaline, implementada em MATLAB®, com a base os dados das amostras de óleo e de acionamento de duas válvulas, respectivamente, com o número de épocas necessárias para cada iteração e o tempo médio para convergência.

Tabela 29 – Resultados do conjunto de treinamento referente as amostras de óleo

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-2,92	-2,92	-2,92	-2,92	-2,92	-2,92	-2,92	-2,92	-2,92	-2,92
W1	1,552	1,552	1,552	1,552	1,552	1,552	1,552	1,552	1,552	1,552
W2	2,389	2,389	2,389	2,389	2,389	2,389	2,389	2,389	2,389	2,389
W3	-0,705	-0,705	-0,705	-0,705	-0,705	-0,705	-0,705	-0,705	-0,705	-0,705
Nº de épocas	7062	7033	6965	6893	6943	6986	6841	7139	7075	6946
Tempo (ms)	240	212	213	230	211	212	208	220	216	208
Tempo médio(ms)	217									

Tabela 30 – Resultados do conjunto de treinamento referente as aberturas de válvulas.

Iteração	1	2	3	4	5	6	7	8	9	10
W0	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837	-1,837
W1	1,315	1,315	1,315	1,315	1,315	1,315	1,315	1,315	1,315	1,315
W2	1,659	1,659	1,659	1,659	1,659	1,659	1,659	1,659	1,659	1,659
W3	-0,449	-0,449	-0,449	-0,449	-0,449	-0,449	-0,449	-0,449	-0,449	-0,449
W4	-1,196	-1,196	-1,196	-1,196	-1,196	-1,196	-1,196	-1,196	-1,196	-1,196
Nº de épocas	1436	1385	1415	1429	1447	1439	1434	1454	1394	1421
Tempo (ms)	48	47	44	44	44	44	44	44	43	44
Tempo médio(ms)	44,6									

Observa-se que nas Tabela 7 a 10 (correspondentes as amostras binárias implementadas em uma rede Perceptron em LabVIEW™), e nas Tabelas 19 a 22 (correspondente as implementações em MATLAB®), o tempo de obtido de processamento é aproximadamente seis vezes menor e o número de épocas significativamente reduzido. Ainda nas bases de dados binárias, foram observadas nas Tabelas 13 a 16 (Adaline em LabVIEW™) e nas Tabelas 25 a 28 (Adaline em MATLAB®), pode-se observar que os resultados seguiram a tendência onde o MATLAB® é mais rápido no processamento dos dados.

Nas Tabelas 11 e 12 (Perceptron em LabVIEW™) e nas Tabelas 23 e 24 (Perceptron em MATLAB®), correspondentes as bases não-binárias, pode-se perceber que o LabVIEW™ foi mais eficaz no processamento, com poucos um

número de épocas menor para obter a resposta, resultando em um tempo de processamento menor.

Por fim, nas Tabela 17 e 18 (Adaline em LabVIEW™) e nas Tabelas 29 e 30 (Adaline em MATLAB®), percebe-se que o tempo de execução no ambiente MATLAB® é aproximadamente oito vezes menor no ambiente LabVIEW™, devido a velocidade de processamento do próprio MATLAB®.

5.3 GRÁFICOS DA EVOLUÇÃO DOS PESOS

Nesta seção estão apresentados os gráficos que representam a evolução dos pesos sinápticos com as bases de dados referidas na Tabela 5. O treinamento foi efetuado por uma rede Perceptron e uma rede Adaline, respectivamente, utilizando a plataforma LabVIEW™. Estão descritos nos Gráfico 1 e 2 o número de épocas referentes a aquisição dos pesos e o padrão de evolução dos mesmos.

Gráfico 1 – Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Perceptron LabVIEW™.

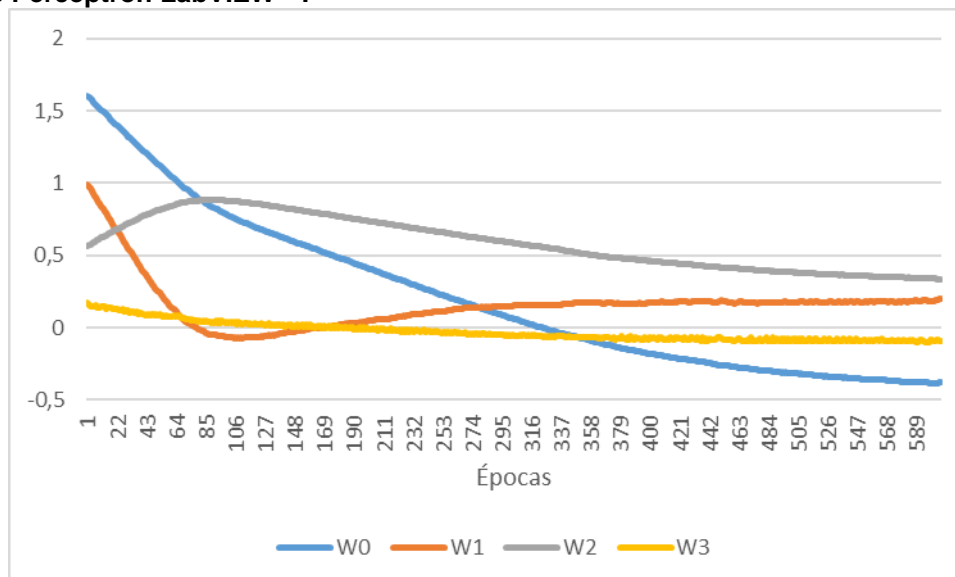
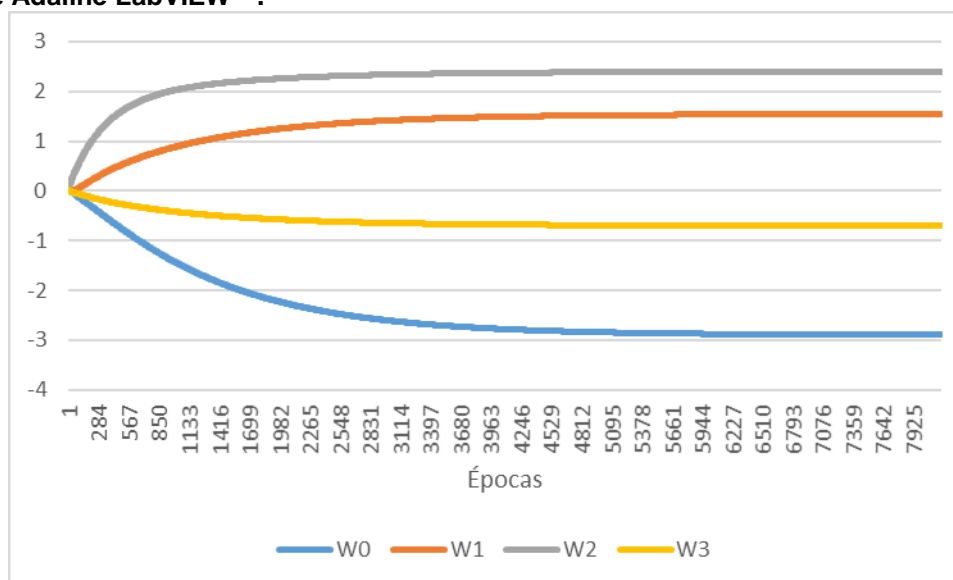


Gráfico 2 – Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Adaline LabVIEW™.



Os Gráfico 3 e 4 representam a evolução dos pesos sinápticos com as bases de dados referidas na Tabela 5. O treinamento foi efetuado por uma rede Perceptron e uma rede Adaline, respectivamente, utilizando a plataforma MATLAB®. Além disso, são apresentados o número de épocas referentes a aquisição dos pesos e o padrão de evolução dos mesmos.

Gráfico 3 - Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Perceptron MATLAB®.

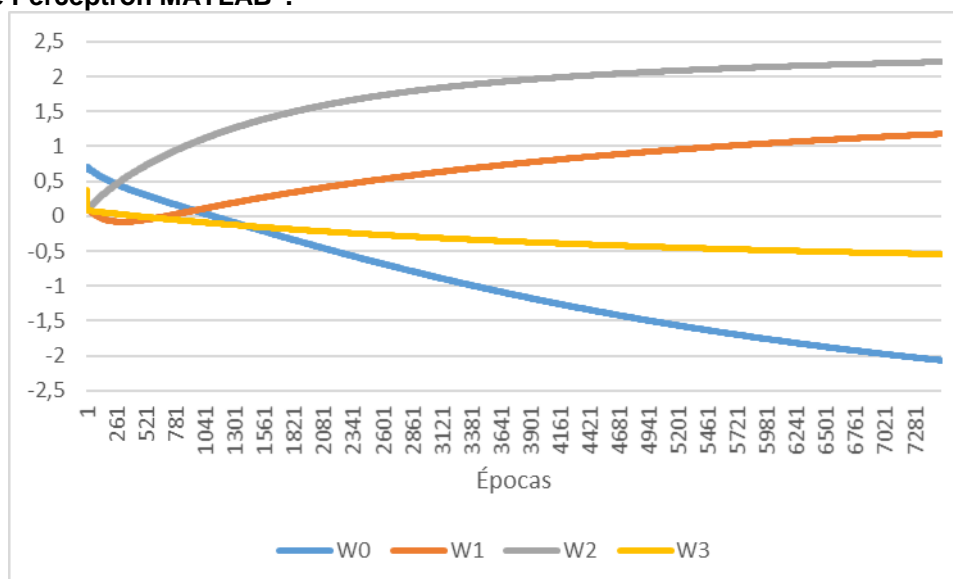
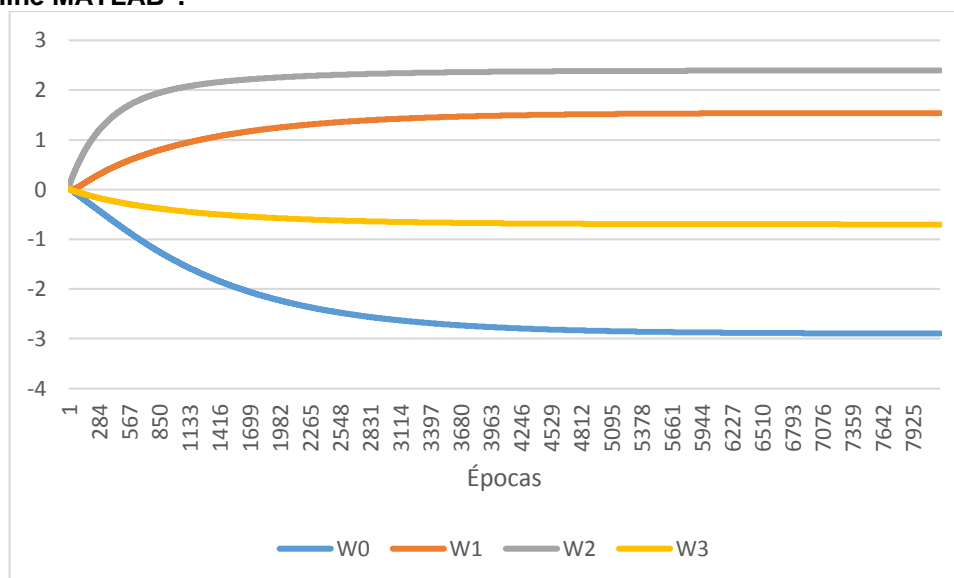


Gráfico 4- Evolução dos pesos sinápticos referentes á amostra de óleos implementada em uma rede Adaline MATLAB®.



Pode-se observar nos Gráficos de 1 a 4 a evolução dos pesos sinápticos gerados pela base de dados não binária presente na Tabela 5, na qual percebe-se que tanto a rede Adaline quanto a Perceptron apresentam um mesmo padrão nos valores de pesos gerados pelas iterações com a mesma base de dados. A diferença observada ocorre no número de épocas necessárias para que as redes atinjam seu critério de parada. Com esta apresentação, percebe-se que o mesmo comportamento ocorre nas duas plataformas, não havendo diferenças no resultado das redes nos dois ambientes de programação.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento do trabalho apresentado possibilitou um estudo na área de instrumentação virtual e processamento de dados em ambiente LabVIEW™ para demonstrar como a implementação de RNAs podem ser utilizadas nesta plataforma em processos de instrumentação virtual.

Com a comparação do desempenho do *software* LabVIEW™, observou-se as suas capacidades de processamento em conjunto com a programação de redes neurais em seu diagrama de blocos. Em comparação a utilização do MATLAB®, a plataforma LabVIEW™ mostrou-se eficaz, quanto ao tempo de resposta e processamento, com as diferentes bases de dados apresentadas e versátil no desenvolvimento e apresentação de resultados referentes as redes neurais Perceptron e Adaline. Por se tratar de um ambiente voltado para o processamento de dados e com entradas essencialmente em forma de matriz, o MATLAB® permite um processamento mais rápido dos dados com o *hardware* utilizado e também solucionou os problemas apresentados em menor tempo.

Por sua vez, o LabVIEW™, que utiliza a programação em blocos, permite diversos tipos de entrada de dados, se torna mais eficaz na comunicação com o ambiente externo para aquisição de dados. Estes dados necessitam de formatação e processamento de acordo com a utilização e a operação pretendida, portando, toma mais tempo de processamento do que o MATLAB®, sendo então um aproximadamente seis vezes mais lento ao processar dados.

Outra capacidade que se destaca no LabVIEW™ é a de supervisão de processos por meio do Painel Frontal, que apresenta de forma clara os dados referentes ao diagrama de blocos. Essa função demonstrou-se muito eficaz para a aplicação proposta com as RNAs

Ambos os ambientes permitem a criação de aplicações fechadas, sendo uma ferramenta mais utilizada e intuitiva no LabVIEW™, portanto, mostrando-se como um diferencial do mesmo e criando uma possibilidade de utilização de RNAs em aplicação fechada para processamentos de dados futuros.

Para oportunidades de prosseguimento deste trabalho, pode-se destacar a possibilidade de estudo similar com outras redes neurais, a criação de uma aplicação fechada para o processamento de dados de diversos processos e outras aplicações deste tipo de programação em instrumentação virtual.

REFERÊNCIAS

AGUIAR, F. G. **UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS PARA DETECÇÃO DE PADRÕES DE VAZAMENTO INDUTOS**. São Carlos: Universidade de São Paulo, 2010.

ANJOS, J. M. S.; VILLANI, E. **MODELAGEM E VERIFICAÇÃO DE UMA PROPOSTA PARA ARQUITETURA DE CONTROLE DE UM EFETUADOR ROBÓTICO BASEADA EM LABVIEW™**. 2010.

AZEVEDO, F. M. DE; BRASIL, L. M.; OLIVEIRA, R. C. L. DE. **Redes neurais com aplicação em controles e em sistemas especialistas**. Florianópolis: Visual Books Editora, 2000.

CARDON, A.; MÜLLER, D. N. **Introdução Às Redes Neurais Artificiais**. Porto Alegre: Universidade Federal do Rio Grande do Sul, 1994.

CAVALCANTE, M. M. et al. A Plataforma Arduino para fins didáticos: Estudo de caso com recolhimento de dados a partir do PLX-DAQ. n. XXXIV Congresso da Sociedade Brasileira de Computação-CSBC, 2014.

CUNHA, F. L. et al. O Uso de Redes Neurais Artificiais Para o Reconhecimento de Padrões em uma Prótese Mioelétrica de Mão. n. Laboratório de Biocibernética e Engenharia de Reabilitação – D.E.E. – E.E.S.C./ USP, 2007.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. 1ª ed. Porto Alegre: Editora da UFRGS, 2009.

GOMIDE, F. A. REDES NEURAIS ARTIFICIAIS PARA ENGENHARIA E CIÊNCIAS APLICADAS: CURSO PRÁTICO. **Resenha Livro**, v. Vol.23 no.5, n. Revista Controle & Automação, 2012.

GUYTON, A. C.; HALL, J. E. **Tratado de Fisiologia Médica**. 12ª ed. São Paulo: Elsevier Editora Ltda., 2011.

HAYKIN, S. **Redes Neurais Princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001.

MOLETTA, EDUARDO. **REDES NEURAIS COM ESTADOS DE ECO APLICADAS EM CONTROLE DEPENDENTE DOS ESTADOS**. Ponta Grossa: Universidade Tecnológica Federal do Paraná, 2015.

MONTANAR, T. **Histologia: Texto, atlas e roteiro de aulas práticas**. 3ª ed. Porto Alegre: Edição do Autor, 2016.

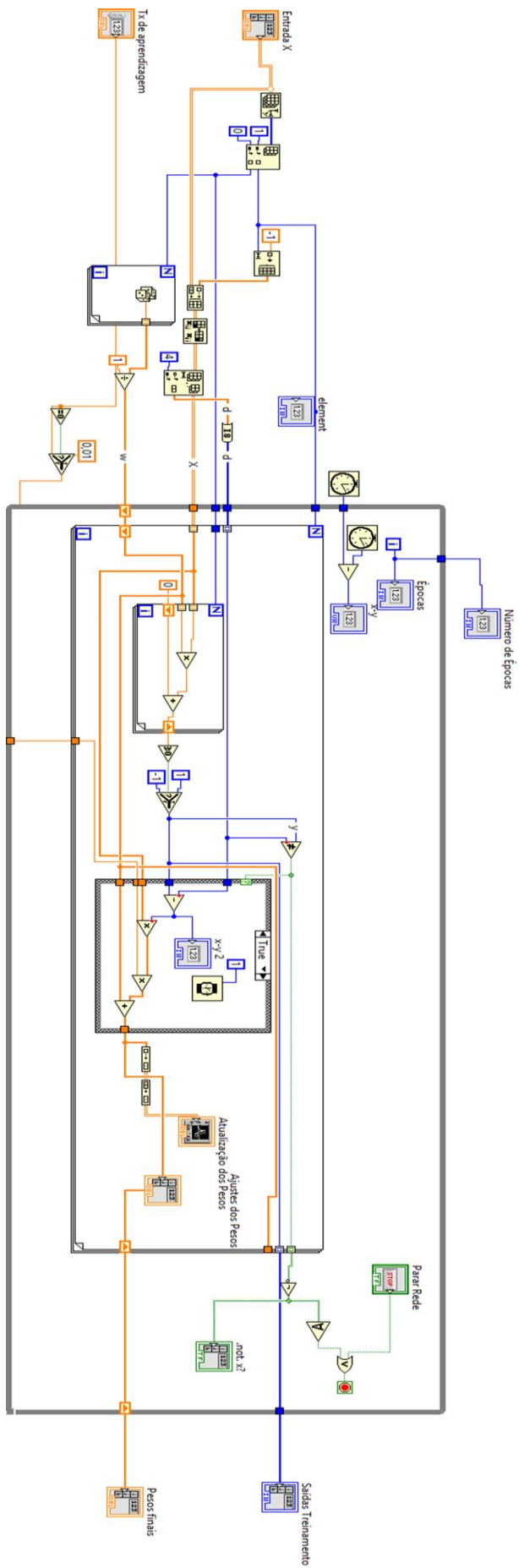
NATIONAL INSTRUMENTS. **Líder global em soluções de teste, medição e controle**, 2006. Disponível em: <<http://www.ni.com/pt-br.html>>

PAGANI, D. H. **Estudo Comparativo entre redes neurais artificiais e redes neurais pulsadas usando MATLAB®**. Cascavel: Universidade Estadual do Oeste do Paraná, 2012.

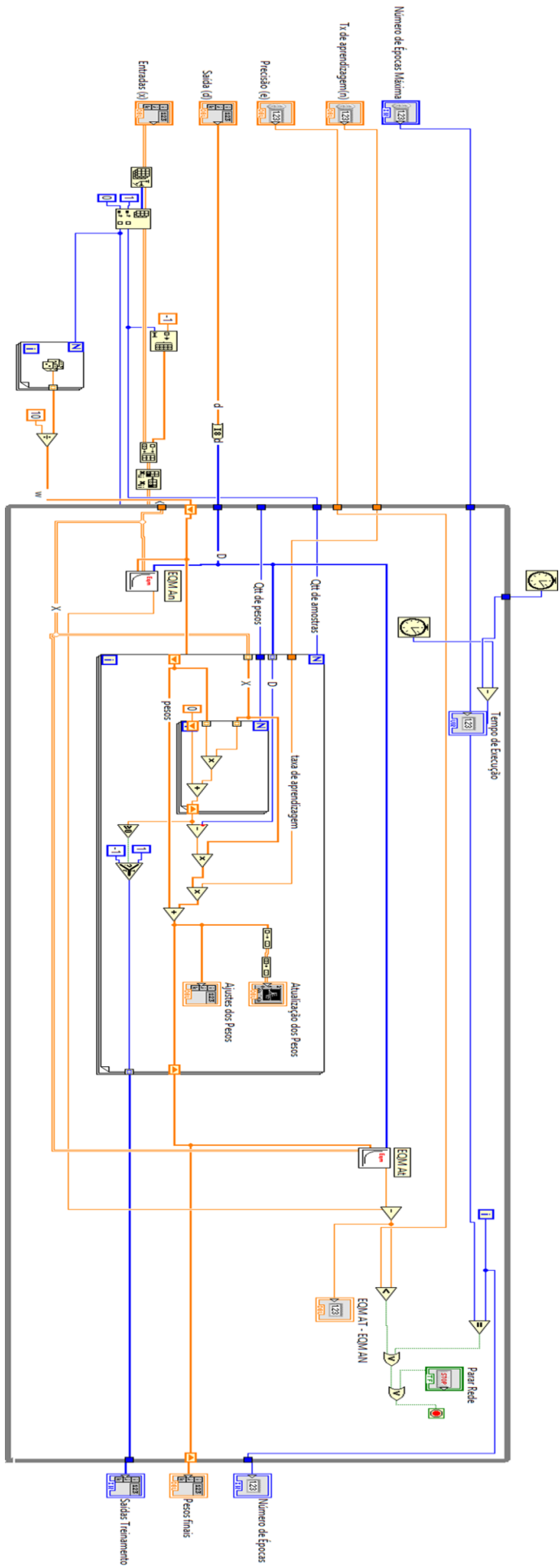
SALLE CTP et al. **Uso de Redes Neurais Artificiais para Estimar Parâmetros de Produção de Galinhas Reprodutoras Pesadas em Recria**. n. CDPA-Faculdade de Veterinária / UFRGS, 2012.

SILVA, I. N. DA; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber Editora Ltda., 2010.

APÊNDICE A - Diagrama da rede Perceptron em LabVIEW™.



APÊNDICE B - Diagrama da rede Adaline em LabVIEW™.



APÊNDICE C - Código da rede Perceptron em MATLAB®

CÓDIGO EM C DESENVOLVIDO PARA O EXPERIMENTO DA REDE ADALINE EM MATLAB®

```

tic; % Medição do tempo de resposta.
K=TreinamentoEntradasA; % Entradas.
[m,n]=size(K); % Identifica o tamanho da matriz.
teta=ones(m,1)*(-1); % Insere o limiar de ativação.
X=[teta,K]; %Define matriz para cálculo.

D=TreinamentoSadass; % Saída desejada.

epoca=0; % Contador de épocas zerado.
N=0.0025; % Define a taxa de aprendizado.
W=rand((n+1),1); % Gerador de pesos iniciais aleatórios.
erro=1; % Define erro inicial como critério de parada.

while erro~=0 && epoca<10000 %Loop para cálculo da rede.
u=X*W; % Multiplicação das entradas sobre os pesos.

    for i=1:m % Define uma iteração para cada amostra proposta.
        if u(i)>=0 % % Função Sinal para 1 , no caso quando u(i) for
maior que 0 atribui o valor de 1 para y(i).
            y(i)=1;
        end
        if u(i)<0 % Função Sinal para 0, no caso quando u(i) for menor
que 0 atribui o valor de -1 para y(i).
            y(i)=-1;
        end
    end
    tf=isequal(y',D); % Conferencia da saida da função com o valor da
saida desejada.
    if tf==0 % Caso a saida não seja conferente com o resultado desejado.
        W=W+(N*(D-y')'*X)'; % Correção do valor do peso para aproximação
com o resultado desejado.
        erro=1; % Erro igual a 1 quando houver correção de pesos.
    else erro=0; % Quando não houver correção de pesos, ativa critério de
parada.
    end

    epoca =epoca+1 % Incrementa contador de épocas.
end
toc; % Medição do tempo de resposta.

W % Plot dos valores de pesos obtidos.

```

APÊNDICE D - Código da rede Adaline em MATLAB®

CÓDIGO EM C DESENVOLVIDO PARA O EXPERIMENTO DA REDE ADALINE EM MATLAB®.

```

tic % Medição do tempo de resposta.
K=TABELAENTRADA;% Entradas.
D=RNAND; % Saida desejada.
[m,n]=size(K); % Identifica o tamanho da matriz.

teta=ones(m,1)*(-1); %Insere o limiar de ativação.
X=[teta,K]; %Define matriz para cálculo.

epoca=0; % Contador de épocas zerado.
N=0.0025; % Define a taxa de aprendizado.
W=(rand((n+1),1)); % Gerador de pesos iniciais aleatórios.
aux=1; % Critério de parada.
E=0; %Define erro inicial como zero.
while aux~=0 && epoca<100000 %Loop para cálculo da rede.
u=X*W; % Multiplicação das entradas sobre os pesos.
    for i=1:m % Define uma iteração para cada amostra proposta.
        if u(i)>0 % Função degrau para 1 , no caso quando u(i) for
maior que 0 atribui o valor de 1 para y(i).
            y(i)=1;
        end

        if u(i)==0 % Função degrau para 1 , no caso quando u(i) for
maior que 0 atribui o valor de 1 para y(i)
            y(i)=0;
        end

        if u(i)<0 % Função degrau para -1.
            y(i)=-1;
        end
    end
    Eqm=0; %Define erro quadrático médio como 0.
    for i=1:m %Define uma iteração para cada amostra proposta.
        Eqm=Eqm+(D(i)-u(i))^2; %Cálculo do erro quadrático médio.
    end
    Ean=Eqm/m; % Média de valores gerados no cálculo do EQM.

    W=W+(N*(D-u) '*X)'; % Correção do valor do peso para aproximação
com o resultado desejado.
    u=X*W; % Multiplicação das entradas sobre os pesos
    for i=1:m % Define uma iteração para cada amostra proposta.
        if u(i)>=0 % Função degrau para 1 , no caso quando u(i) for
maior que 0 atribui o valor de 1 para y(i)
            y(i)=1;
        end
        if u(i)<0 % Função degrau para -1.
            y(i)=-1;
        end
    end
end

```