



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CAMPUS CARAÚBAS
CURSO DE CIÊNCIA E TECNOLOGIA

ALEFJOHN FEITOSA AMORIM

**IDENTIFICAÇÃO DO GLAUCOMA UTILIZANDO REDES NEURAIS COM
IMPLEMENTAÇÃO EM FPGA**

CARAÚBAS – RN

2016

ALEFJOHN FEITOSA AMORIM

**IDENTIFICAÇÃO DO GLAUCOMA UTILIZANDO REDES NEURAIIS COM
IMPLEMENTAÇÃO EM FPGA**

Monografia apresentada à Universidade Federal Rural do Semi-Árido (UFERSA), como exigência final para obtenção do título de Bacharel em Ciência e Tecnologia.

Orientador: Francisco de Assis Brito Filho – UFERSA.

CARAÚBAS – RN

2016

ALEFJOHN FEITOSA AMORIM

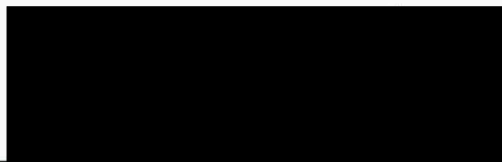
**IDENTIFICAÇÃO DO GLAUCOMA UTILIZANDO REDES NEURAIS COM
IMPLEMENTAÇÃO EM FPGA**

Monografia apresentada à Universidade Federal Rural do Semi-Árido (UFERSA), como exigência final para obtenção do título de Bacharel em Ciência e Tecnologia.

Orientador: Francisco de Assis Brito Filho – UFERSA.

APROVADO EM 13/05/2016

BANCA EXAMINADORA



Dr. Francisco de Assis Brito Filho
Orientador



Dr. Tania Luna Laura



Dr. Dorgival Albertino da Silva Junior

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tornar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

A524i Amorim, AlefJohn Feitosa .
Identificação do glaucoma utilizando redes
neurais com implementação em FPGA / AlefJohn Feitosa
Amorim. - 2016.
65 f. : il.

Orientador: Francisco de Assis Brito Filho.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2016.

1. Rede Neural Artificial. 2. Glaucoma. 3.
Processamento de Imagem Digital . 4. FPGA. I. Brito
Filho, Francisco de Assis, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

À meu pai, João Pereira e a minha mãe,
Alessandra de Brito. As duas pessoas que mais
admiro, respeito e amo.

DEDICO

AGRADECIMENTOS

Agradeço as pessoas mais importantes da minha vida, meu pai João Pereira e minha mãe Alessandra, pelo amor e ensinamentos a mim dedicados.

Aos meus familiares pelos momentos de solidariedade durante as situações difíceis.

Aos meus amigos Anderson, Esdras, Gonçalo, Aline, Allan e Gabriel que me incentivaram e ajudaram em vários momentos importantes.

À minha namorada Daniela, que sempre me apoiou e esteve presente quando eu mais precisava.

Aos meus companheiros de equipe da Acceptor Aerodesing, que considero como uma segunda família.

Ao meu orientador, Brito Filho, por ter depositado confiança e pela oportunidade de realizar este trabalho.

“O insucesso é apenas uma oportunidade para
recomeçar de novo com mais inteligência.”

(Henry Ford)

RESUMO

O Glaucoma é uma doença causada pela lesão do nervo óptico relacionada à alta pressão ocular, podendo acarretar uma redução da visão periférica ou até mesmo a perda súbita da visão. Nesse trabalho será apresentado o desenvolvimento e treinamento de uma Rede Neural Artificial (RNA) que tem como finalidade detectar o Glaucoma, utilizando processamentos de imagens digitais, de modo que seja possível identificar e diagnosticar essa doença nos seus estágios iniciais. Primeiramente, será abordado um pouco sobre as RNA's enfatizando os principais tipos de redes, suas topologias e os tipos de aprendizado, em seguida será denotado todo o processo de aquisição de dados (por meio de um processamento de imagens intraoculares), criação, treinamento e obtenção dos pesos sinápticos da RNA feito com auxílio computacional. Por fim, com os parâmetros da rede definidos, será feita a implementação em FPGA utilizando a linguagem de descrição de *hardware* VHDL, de modo a se reproduzir uma característica inerente das redes neurais biológicas: a execução de operações de adição e multiplicação em paralelo.

Palavras-Chave: Glaucoma, Redes Neurais Artificiais, Processamento de Imagem Digital e FPGA.

ABSTRACT

Glaucoma is a disease caused by optic nerve damage related to high eye pressure, which may cause a reduction in peripheral vision or even sudden loss of vision. In this work will be presented the development and training of an Artificial Neural Network (ANN), which aims to detect glaucoma using digital image processing, so that you can identify and diagnose the disease in its early stages. First, some of the ANN's will be discussed highlighting the main types of networks , their topologies and types of learning, then the entire data acquisition process will be shown (through a processing intraocular images) , setting , training and obtaining the ANN synaptic weights made with computer aid. Finally, with the parameters set of networks, the implementation will be made using the FPGA hardware description language VHDL, to reproduce an inherent characteristic of the biological neural network: executing addition and multiplication operations in parallel.

Keywords: Glaucoma, Artificial Neural Networks, Digital Image Processing and FPGA.

LISTA DE ABREVIATURAS

CLB – *Configuration Logical Blocks*

FPGA – *Field Programmable Gate Array*

HDL – *Hardware Description Language*

IEEE – *Institute of Electrical and Electronics Engineers*

IOB – *In/Out Block*

IOC – Instituto de Olhos Continente

IORJ – Instituto de Oftalmologia do Rio de Janeiro

JTAG – *Joint Test Action Group*

MATLAB – *Matrix Laboratory*

MLP – *Multilayer Perceptron*

NNTOOL – *Neural Network Toolbox*

PID – Processamento de Imagem Digital

PIXEL – *Picture Element*

RNA – Rede Neural Artificial

UFERSA – Universidade Federal Rural do Semiárido

USB – *Universal Serial Bus*

VHDL – *VHSIC Hardware Description Language*

VHDSIC – *Very High Speed Integrated Circuits*

LISTA DE TABELAS

Tabela 1 – Bibliotecas utilizadas.	51
Tabela 2 – Verificação da rede <i>Perceptron</i> sintetizada.	59

LISTA DE FIGURAS

Figura 1 – Evolução no custo de FPGAs comparado com os custos de portas lógicas.....	16
Figura 2 – Representação do Nervo Óptico de um olho: a) Normal; b) Com Glaucoma Moderado e c) Com Glaucoma Avançado	17
Figura 3 – Simulação da perda de visão de pacientes que possuem o Glaucoma, (a) visão normal e (b) visão com Glaucoma.....	18
Figura 4 – Exemplo de diferentes morfologias do neurônio, onde os mesmos possuem: soma, axônio, dendritos e terminais sinápticos.....	20
Figura 5 – Conexão sináptica entre neurônios, constatando a terminação pré-sináptica do axônio com a membrana pós-sináptica do neurônio seguinte.	21
Figura 6 – Análise do funcionamento de um neurônio de seis entrada e uma saída.	22
Figura 7 – Modelo do neurônio de McCulloch e Pitts.....	23
Figura 8 – Gráfico da função linear.....	24
Figura 9 – Gráfico da função degrau.	25
Figura 10 – Gráfico da função Sigmóide.....	25
Figura 11 – Exemplo de rede <i>feedforward</i> de camada única.	26
Figura 12 – Exemplo de rede <i>feedforward</i> de múltiplas camadas.....	27
Figura 13 – Exemplo de rede recorrente.	27
Figura 14 – Representação gráfica de um separador linear para <i>Perceptron</i> de duas entradas.	29
Figura 15 – Ilustração de uma rede <i>Adaline</i>	31
Figura 16 – Rede Hopfield.	33
Figura 17 – Matriz 10 x 10 representando uma imagem ampliada, com 256 tons de cinza. ...	35
Figura 18 – Passos fundamentais para PID.	36
Figura 19 – Arquitetura interna do FPGA.	38
Figura 20 – Diagramas de blocos da metodologia.	40
Figura 21 – Amostra do exame de Fundoscopia: a) OLHO1; b) OLHO2; c) OLHO3; d) OLHO4; e) OLHO5; f) OLHO6.	41
Figura 22 – Esquema do processo de preparação de dados, utilizando a imagem de um exame de Fundoscopia de um olho que possui o Glaucoma.....	42
Figura 23 – Interface gráfica do <i>NNTool</i> com as entradas e saídas.....	44
Figura 24 – Janela para criar a RNA e definição dos seus parâmetros.	45

Figura 25 – Diagrama da rede.	46
Figura 26 – Configuração dos dados para treinamento da RNA.	47
Figura 27 – Simulação da rede.	48
Figura 28 – <i>Kit A – C8V4</i>	49
Figura 29 – Entidade da rede <i>Perceptron</i>	50
Figura 30 – Arquitetura da rede <i>Perceptron</i>	51
Figura 31 – Processamento de imagem realizado com o OLHO1.	52
Figura 32 – Processamento de imagem realizado com o OLHO2.	52
Figura 33 – Processamento de imagem realizado com o OLHO3.	52
Figura 34 – Processamento de imagem realizado com o OLHO4.	53
Figura 35 – Processamento de imagem realizado com o OLHO5.	53
Figura 36 – Processamento de imagem realizado com o OLHO6.	53
Figura 37 – Pesos sinápticos obtidos após o treinamento.	54
Figura 38 – Biais obtido após o treinamento.	54
Figura 39 – Circuito equivalente à descrição da rede <i>Perceptron</i>	55
Figura 40 – Interface do <i>software</i> Quartus II, exibindo o número de componentes internos do FPGA utilizados para o projeto.	56
Figura 41 – OLHO2.	57
Figura 42 – OLHO4.	57
Figura 43 – OLHO5.	58
Figura 44 – OLHO6.	58
Figura 45 – Circuito da rede <i>Perceptron</i> implementado no <i>kit A – C8V4</i> , resultando a saída y1 , correspondente as amostras OLHO2 e OLHO5.	60
Figura 46 – Circuito da rede <i>Perceptron</i> implementado no <i>kit A – C8V4</i> , resultando a saída y1 , correspondente as amostras OLHO4 e OLHO6.	60

SUMÁRIO

1.	INTRODUÇÃO.....	14
1.1.	MOTIVAÇÃO.....	14
1.2.	OBJETIVOS	16
1.2.1.	Objetivo Geral.....	16
1.2.2.	Objetivo Específico	16
2.	FUNDAMENTAÇÃO TEÓRICA	17
2.1.	GLAUCOMA	17
2.2.	REDES NEURAIS ARTIFICIAIS	19
2.2.1.	Neurônio Biológico.....	20
2.2.2.	Neurônio Artificial.....	22
2.2.3.	Tipos de Arquiteturas em Redes Neurais	26
2.2.4.	Tipos de Aprendizagem em Redes Neurais	28
2.2.5.	Rede <i>Perceptron</i>	28
2.2.6.	Rede <i>Adaline</i>	31
2.2.7.	Redes Recorrentes de Hopfield.....	33
2.3.	PROCESSAMENTO DE IMAGEM DIGITAL.....	34
2.3.1.	Definição de Imagem Digital.....	35
2.3.2.	Passos essenciais para PID	35
2.4.	FPGA	38
3.	METODOLOGIA	40
3.1.	AQUISIÇÃO E PROCESSAMENTO DAS IMAGENS NO MATLAB.....	40
3.2.	ESCOLHA E CRIAÇÃO DA REDE NEURAL ARTIFICIAL.....	43
3.3.	IMPLEMENTAÇÃO EM FPGA.....	49
4.	RESULTADOS	52
4.1.	PROCESSO DE CONSTRUÇÃO DE DADOS.....	52
4.2.	PROCESSO DE OBTENÇÃO DOS PARÂMETROS DA RNA	54
4.3.	IMPLEMENTAÇÃO.....	54
5.	CONCLUSÃO	61
5.1.	TRABALHOS FUTUROS	62
6.	REFERÊNCIAS.....	63

1. INTRODUÇÃO

Segundo o Instituto de Olhos Continente (2016), o Glaucoma é uma doença do nervo óptico que ocorre, geralmente, pelo aumento da pressão intraocular, dificultando o transporte de oxigênio e nutrientes dentro dos neurônios que formam o nervo, ocasionando a compressão dos vasos sanguíneos que nutrem as estruturas no fundo do olho. A partir dessa compressão inicia-se o processo de degeneração do nervo óptico e dos neurônios que compõem a retina, acarretando uma perda progressiva e irreversível da visão.

Os sintomas dessa doença são raros, de tal forma que a doença se desenvolve lentamente e silenciosamente, no decorrer de meses ou anos, fazendo com que paciente vá em busca de auxílio apenas quando a doença se encontra em estágios já avançados.

Diante disso é necessário a criação de um método para obtenção de um diagnóstico veramente eficiente. Pensando nisso o presente trabalho busca esse método a partir das Redes Neurais Artificiais (também conhecidas como RNAs), onde as mesmas processam imagens intraoculares de olhos de pacientes saudáveis e com glaucoma, utilizando processamento de imagens digitais de modo que a RNA utilizada, consiga – por meio de um treinamento específico – diferenciar as duas classes. Tal treinamento é realizado de forma *off-line* com o auxílio da ferramenta *Neural Network Toolbox* do *software* MATLAB, sigla para *Matrix Laboratory*.

O circuito a ser desenvolvido neste trabalho será descrito utilizando linguagem de descrição de *hardware*, permitindo o seu desenvolvimento de forma programável. Tal circuito é implementado em FPGA, sigla para *Field Programmable Gate Array*, que é uma categoria de *hardware* configurável, onde suas funcionalidades são definidas diretamente pelo usuário, e não pelo fabricante. Após sua implementação será observado o comportamento da RNA quando aplicadas determinadas amostras como entrada da rede.

1.1. MOTIVAÇÃO

Segundo Mendonça (2013), no Brasil, a principal causa de cegueira irreversível é o Glaucoma, atingindo quase 1 milhão de pessoas, de modo que 80% dos portadores do glaucoma só buscam o oftalmologista após perceber alguma mudança de campo visual (vista borrada, perda de visão, olhos vermelhos e desconforto).

Diante desse fato, é visível a falta de métodos de identificação e diagnóstico do Glaucoma no seu estado inicial, com isso inicia-se uma busca por métodos mais eficientes, de modo que podemos destacar a aplicação das redes neurais artificiais.

A utilização de redes neurais artificiais vem se mostrando como uma ótima opção para aplicações que necessitam de reconhecimento de padrões, já que os métodos atuais não suprem a necessidade da sociedade, que busca formas de detecção e diagnósticos prévios de doenças.

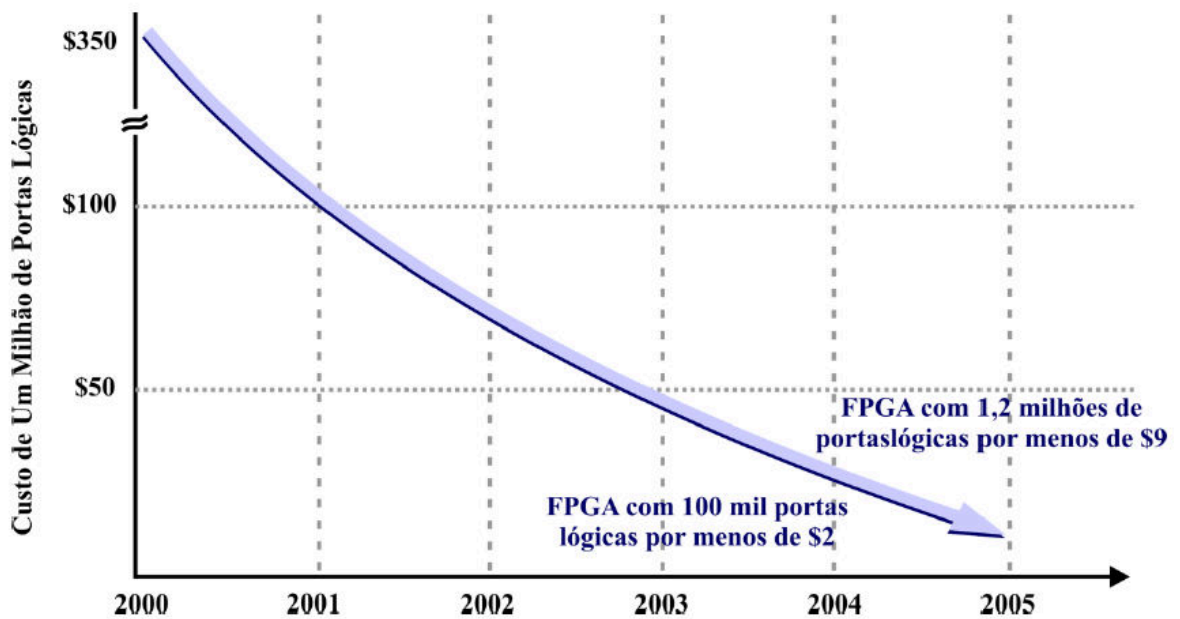
As principais vantagens da utilização de RNAs são:

- Enorme capacidade de processamento;
- Execução de operações em paralelo;
- Predisposição de aprender com seus próprios erros.

Como a maioria dos *softwares* de redes neurais é executada por arquiteturas sequenciais (CORIC, 2000), não é possível efetuar múltiplas operações em paralelo, que são características de RNAs, logo não se pode obter o máximo de rendimento da rede. Para solucionar esse problema é necessário implementá-la em *hardwares* de unidades capazes de realizar operações em paralelo e de forma rápida. Logo após o ressurgimento das pesquisas sobre redes neurais artificiais em meados dos anos 80, percebeu-se que para o total proveito das capacidades de aprendizado e paralelismo massivo, destas, é essencial a implementação em *hardware* (HARIPRASATH, 2012).

O FPGA é uma excelente opção para a implementação de RNAs, pois esse dispositivo possui um arranjo eletrônico interno que lhe permite ser reconfigurável e ter uma alta capacidade de processamento em paralelo. Segundo Pião (2012), existem diversas alternativas de configuração deste dispositivo oferecidas por diferentes *softwares*, como o Quartus II da Altera, facilitando o desenvolvimento de projetos. Outra forma de constatar a vantagem da utilização do FPGA é quando ao seu desenvolvimento é aplicada a Lei de Moore, o que permite observar os avanços tecnológicos desse dispositivo em termos de potência, desempenho e preço, como mostra a Figura 1.

Figura 1 – Evolução no custo de FPGAs comparado com os custos de portas lógicas.



Fonte: Pião, 2012.

1.2. OBJETIVOS

Neste tópico serão abordados os objetivos, geral e específico, relacionados ao trabalho proposto.

1.2.1. Objetivo Geral

Desenvolver e aplicar um modelo de uma Rede Neural Artificial, capaz de identificar o Glaucoma no seu estágio inicial, por meio de reconhecimento padrão de imagens digitais implementado em FPGA.

1.2.2. Objetivo Específico

- Realizar estudos sobre Redes Neurais Artificiais e o Glaucoma;
- Desenvolver métodos para processamentos de imagens no MATLAB;
- Desenvolver e aplicar um treinamento específico no MATLAB para a RNA;
- Criar um algoritmo em VHDL para implementação da rede em FPGA;

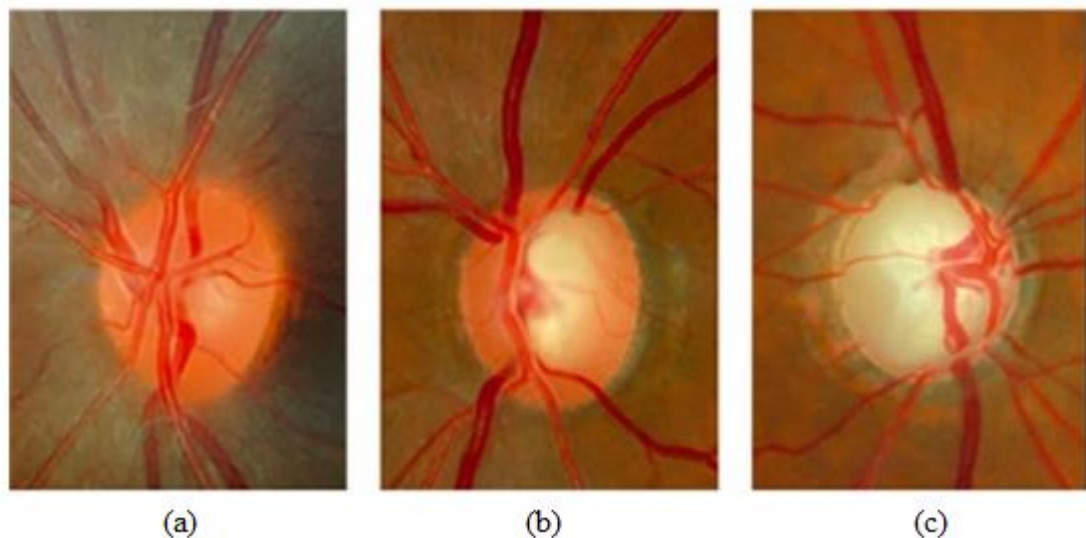
2. FUNDAMENTAÇÃO TEÓRICA

2.1. GLAUCOMA

O Glaucoma é uma neuropatia óptica de causa multifatorial, caracterizada pela lesão progressiva do nervo óptico, com consequente repercussão no campo visual (URBANO *et al.*, 2003). Segundo o Instituto Benjamin Constant (2015), o estado crônico dessa doença é caracterizado pela perda da visão periférica (visão que permite perceber objetos ao nosso redor), ocasionado pela lesão das fibras existentes no nervos que originam a retina e as que formam o nervo óptico. Quando o estado é agudo, ocorre uma perda extremamente grave ou súbita da visão, devido ao aumento extremo da pressão interna do olho.

Segundo Póvoa *et al.* (2001), o Glaucoma é a segunda causa de cegueira no mundo, de tal modo que a estimativa do número de indivíduos com Glaucoma em todo o mundo para o ano 2000 foi de 66,8 milhões de pessoas, com cerca de 8 milhões apresentando cegueira bilateral devido à lesão irreversível do nervo óptico. As diferentes formas de Glaucoma constituem a principal causa de cegueira irreversível nos países industrializados e em desenvolvimento (URBANO *et al.*, 2003).

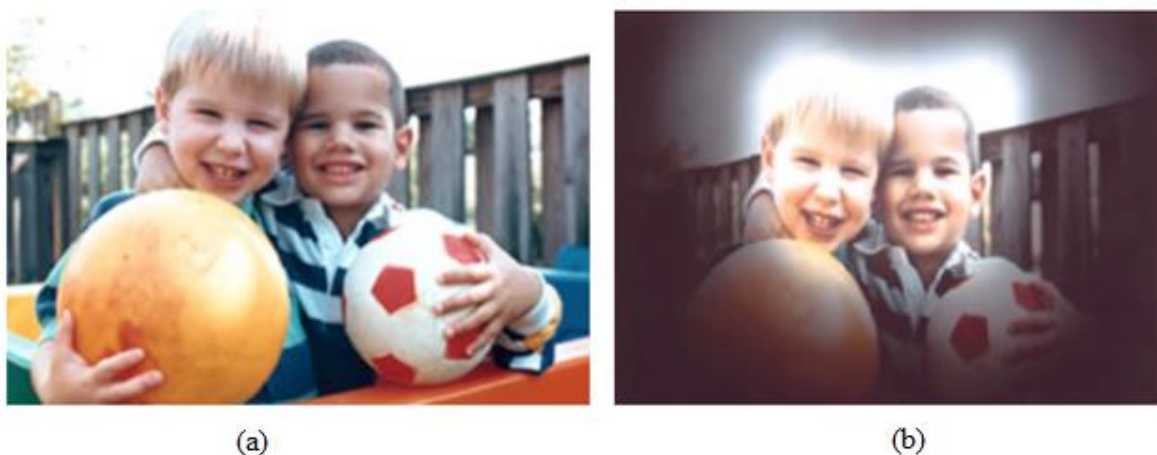
Figura 2 – Representação do Nervo Óptico de um olho: a) Normal; b) Com Glaucoma Moderado e c) Com Glaucoma Avançado



Fonte: <http://cvi.med.br/?dica=como-cuidar-da-sua-lentes-contato>.

Segundo o IORJ (2016) o Glaucoma raramente demonstra sintomas, entretanto existe alguns sinais típicos que são ocasionados pela doença. Na sua forma aguda são evidenciados fortes dores de cabeça, dor ocular, sensibilidade a luz e enjoo, já na sua forma crônica a doença se manifesta pela diminuição da visão periférica, que pode ou não ser percebida pelo paciente dependendo do estágio da doença.

Figura 3 – Simulação da perda de visão de pacientes que possuem o Glaucoma, (a) visão normal e (b) visão com Glaucoma.



Fonte: <http://www.iorj.med.br/glaucoma-2/>.

De acordo com o Instituto Benjamin Constant (2015), atualmente o Glaucoma é diagnosticado por exames, tais como a Tonometria de Aplanação (exame para a medição da pressão intraocular); Fundo de olho ou Fundoscopia (exame para avaliar a existência de alguma lesão no nervo óptico, ocasionado pelo Glaucoma); Gonioscopia (exame para classificar o estado do Glaucoma – agudo ou crônico -) e o Campo Visual (exame para avaliação se já houve alguma perda do campo visual causado pelo Glaucoma). Para evitar a perda de visão irreversível causado pelo Glaucoma é necessário o tratamento ideal, que melhor proporcione o controle dessa, ou seja, ausência de progressão. Esse tratamento deve ser feito com colírios (que diminuam a pressão interna do olho), laser e cirurgias (MELLO, 2013).

Caso o Glaucoma não seja tratado, o campo visual diminui progressivamente, ocasionando um obscurecimento da visão central de modo a progredir até a cegueira total. Logo o paciente não deve aguardar que os sintomas sejam evidentes, pois caso o paciente já possua uma perda de visão relacionada com o Glaucoma, ela se torna irreversível. Diante

disso é necessário a prevenção adequada que possa monitorar, periodicamente e de maneira eficiente, o estado do nervo óptico e o de visão do paciente (BONFADINI, 2013).

2.2. REDES NEURAIS ARTIFICIAS

De acordo com Silva (2010), as Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados no sistema nervoso de seres vivos, dos quais possuem a capacidade de aquisição e manutenção de conhecimento (baseado em informações) e podem ser definidas como unidades de processamentos, caracterizadas por neurônios artificiais.

Também conhecidos como nós computacionais, os neurônios de uma RNA, são interligados por um grande número de interconexões denominado sinapses artificiais, sendo as mesmas representadas por vetores ou matrizes de pesos sinápticos.

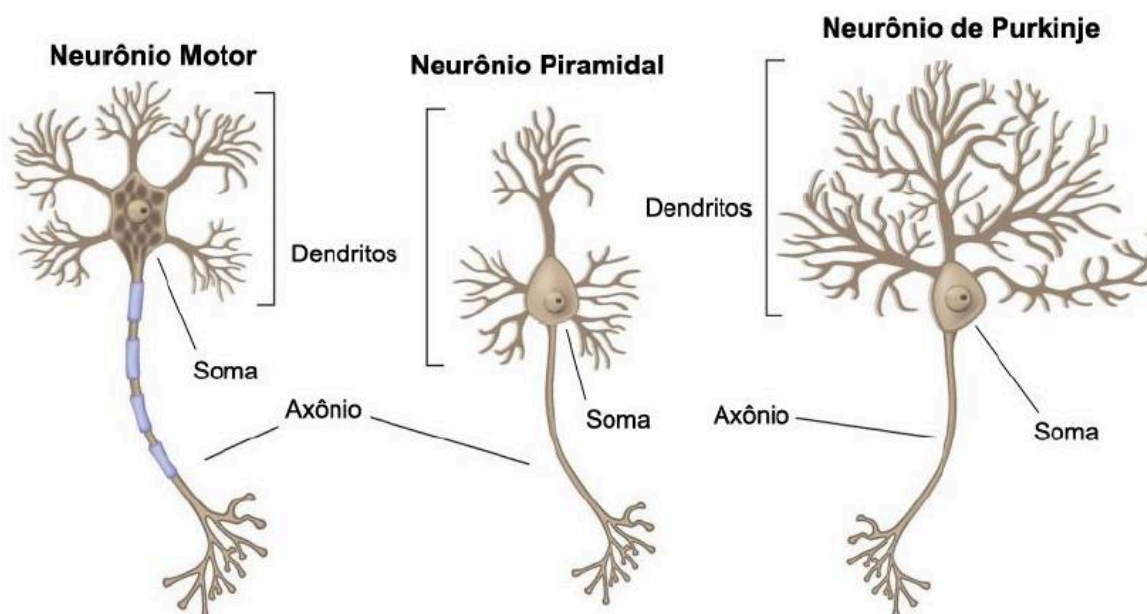
As características mais relevantes envolvendo aplicações com Redes Neurais Artificiais são (SILVA, 2010):

- Adaptação por experiência: as adaptações dos parâmetros internos da rede, conhecido como pesos sinápticos e bias, são ajustados a partir da apresentação sucessiva de exemplos relacionados ao comportamento do processo, possibilitando a aquisição do conhecimento por experiência;
- Capacidade de aprendizado: essa etapa corre por intermédio de uma aplicação de algum método de treinamento, do qual a rede consegue extrair a relação existente entre as variáveis de entrada da rede;
- Habilidade de generalização: após o treinamento finalizado, a RNA é capaz de generalizar o conhecimento adquirido, de tal forma que a rede pode estimar uma solução que até então era desconhecida;
- Organização de dados: a rede é capaz de realizar uma organização interna, visando a possibilidade de agrupar padrões que apresentem particularidades em comum.

2.2.1. Neurônio Biológico

O neurônio biológico é dividido em quatro seções: o soma (corpo celular), o axônio, os dendritos e os terminais sinápticos, onde a primeira descrição detalhada foi fruto das observações do neurologista espanhol Ramón y Cajal, no século 19.

Figura 4 – Exemplo de diferentes morfologias do neurônio, onde os mesmos possuem: soma, axônio, dendritos e terminais sinápticos.



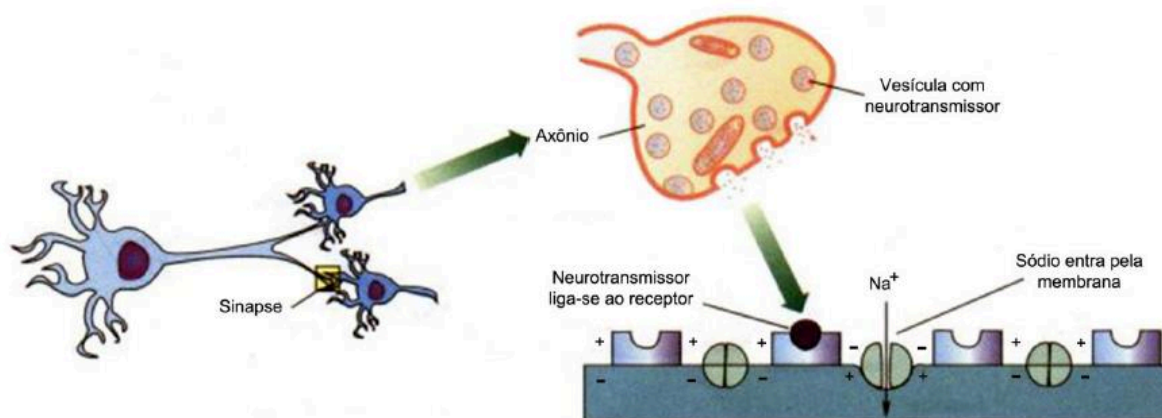
Fonte: Pião, 2012.

Com várias ramificação maiores que o próprio corpo celular, os dendritos são responsáveis por receber impulsos nervosos (informações) de outros neurônios e conduzi-las até o soma, que é parte central do neurônio, onde estão contidos o núcleo e o citoplasma. Aqui a informação é processada e novos impulsos são gerados. Esses impulsos são transmitidos por meio de um canal denominado axônio até os dendritos dos neurônios seguintes, entretanto, segundo Kovács (2006), essa transmissão de impulsos não são resultado de uma conexão física direta entre neurônios, mas sim, uma conexão sináptica.

O axônio possui terminações, que são constituídas por uma membrana denominada pré-sináptica, que converte o impulso elétrico proveniente do soma, na emissão de neurotransmissores anteriormente armazenados nas vesículas (pequenas bolsas membranosas e esféricas localizadas nos extremos dos axônios). Esses neurotransmissores ligam-se aos receptores da membrana pós-sináptica, presente nos dendritos do neurônio seguinte,

propiciando a absorção de íons de sódio, ocasionando uma inversão de cargas elétricas e a condição de um novo pulso elétrico no neurônio estimulado, caracterizando um caso de conexão excitatória, como mostra a figura 5.

Figura 5 – Conexão sináptica entre neurônios, constatando a terminação pré-sináptica do axônio com a membrana pós-sináptica do neurônio seguinte.



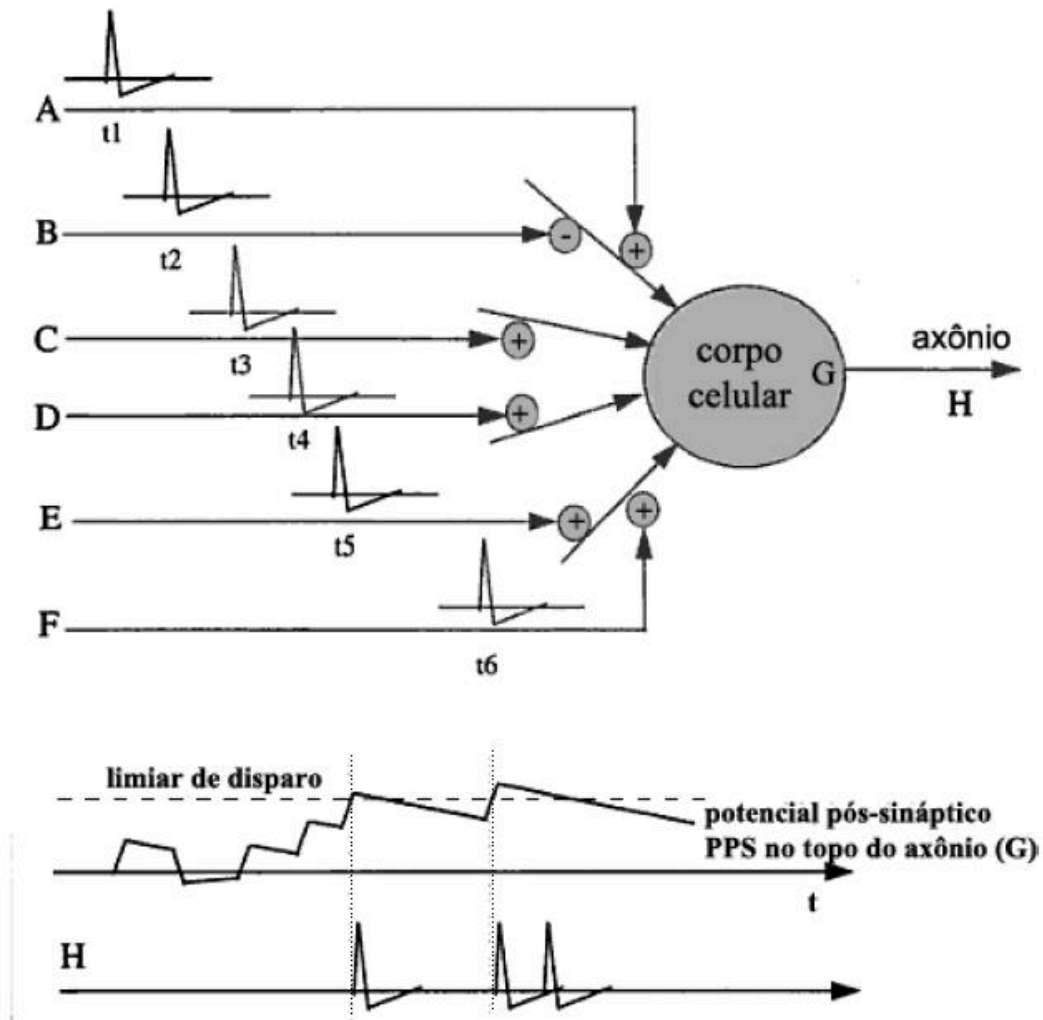
Fonte: <http://www.afh.bio.br/nervoso/nervoso2.asp>

Segundo Pião (2012) uma conexão excitatória produz uma alteração no potencial elétrico da membrana, ocasionando a transmissão de impulso elétrico, já uma conexão inibitória tem o efeito oposto. Esse comportamento depende diretamente dos neurotransmissores envolvidos. A formação de potencial nas terminações do axônio do segundo neurônio só é possível caso ocorra um acúmulo suficiente de neurotransmissores no primeiro, de modo que ocorra uma despolarização da membrana pós-sináptica, suficiente para atingir um valor, denominado limiar de disparo. Logo de acordo com Kovács (2006), o funcionamento do neurônio está ligado diretamente com a ponderação da intensidade das ligações sinápticas estabelecidas.

Como exemplo a figura 6, denota um modelo de neurônio com seis entradas (A, B, C, D, E e F) e uma saída H. Os pulsos de entradas não ocorrem simultaneamente, entretanto devido a duração de cada pulso ser da ordem de milissegundos até alguns segundos e a degradação de cada pulso não ser instantânea (efeito conhecido como integração temporal de estímulos), é possível, após o pulso excitante t_5 ultrapassar o limiar de disparo, ocasionar uma saída nesse instante em H. O mesmo evento ocorre no instante t_6 , onde devido ao

comportamento de restabelecimento do neurônio, é possível obter dois novos pulsos na saída H.

Figura 6 – Análise do funcionamento de um neurônio de seis entrada e uma saída.



Fonte: Kovács, 1996, página 19.

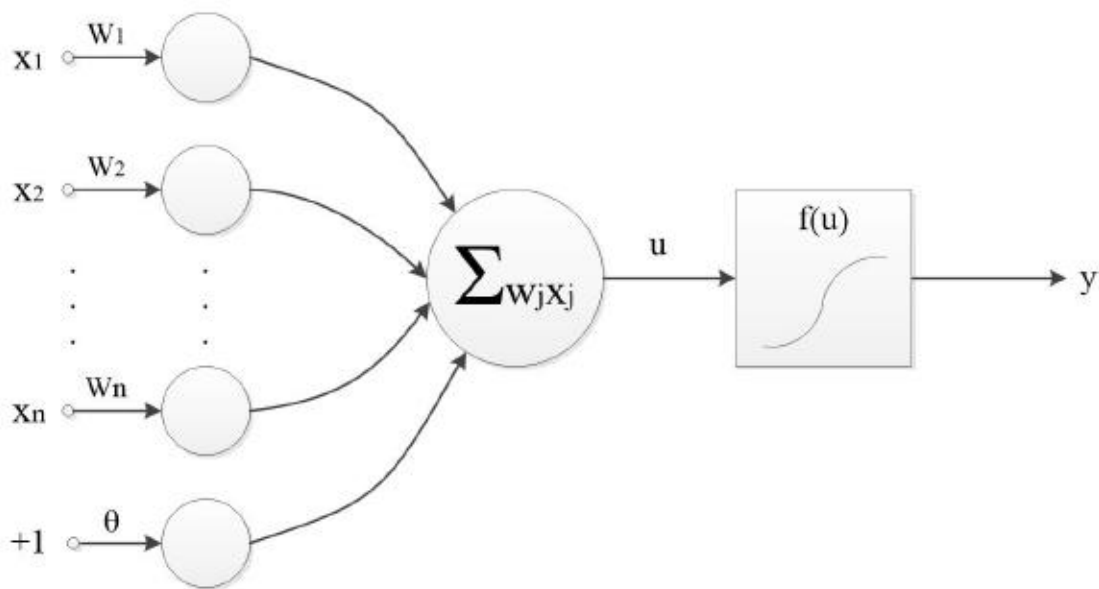
2.2.2. Neurônio Artificial

O neurônio artificial proposto inicial por McCulloch e Pitts, em 1943, apesar de simples, engloba todas as características de uma rede neural biológica, e é ainda o modelo mais usado para criar arquiteturas de RNAs (SILVA, 2010). Esse tipo de neurônio possui características específicas, são elas:

- Entradas de $X_1 \dots X_n$, representam os pulsos recebidos pela entrada da rede ou por outros neurônios

- Os pesos sinápticos $W_1 \dots W_n$, representam a sensibilidade da membrana do neurônio
- A soma do produto das entradas X_j com seu respectivo peso W_j , é efetuada pelo bloco de soma aditiva $\sum_{j=1}^n W_j X_j$
- O parâmetro θ representa o limiar de disparo de um neurônio, e a subtração deste com a função aditiva representa a comparação do potencial pós-sináptico com o limiar de disparo
- A saída u , representa que o limiar foi ultrapassado pela função aditiva, logo um sinal é emitido.
- A função $f(u)$ representa a concentração de neurotransmissores na membrana pré-sináptica de onde será implementada uma função de ativação
- A saída y representa o valor de saída do neurônio

Figura 7 – Modelo do neurônio de McCulloch e Pitts.



Fonte: Holanda, 2013.

Esse modelo ainda pode ser representado na sua forma algébrica pelas equações (1) e (2), na qual pode se obter uma saída y do neurônio em função das entradas X_j , dos pesos sinápticos W_j , do bias θ e através de uma função de ativação.

$$u = \sum_{j=1}^n W_j X_j - \theta \quad (1)$$

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

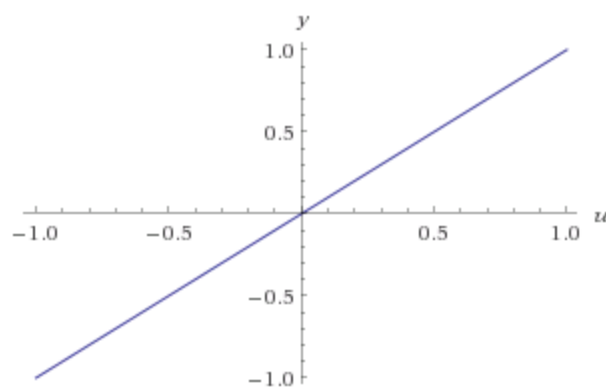
A variação da função de ativação também conhecida como função de transferência, dos parâmetros W_i e θ , e da quantidade de neurônios permite que a rede neural seja capaz de desempenhar funções que exigem um maior grau de complexidade. Essa variação é denominada treinamento.

Segundo Haykin (2001) a função de ativação, representada por $f(u)$, define a saída do neurônio em termos do campo local induzido, ou seja ela é escolhida de acordo com a necessidade do problema em que a rede está trabalhando. Segundo Soares (2013), dentre as funções de ativação, as mais utilizadas são:

- **Função Linear:** Esse tipo de função não limita a saída da rede e é utilizada para armazenamento de dados, de modo que o neurônio que possua essa função atue como um aproximador linear, como mostra a figura 8.

$$f(u) = u \quad (3)$$

Figura 8 – Gráfico da função linear.

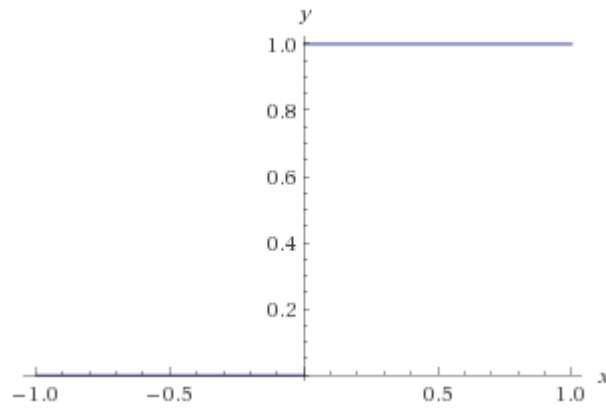


Fonte: <https://www.wolframalpha.com/>.

- **Função Limiar ou Degrau:** Nesse tipo de função o neurônio vai gerar uma saída com valor 1 quando o resultado da função aditiva for maior ou igual a zero, como mostra a figura 9.

$$f(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases} \quad (4)$$

Figura 9 – Gráfico da função degrau.

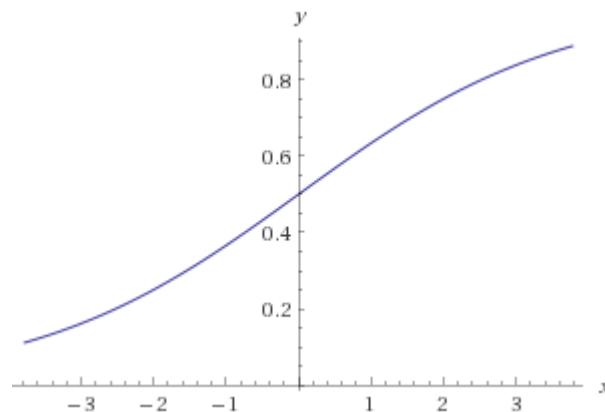


Fonte: <https://www.wolframalpha.com/>.

- **Função Sigmóide:** Segundo Braga *et al.* (2007) esse tipo de função é a mais utilizada na construção de redes neurais. Isso ocorre devido sua característica de ser uma função estritamente crescente, exibindo um comportamento balanceado linear e não-linear, como mostra a figura 10.

$$f(u) = \frac{1}{(1 + e^{-u})} \quad (5)$$

Figura 10 – Gráfico da função Sigmóide.



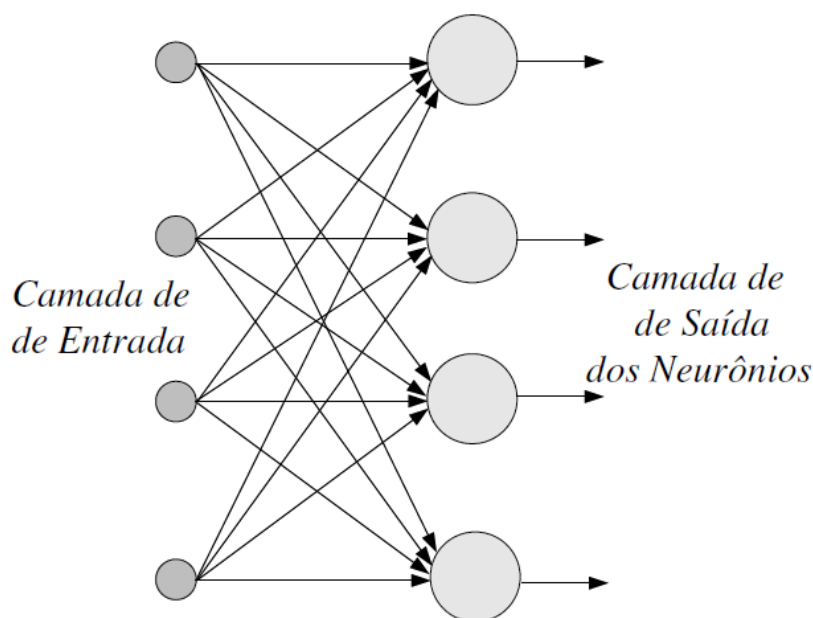
Fonte: <https://www.wolframalpha.com/>.

2.2.3. Tipos de Arquiteturas em Redes Neurais

Segundo Haykin (2001) o algoritmo de aprendizado utilizado para treinar a rede, está ligada diretamente a maneira pela qual os neurônios estão estruturados. Em geral, podemos identificar três tipos de classes de arquiteturas de redes fundamentalmente diferentes:

- **Redes Alimentadas Adiante com Camada Única:** Conhecida também como rede *feedforward* de camada única, esse tipo de rede neural é alimentada de forma acíclica. Em outras palavras, a camada de entrada se projeta sobre uma camada de saída e não vice-versa. Tal camada de saída é designada como “camada única”, já que nesse tipo de arquitetura essa camada é a única que possui nós computacionais (neurônios), como demonstra a figura 11.

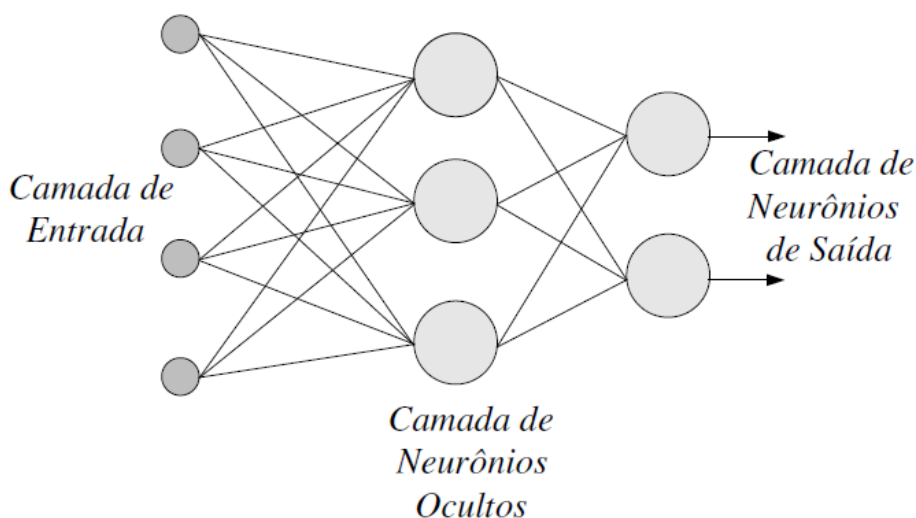
Figura 11 – Exemplo de rede *feedforward* de camada única.



Fonte: Matsunaga, 2012, página 7.

- **Redes Alimentadas Diretamente com Múltiplas Camadas:** Esse tipo de rede, também é conhecida como *feedforward* de múltiplas camadas, é alimentada de forma acíclica e possui várias camadas ocultas – camada de neurônio que se encontra entre a entrada e a saída da rede -. Esta camada desempenha um papel de intervir, de maneira útil, entre a camada de entrada e a camada de saída da rede, tornando-a capaz de extrair estatística de ordem mais elevada, como mostra a figura 12.

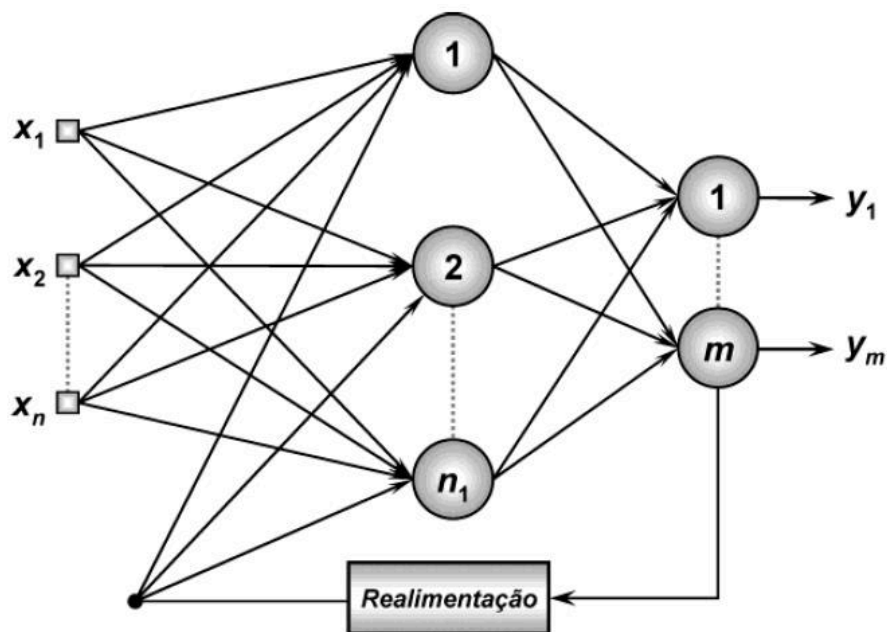
Figura 12 – Exemplo de rede *feedforward* de múltiplas camadas.



Fonte: Matsunaga, 2012, página 8.

- **Rede Recorrentes:** Esse tipo de rede, também conhecida como *feedback*, possui pelo menos uma camada de nós computacionais, porém se distingue das demais, pelo fato de ter os laços de realimentação. Segundo Haykin (2001) a presença desses laços tem um impacto profundo na capacidade de aprendizagem da rede e no seu desempenho, como indica a figura 13.

Figura 13 – Exemplo de rede recorrente.



Fonte: Silva, 2010, página 49.

2.2.4. Tipos de Aprendizagem em Redes Neurais

Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural artificial são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (HAYKIN, 2001). De modo geral, podemos citar como os principais tipos de aprendizagem:

- **Aprendizagem Supervisionada:** Conhecida também como aprendizagem com um professor. Para aplicação desse tipo de aprendizagem é necessário um mentor, pessoa ou professor, que seja capaz de fornecer à rede neural uma resposta desejada para um vetor de treinamento inicial.
- **Aprendizagem Não-Supervisionada:** Conhecida também como aprendizagem auto organizada. Nesse tipo de aprendizagem não há um mentor para supervisionar o processo de aprendizado. Neste caso são fornecidas condições para a realização uma medida independente da tarefa da qualidade da representação que a rede deve aprender, de modo que os parâmetros livres da rede sejam otimizados em relação a esta medida.

2.2.5. Rede *Perceptron*

A primeira rede neural implementada foi a do tipo *Perceptron*, idealizada por Rosenblatt em 1958 e baseada no modelo criado por McCulloch-Pitts. Segundo Faceli *et al.* (2011), embora a arquitetura desse tipo de rede seja simples, apresentando apenas uma camada de neurônios, o *Perceptron* apresentou uma boa acurácia preditiva em diversos problemas de classificação linear e não-linear.

Segundos Kovács (2006) a proposta de Rosenblatt era encontrar um método que permitisse a determinação dos parâmetros da rede, para isso ele baseou-se em sistemas biológicos de tal forma que o mais intuitivo método de treinamento de uma rede neural consiste no treinamento através de exemplos, em que são apresentadas entradas associadas as suas respectivas saídas, até que a rede obtenha o comportamento desejado.

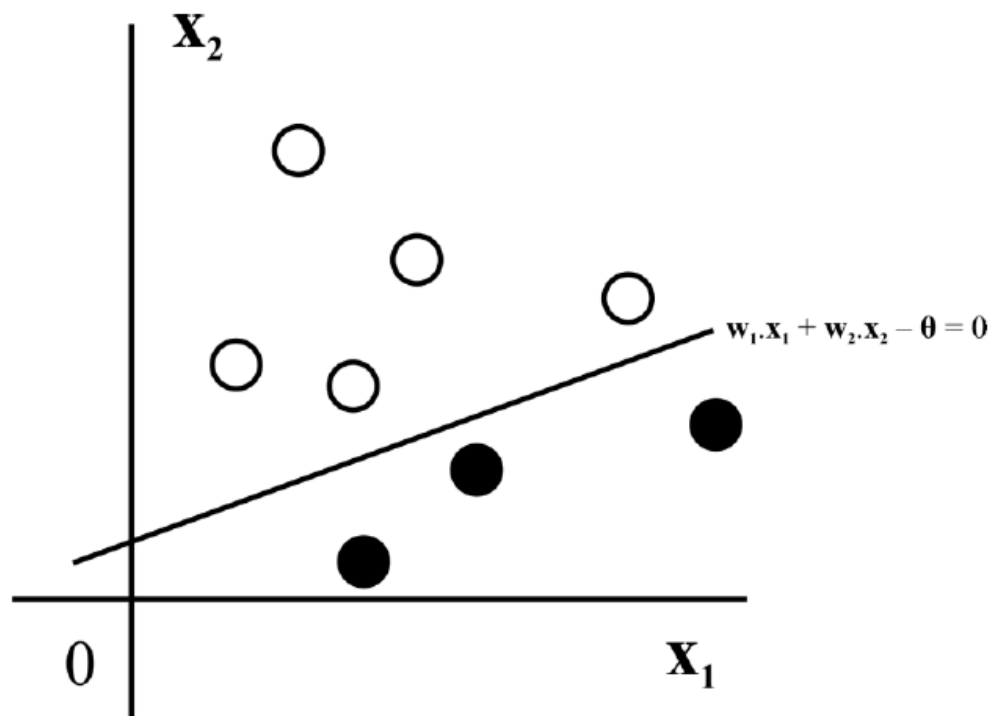
A rede *Perceptron* de camada simples pode ser descrita matematicamente por $y = f(u)$, onde $f(u)$ pode ser uma função degrau ou sinal, e u é definido pela função (3) (SILVA, 2010).

$$u = \sum_{i=1}^n W_i X_i - \theta \quad (6)$$

Portanto, uma rede *Perceptron* de camada simples com duas entradas pode classificar valores separáveis pela reta descrita pela função (4), e ilustrada na figura (14) (SILVA, 2010).

$$X_1 W_1 + X_2 W_2 - \theta = 0 \quad (7)$$

Figura 14 – Representação gráfica de um separador linear para *Perceptron* de duas entradas.



Fonte: Kovács, 2006.

Observando as características da rede *Perceptron* de camada simples é notório concluir que esta é somente capaz de obter uma classificação exata de amostras que são linearmente separáveis, independentemente do número de entradas, ou seja, esse tipo de rede não

consegue classificar valores não-lineares. O separador linear pode, também, variar de acordo com os parâmetros iniciais adotados para o treinamento da rede (SILVA, 2010,).

O treinamento desse tipo de rede baseia-se no melhor posicionamento do separador linear, de modo que se obtenha uma fronteira dividindo as duas classes em questão utilizando um algoritmo supervisionado de correção de erro. Segundo Kovács (2006) tal algoritmo pode ser representado pelas funções (5) e (6) para determinação dos parâmetros W_n e θ .

$$W_{n,l+1} = W_{n,l} + \Delta W_{n,l} \quad (8)$$

$$\theta_{l+1} = \theta_l + \Delta \theta_l \quad (9)$$

Sendo que:

$$\Delta W_{n,l} = \eta \cdot (y_l^d - y_l) \cdot X_{n,l} \quad (10)$$

$$\Delta \theta_l = \eta \cdot (y_l^d - y_l) \quad (11)$$

Segundo Kovács (2006) e Silva (2011) o parâmetro η se refere a taxa de aprendizado, normalmente compreendido em $0 < \eta < 1$, esse parâmetro deve ser escolhido adequadamente para que possa permitir uma rápida convergência dos parâmetros da rede sem que haja instabilidades no processo de treinamento. A saída desejada é representada por y_l^d e y_l se refere a saída obtida para uma entrada $X_{n,l}$, onde l representa a l -ésima amostra de teste de treinamento da rede.

Para soluções de casos não linearmente separáveis, pode-se utilizar uma derivação do *Perceptron* de camada simples, denominado *Perceptron* de Múltiplas Camadas (MLP), porém o processo de treinamento dessa rede se torna bastante complexo, isso ocorre devido a sua dificuldade em estabelecer corretamente os parâmetros das camadas ocultas – ou intermediárias –. Uma opção para treinamento desse tipo de rede é o algoritmo *backpropagation*.

2.2.6. Rede Adaline

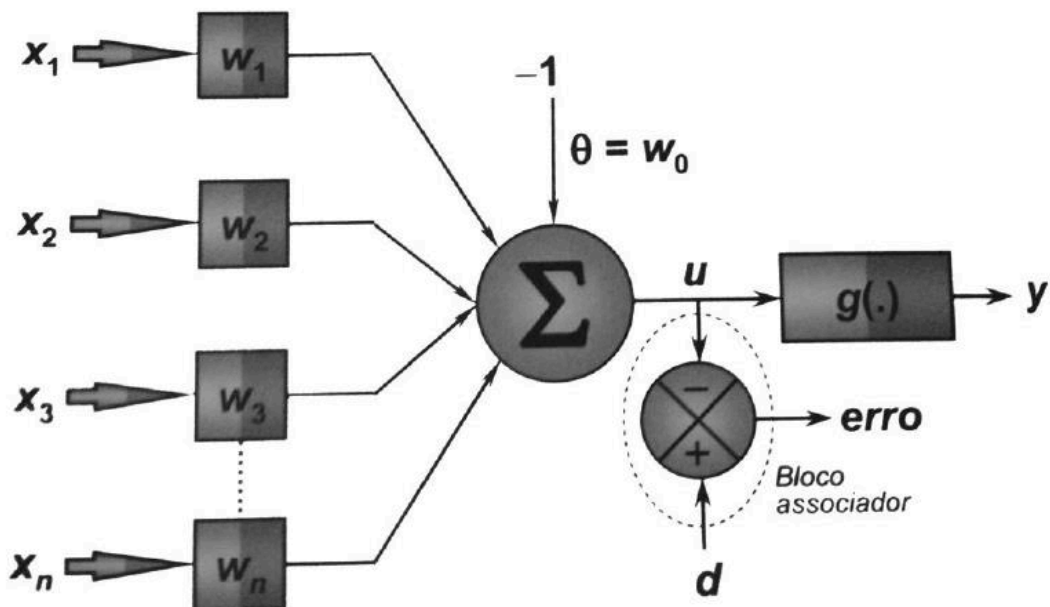
Segundo Silva (2010) a rede *Adaline* foi idealizada por Widrow e Hoff em 1960. Esse tipo de rede tornou-se uma das primeiras redes neurais artificiais a serem aplicadas nas industriais, onde sua principal aplicação foi em sistemas de chaveamento de circuitos telefônicos. Sua grande contribuição foi a introdução do algoritmo de aprendizado denominado regra Delta, pois este é considerado o precursor da regra Delta generalizada, que é utilizada para treinamento de redes *Perceptron* de múltiplas camadas (SILVA, 2010)

A rede *Adaline* possui uma configuração estrutural bastante similar a rede *Perceptron* de camada única, de modo que essa rede é constituída por apenas uma camada neural, sendo composta por apenas um neurônio artificial. Segundo Kovács (2006), a composição de várias *Adaline*, constituindo uma rede única, é denominada rede *Madaline*.

Ilustrada na figura 10, a rede, análoga à rede *Perceptron*, possui uma arquitetura *feedforward* com X_n entradas e somente uma saída. Em conformidade com Silva (2010), a representação matemática dessa rede é dada por $y = f(u)$ onde $f(u)$ pode ser uma função degrau ou sinal, e u pode ser estabelecido pela equação (9)..

$$u = \sum_{i=1}^n X_i W_i - \theta \quad (12)$$

Figura 15 – Ilustração de uma rede *Adaline*.



Fonte: Silva, 2010.

Na figura 15 observa-se a presença de um bloco associador, cuja função é simplesmente auxiliar no processo de ajuste dos pesos sinápticos durante o treinamento da rede. A equação (10) que representa o funcionamento do bloco associador.

$$erro = d - u \quad (13)$$

O processo de ajuste dos pesos e do limiar da rede *Adaline* é baseado no algoritmo de aprendizado denominado de regra Delta ou regra de aprendizado de Widrow-Hoff, conhecido também como algoritmo LMS (*least mean square*) ou método do Gradiente Descendente (SILVA, 2010).

A regra Delta baseia-se na minimização do erro quadrático entre a saída obtida $\{u\}$ e a saída desejada $\{d\}$. Em suma, o objetivo dessa regra Delta é obter um W^* ótimo tal que o erro quadrático $\{E(W^*)\}$ sobre todo o conjunto de amostras seja o mínimo possível (SILVA, 2010).

As equações (11) e (12) descrevem como ocorre a atualização dos parâmetros de uma rede *Adaline*.

$$W_n^{atual} = W_n^{anterior} + \Delta W_n \quad (14)$$

$$\Delta W_n = \eta \cdot \sum_{i=1}^n (d^{(k)} - u) \cdot X^{(k)} \quad (15)$$

Onde:

- W_n é um vetor contendo os pesos e o limiar;
- $d^{(k)}$ é o valor desejado para a k -ésima amostra de treinamento;
- $X^{(k)}$ é a k -ésima amostra de treinamento;
- u é o valor de saída do combinador linear;
- η é uma constante que define a taxa de aprendizagem da rede;
- ΔW_n é o ajuste obtido através da minimização do erro quadrático

Diante disso é notório a diferença entre o treinamento da rede *Adaline* e *Perceptron*, porém a análise matemática efetuada para demonstrar as condições de convergência do *Perceptron* pode ser também aplicadas para à rede *Adaline*. Em suma, as classes do problema a ser mapeado devem ser linearmente separáveis para que haja a sai completa identificação (SILVA, 2010).

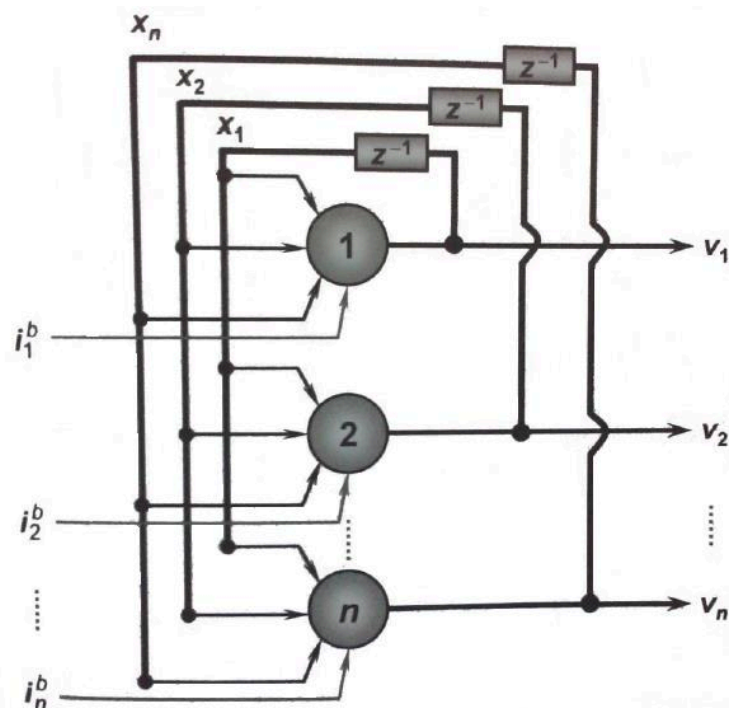
2.2.7. Redes Recorrentes de Hopfield

As redes neurais consideradas recorrentes são aquelas em que as saídas de uma camada neural podem ser realimentadas às suas entradas (SILVA, 2010). A principal rede recorrente foi idealizada por Hopfield em 1982, da qual é denominada rede Hopfield. Tal arquitetura possui as seguintes características:

- Comportamento tipicamente dinâmico;
- Capacidade de memorizar relacionamentos;
- Possibilidade de armazenamento de informações;
- Facilidade de implementação em *hardware* analógico.

A figura 16 ilustra a proposta original para a rede Hopfield, onde o z^{-1} indica o atraso temporal de cada neurônio, X_n e i_n^b indicando respectivamente o n -ésimo neurônio e seu respectivo limiar de disparo. Observa-se que esse tipo de rede possui apenas uma camada de neurônios, e todas as saídas realimentam todas as entradas.

Figura 16 – Rede Hopfield.



Fonte: Silva, 2010.

As equações (13) e (14) expressam o comportamento, em tempo contínuo, de cada neurônio da rede de Hopfield (SILVA, 2010).

$$u_j(t) = -\eta_j u_j(t) + \sum_{i=1}^n W_{ji} \cdot v_i(t) + i_j^b, \text{ onde } j = 1, \dots, n \quad (16)$$

$$v_j(t) = g(u_j(t)) \quad (17)$$

Onde:

- $u_j(t)$ é o estado interno do j -ésimo neurônio, com $u_j(t) = du/dt$;
- $v_i(t)$ é a saída do i -ésimo neurônio;
- W_{ji} é o valor do peso sináptico conectando o j -ésimo neurônio ao i -ésimo neurônio;
- i_j^b é o limiar (bias) aplicado ao j -ésimo neurônio;
- $g(.)$ é uma função de ativação, monótona crescente, que limita a saída de cada neurônio em um intervalo predefinido;
- $\eta_j u_j(t)$ é um termo de decaimento passivo;

Desse modo, a entrada $x(t_0)$ gera uma saída inicial $v(t_0)$, de tal forma que nos próximos passos a rede retorna a saída à entrada até que a saída convirja para um valor esperado.

Tendo em vista que se tratam de redes dinâmicas, as redes de Hopfield nem sempre convergem, logo para obter uma convergência é necessário escolher adequadamente os parâmetros, de modo que suas derivadas temporais sejam iguais ou menores que zero.

2.3. PROCESSAMENTO DE IMAGENS DIGITAIS

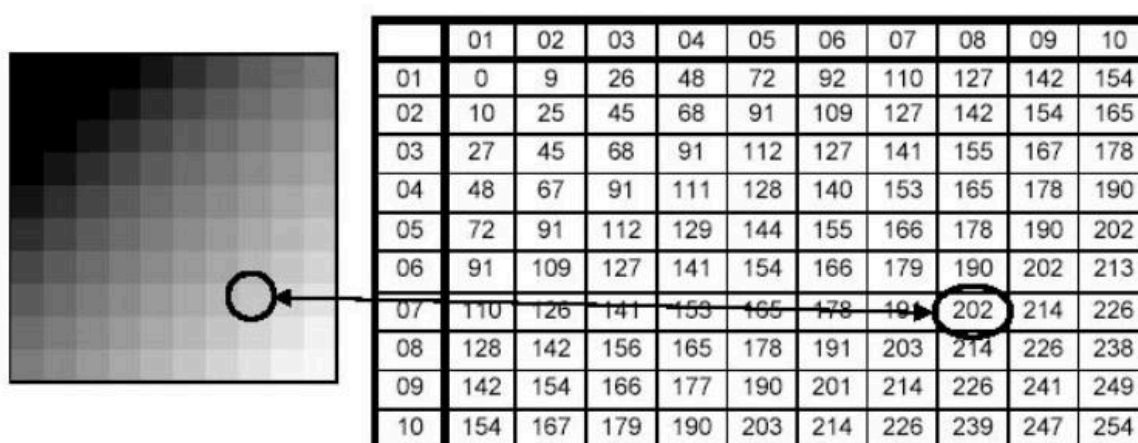
Segundo Maria (2000), o Processamento de Imagens Digitais (PID) é a manipulação de uma imagem por um computador de modo que a entrada e a saída desse processo sejam imagens. O interesse em métodos de Processamento de Imagens Digitais decorre de duas áreas principais de aplicação: melhoria de informação visual para a interpretação humana e o processamento de dados de cenas para percepção automática através de máquinas (GONZALEZ, 2000).

2.3.1. Definição de Imagem Digital

De acordo com Gonzalez (2000), uma imagem digital é uma imagem $f(x,y)$ particionada tanto em coordenadas espaciais quanto em brilho, de modo que pode ser considerada como sendo uma matriz cujos índices de linhas e colunas identificam um ponto específico da imagem, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto.

A menor unidade de uma imagem digital é denominada *picture element (pixel)*, de modo que um *pixel* é a representação numérica da luminosidade de um ponto na imagem. A figura 17 ilustra a representação numérica de uma imagem ampliada de 10 x 10 *pixels*.

Figura 17 – Matriz 10 x 10 representando uma imagem ampliada, com 256 tons de cinza.



Fonte: Souza, 2011.

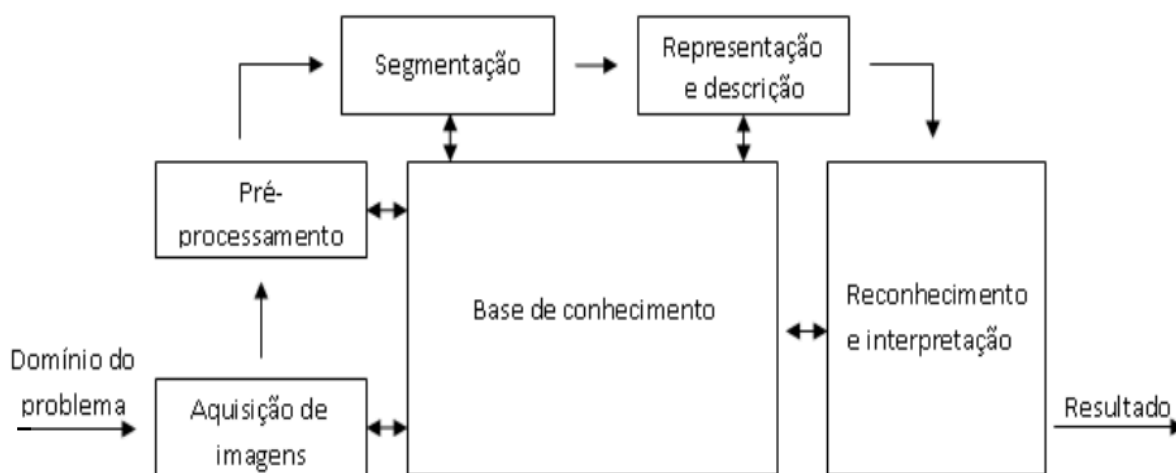
2.3.2. Passos essenciais para PID

Dentro das técnicas do processamento de imagens, temos classificações que se dão a partir da redução progressiva da quantidade de informações manipuladas, são eles o de baixo nível, de nível médio e alto nível.

Segundo Falcon (1993), no baixo nível, os dados de entrada do processamento são *pixels* da imagem original e os dados de saída representam propriedades da imagem na forma de valores numéricos associados a cada *pixel*. No processamento de nível médio, temos estes conjuntos de valores, produzidos no nível baixo, que resultam em uma lista de características, e no processamento de alto nível uma interpretação do conteúdo da imagem é gerada a partir das informações obtidas com a lista de características.

A análise de imagem envolve diversas etapas, como aquisição da imagem, pré-processamento, segmentação, representação e descrição, reconhecimento e interpretação. Tais passos devem ser desenvolvidos com cautela, pois são subsequentes, e uma falha em alguma das etapas não será corrigida nos próximos passos. A figura 18 ilustra o fluxograma de um sistema de processamento de imagem.

Figura 18 – Passos fundamentais para PID.



Fonte: Gonzalez, 2000.

- Aquisição de Imagem

Esta etapa inicial resume-se em adquirir a imagem digital, ou digitalizar uma imagem pré-existente.

Nesta etapa estão envolvidos fatores como: a escolha do sensor, o conjunto de lentes a utilizar, as condições de iluminação, a resolução, dentre outros (FILHO, 1999). Uma aquisição de imagem de alta qualidade pode ajudar a reduzir a complexidade e o tempo das etapas subsequentes de processamento, que pode diminuir o custo do sistema de processamento de imagem (DU, 2004).

- Pré-processamento

Essa etapa tem como objetivo melhorar a qualidade de uma imagem, para isso, operações devem ser executadas para remover ou diminuir o desgaste sofrido pela imagem durante a sua aquisição. Nesta etapa pode-se suprimir distorções que não são necessárias e melhorar as características de interesse para uma posterior aplicação específica (DU, 2004).

O melhoramento da imagem aumenta consideravelmente as chances de sucesso para os processos seguintes, filtragem, realce, processamento de cores são itens que podem compor esta etapa.

- Segmentação

Essa etapa divide uma imagem de entrada em partes ou objetos constituintes. Segundo Gonzalez (2000) a segmentação automática é uma das tarefas mais difíceis do PID, pois por um lado uma segmentação robusta favorece substancialmente a solução bem sucedida de um problema. Por outro lado, os algoritmos para tal feito são bastantes suscetíveis à falhas.

- Representação e descrição

Os dados oriundos da etapa de segmentação podem ser representados de duas formas: por fronteiras ou regiões complexas. A representação por fronteira é adequada quando o interesse se concentra nas características de forma externa, já a representação por regiões complexas é utilizada quando o interesse se concentra em propriedades internas. A descrição tem como finalidade extrair características que resultem em alguma informação quantitativa de interesse ou que sejam básicas para discriminação entre as classes de objetos (GONZALEZ, 2000).

- Reconhecimento e interpretação

Reconhecimento é o processo que atribui um rotulo a um objeto, baseado na informação fornecida pelo seu descritor. Já a interpretação envolve a atribuição de significado a um conjunto de objetos reconhecidos (GONZALEZ, 2000).

- Base de conhecimento

Todas as tarefas das etapas, descritas anteriormente, pressupõem a existência de um conhecimento sobre o problema a ser resolvido, armazenado em um base de conhecimento, cujo tamanho e complexidade podem variar (FILHO, 1999).

Segundo Gonzalez (2000), esse conhecimento pode ser tão simples quanto o detalhamento de regiões de uma imagem em que se sabe que a informação de interesse pode ser localizada, limitando assim a busca que precisa ser conduzida na procura por aquela determinada informação. Porém a base de conhecimento pode ser bastante

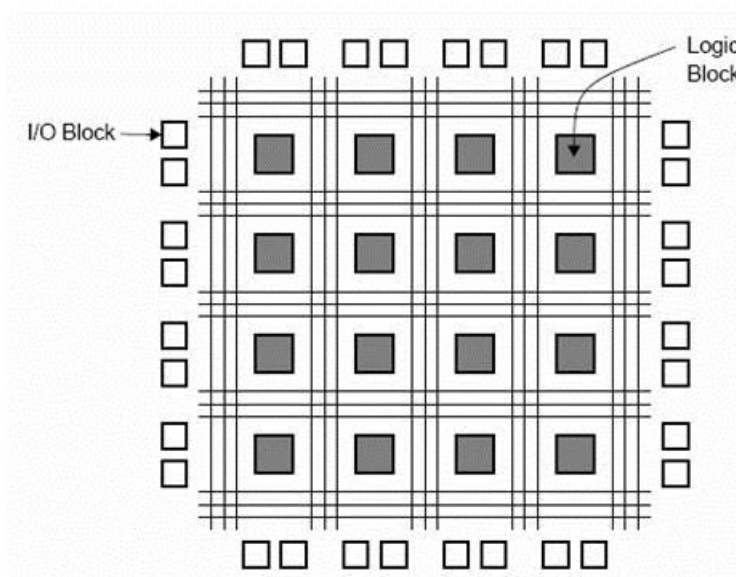
complexa, tal como uma lista que relaciona os principais defeitos possíveis de um problema de inspeção de materiais, ou uma base de imagens de satélites de alta resolução de uma região em conexão com aplicações de detecção de mudanças.

2.4. FPGA

O *Field Programmable Gate Array*, conhecido como FPGA, é um chip diferente dos que estão atualmente no mercado, pois ele possui a capacidade de reconfiguração e de processamento em paralelo de informações (tal característica possibilita a síntese de redes neurais artificiais, pois coincide com a teoria inicial do processamento distribuído de informações de uma rede neural biológica).

Esse tipo de chip possui um arranjo eletrônico interno que lhes permite assumir qualquer funcionalidade de um circuito digital, de modo que podemos programar de um simples timer de micro-ondas até um processador de vários núcleos (DELAÍ, 2009). Isso se deve porque o FPGA possui internamente vários elementos lógicos reprogramáveis chamados de CLBs (do inglês, *Configuration Logical Blocks*), que podem assumir várias configurações lógicas, sendo estes os blocos construtivos de qualquer circuito digital. A figura 19 ilustra como é a arquitetura interna desse dispositivo.

Figura 19 –
do FPGA.



Arquitetura interna

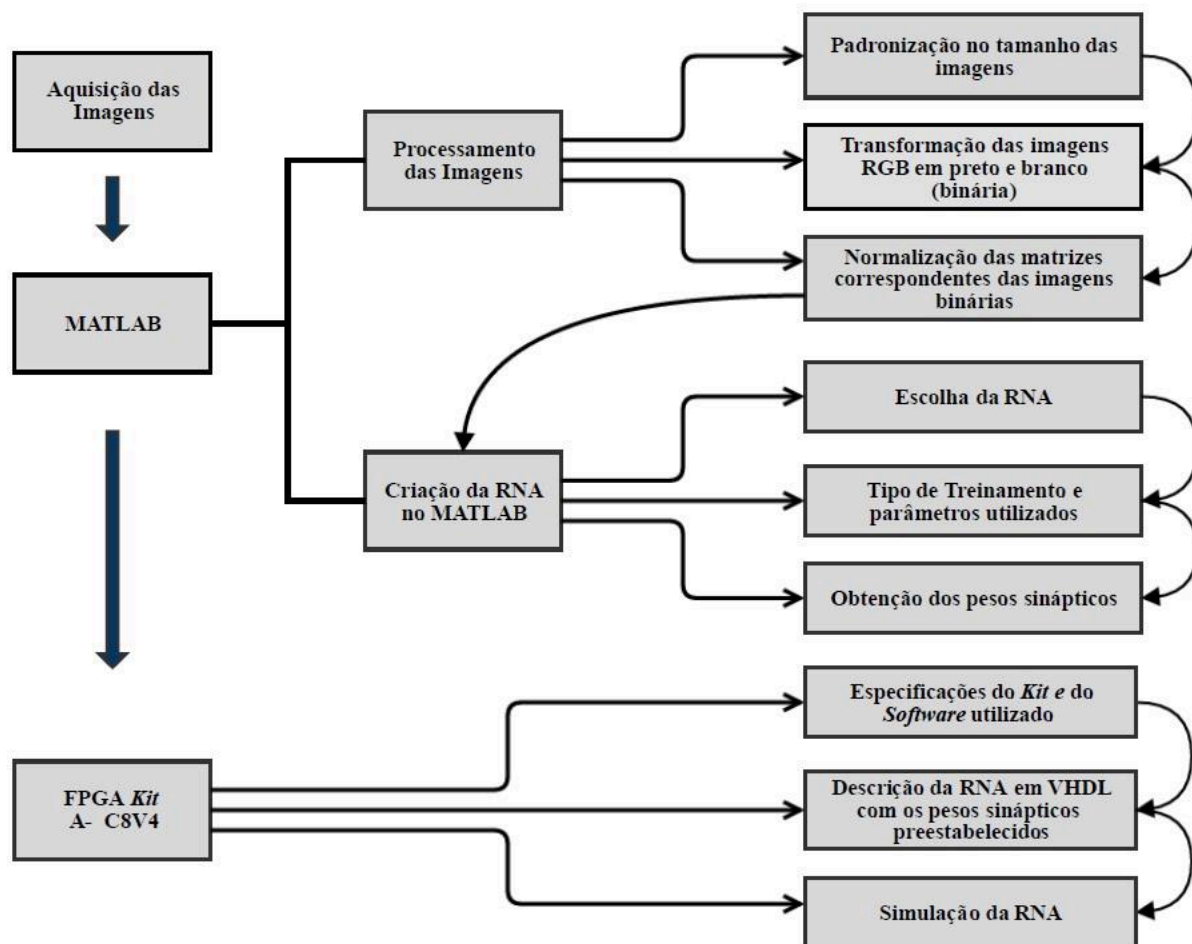
Com o auxílio das malhas de interconexões reconfiguráveis do FPGA, pode-se interligar a lógica do sistema, ou seja, realizar conexões entre CLBs e IOBs (In/Out blocks) (SOUSA, 2011). Dado que os IOBs são blocos de entradas e saídas utilizados para realizar a comunicação entre os pinos externos à lógica do Chip.

O FPGA opera de forma que, ao programar-se nele uma série de tabelas com informações de correspondência entre entradas e saídas, configura-se comportamento para um projeto específico (SOUSA, 2011). Para realizar essa programação é necessário fazer o uso de uma linguagem de descrição de *hardware* (HDL), para descrever o funcionamento da aplicação desejada. Atualmente, dois tipos de linguagem HDL são mais utilizadas, são elas: VHDL e Verilog

3. METODOLOGIA

Nesse tópico será abordado todas as etapas realizadas no presente trabalho. A figura 20 ilustra um diagrama de blocos que representa tais etapas.

Figura 20 – Diagramas de blocos da metodologia.



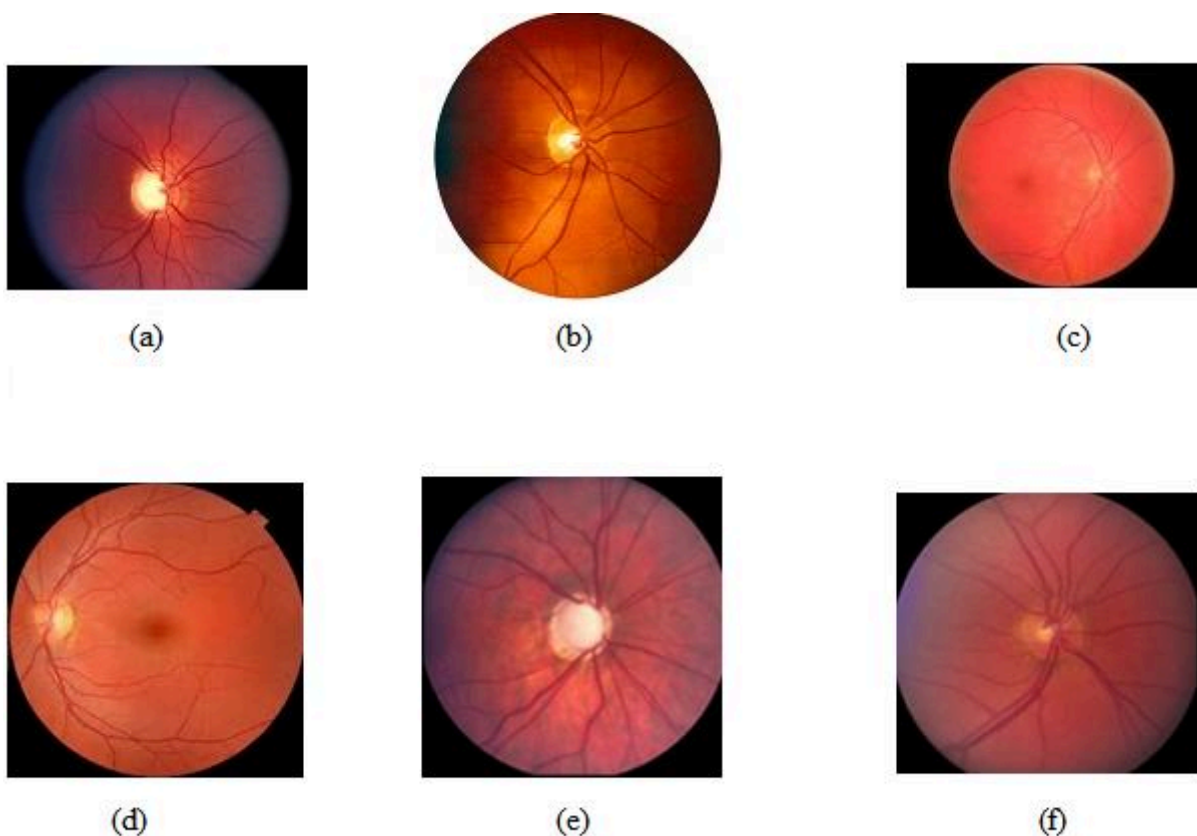
Fonte: Próprio Autor.

3.1. AQUISIÇÃO E PROCESSAMENTO DAS IMAGENS NO MATLAB

Antes de iniciar o processamento de imagem digital é necessário escolher as imagens que serão processadas, as mesmas devem possuir uma boa resolução, possibilitando que o processamento tenha mais chance de êxito.

Para o realização do presente trabalho foram escolhidas seis imagens colhidas na WEB, do exame Fundo de Olho. A Figura 21 representa as 6 amostras, onde três dessas imagens serão de olhos normais (definidas como OLHO3, OLHO4 e OLHO6) e as outras três de olhos que possuem o Glaucoma (definidas como OLHO1, OLHO2 e OLHO5).

Figura 21 – Amostra do exame de Fundoscopia: a) OLHO1; b) OLHO2; c) OLHO3; d) OLHO4; e) OLHO5; f) OLHO6.



Fonte: (a) <http://pdg.estiga.com/glaucoma-cronico-de-angulo-aberto;>

(b) [http://www.ofthalmologistabh.com.br/glaucoma-tutorial-completo-para-pacientes/;](http://www.ofthalmologistabh.com.br/glaucoma-tutorial-completo-para-pacientes/)

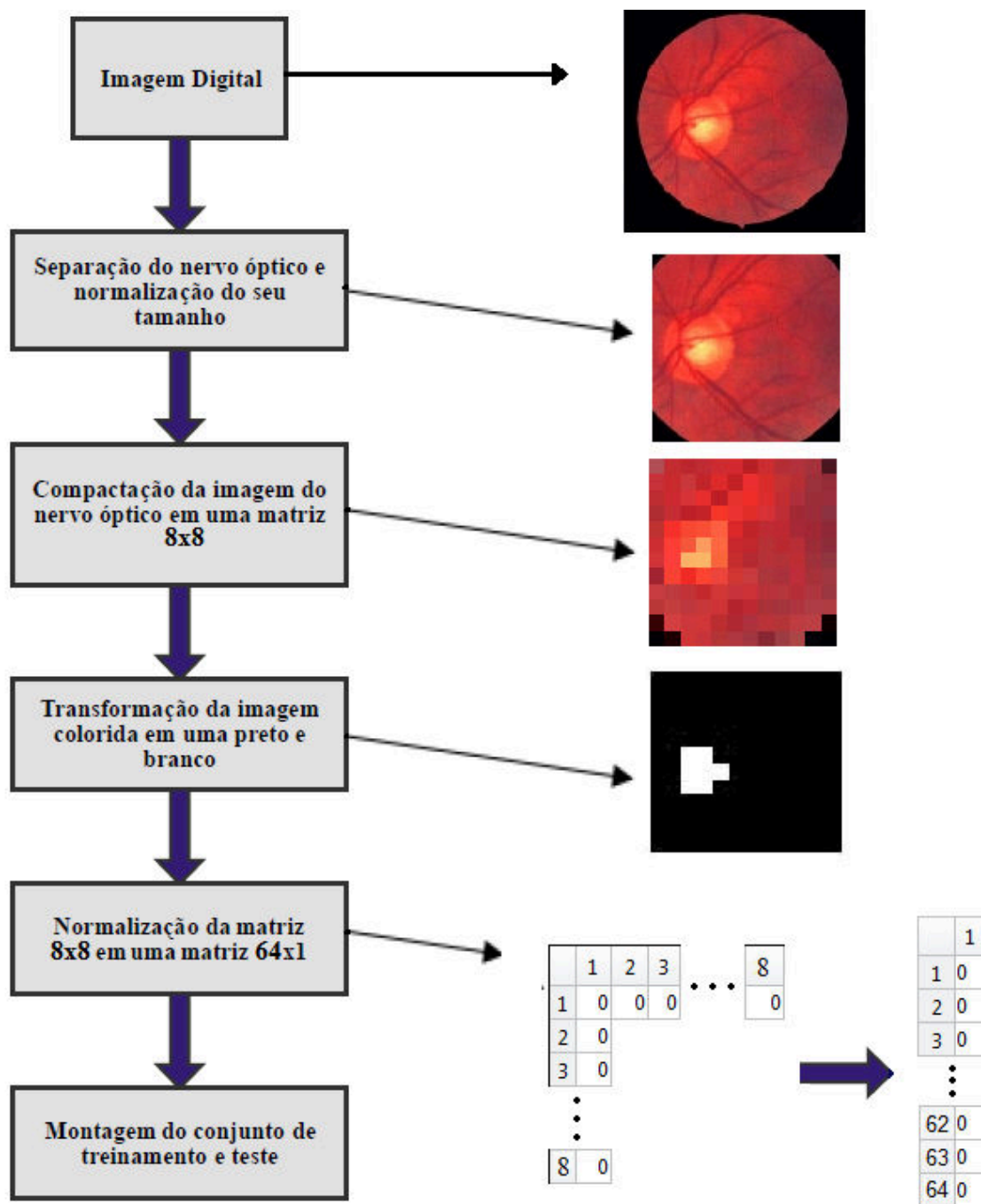
(c) [http://medsimples.com/exame-de-fundo-de-olho/;](http://medsimples.com/exame-de-fundo-de-olho/)

(d) [http://www.clinicaroisman.com.br/retinose-pigmentar/;](http://www.clinicaroisman.com.br/retinose-pigmentar/)

(e) e (f) [http://www.langfamilyeyecare.com/ocular-health/glaucoma/.](http://www.langfamilyeyecare.com/ocular-health/glaucoma/)

Após a construção do banco de dado contendo seis amostras de imagens do exame Fundo de Olho, o próximo passo é o tratamento dessas imagens, de modo que elas possam ser aplicadas como entrada da rede. A figura 22 ilustra o diagrama de blocos do processo de preparação de dados de uma imagem do exame de Fundoscopia.

Figura 22 – Esquema do processo de preparação de dados, utilizando a imagem de um exame de Fundoscopia de um olho que possui o Glaucoma.



Fonte: Próprio Autor.

O primeiro passo a partir da aquisição da imagem digital foi selecionar o tamanho padrão de 120x120 *pixels* que serviu como base para separação do nervo óptico da imagem. Tal separação tem como finalidade focar o processamento da imagem apenas na parte que é afetada pelo Glaucoma, que no caso seria no nervo óptico.

Como cada amostra possuía, em média, um tamanho de 120x120 *pixels*, o que resultaria em um vetor de 14400 entradas para a rede neural, foi necessário realizar a compactação da imagem da amostra de modo que ela se torne uma matriz de 8x8 *pixels*. Esta solução foi escolhida por resultar em um vetor de entrada razoável (64 valores) e por ser possível a implementação no FPGA devido a quantidade de pinos de entrada (para fins de teste).

O próximo passo foi converter a matriz de 8x8 *pixels* para o escala de preto e branco. Em relação à esse padrão, o valor 1 corresponde ao tamanho da escavação do nervo óptico e é representado pela cor branca, e o valor 0 corresponde ao restante do olho representado pela cor preta. Esse passo é de grande importância no sentido de facilitar a identificação do Glaucoma durante o treinamento da rede, pois com a representação de 0 e 1, torna-se evidente a diferença da escavação do nervo óptico de um olho saudável, que é bem pequena quando comparada a um olho que possui a doença.

Por fim, foi feita a normalização das matrizes 8x8 *pixels* para uma matriz de 64x1 *pixels*, de modo que cada linha dessa matriz corresponde a uma entrada da rede. Por consequência a rede irá, durante o treinamento, gerar 64 pesos sinápticos correspondentes para cada entrada. A aplicação direta da matriz 8x8 não poderia ser feita em razão da rede não conseguir identificar componentes contidos no espaço $\mathbb{R}^n$, para $n \geq 2$, ou seja, a RNA só é capaz de reconhecer como entradas componentes que estejam compreendidos no espaço unidimensional.

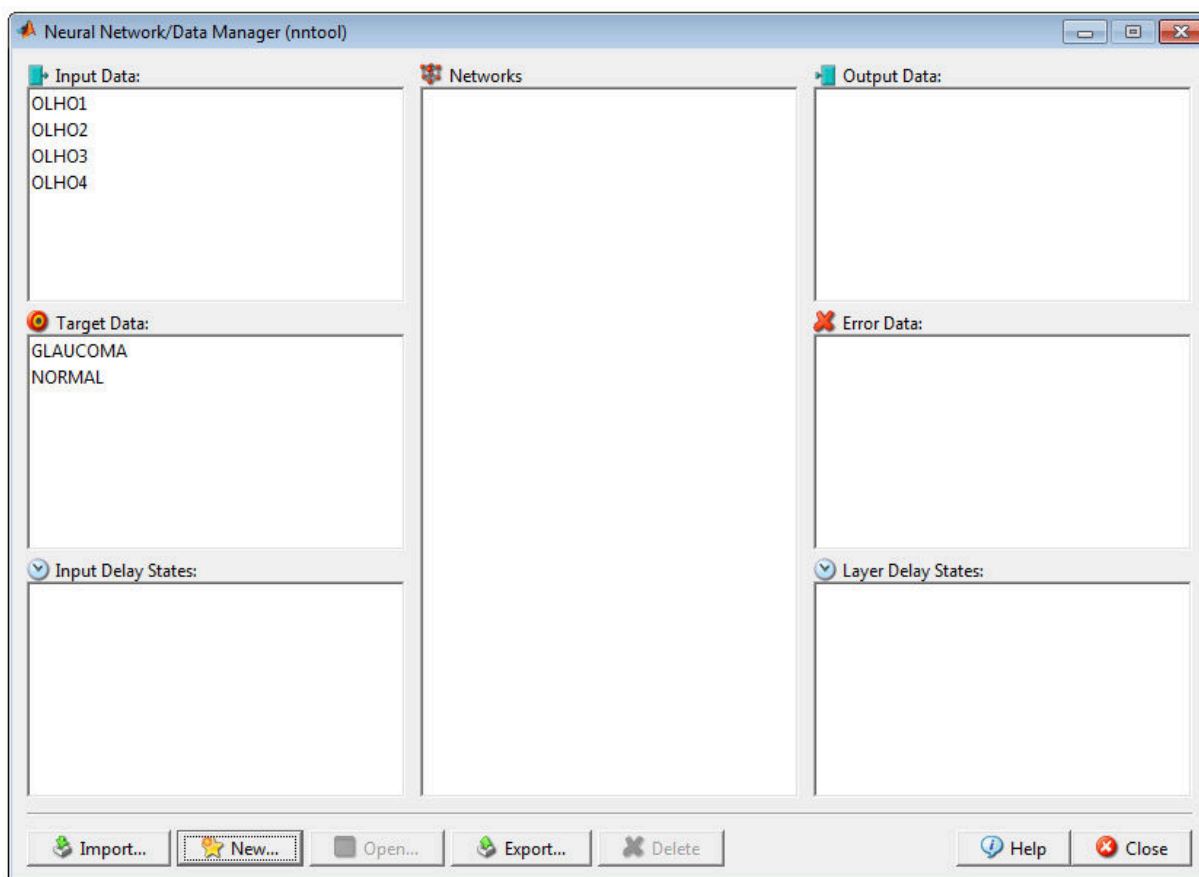
3.2. ESCOLHA E CRIAÇÃO DA REDE NEURAL ARTIFICIAL

A rede neural artificial escolhida para identificar o Glaucoma foi a rede *Perceptron* Simples. Tal escolha se deve ao fato de o problema em questão ser linearmente separável, ou seja, existem duas classes de pacientes, os que possuem o Glaucoma e os que não possuem. Essa escolha foi objetivando também uma maior facilidade de descrição e menor consumo de *hardware* durante sua implementação em FPGA.

Com a RNA escolhida e as matrizes de entradas normalizadas, inicia-se o processo de criação e treinamento da rede. Inicialmente é necessário fazer o uso de uma ferramenta de interface gráfica específica do MATLAB para criação de redes neurais artificiais chamada *Neural Network Toolbox (NNTool)*.

Para execução dessa ferramenta basta digitar o comando “*nntool*” na linha de comando do MATLAB. A figura 23 ilustra a tela inicial do *NNTool*.

Figura 23 – Interface gráfica do *NNTool* com as entradas e saídas.



Fonte: Próprio Autor.

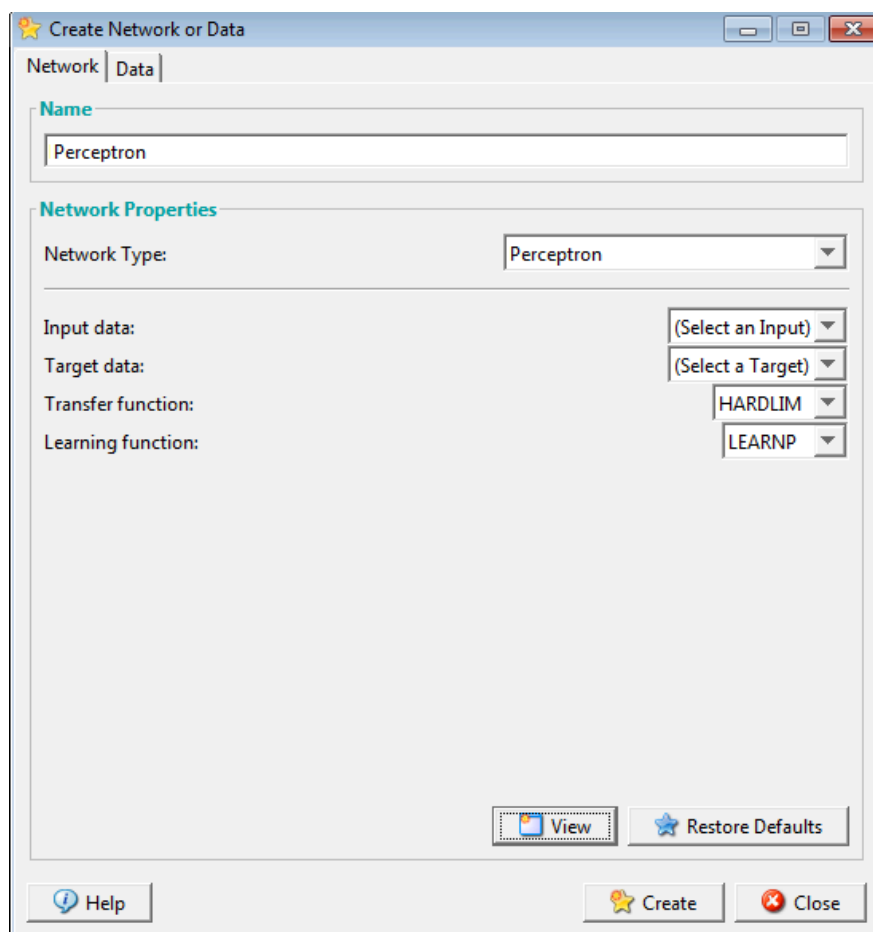
Na janela inicial dessa ferramenta observa-se vários quadros onde pode-se ver e modificar os conjuntos de dados de entrada, saída desejada, saída e erros. Tem-se também botões de importar e exportar dados, criar, abrir e excluir uma RNA.

Foram importadas, como entradas da rede, quatro das seis amostras do exame de Fundoscopia, de modo que duas delas serviram para treinar a rede e as outras duas para verificar se a rede está se comportando como o esperado, são elas: OLHO1, OLHO2, OLHO3 e OLHO4. Como saída desejada da rede, foi padronizado duas variáveis denominadas

GLAUCOMA e NORMAL, de modo que a primeira é representada pelo bit “1” e a segunda pelo bit “0”.

Após a importação dos dados que servirão para treinar a rede *Perceptron*, inicia-se o processo de criação da RNA. A figura 24 ilustra o processo de criação da rede bem como a definição dos seus parâmetros.

Figura 24 – Janela para criar a RNA e definição dos seus parâmetros.

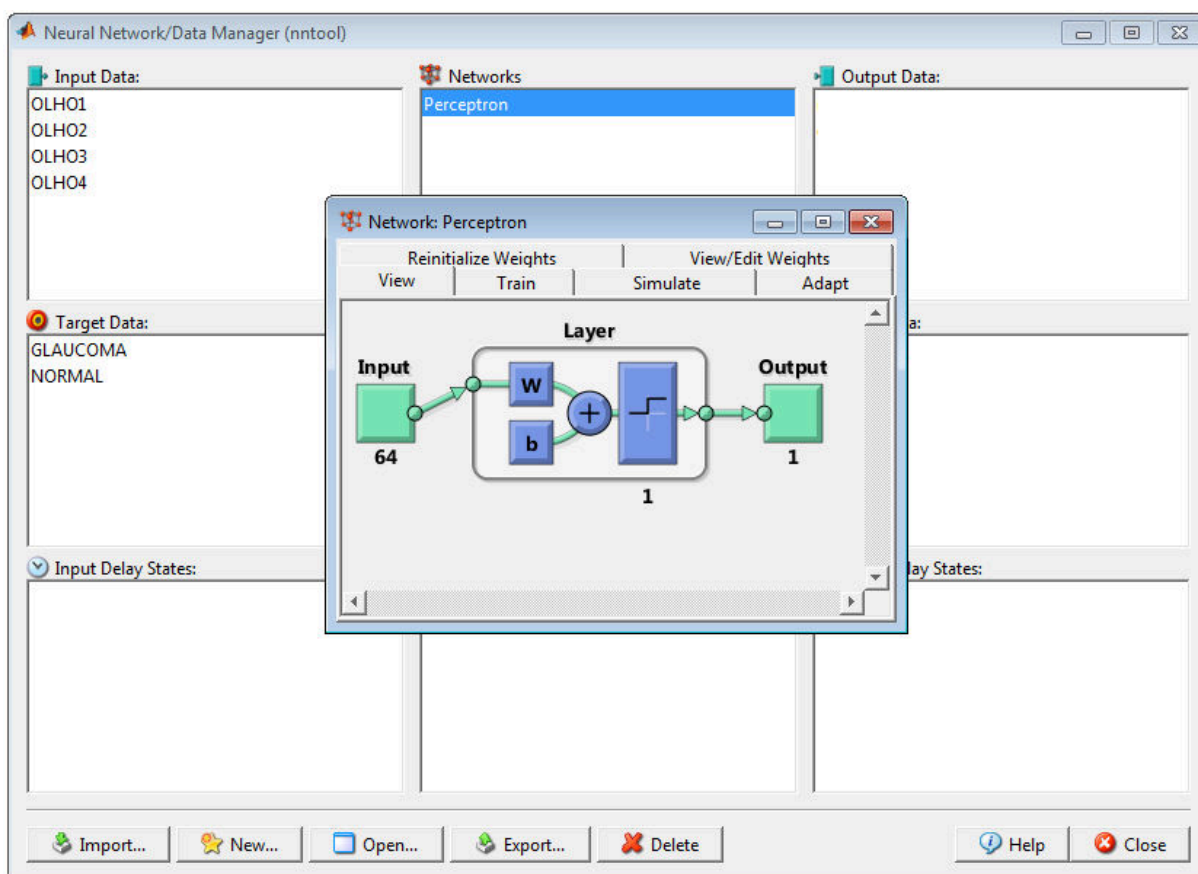


Fonte: Próprio Autor.

O tipo de rede, como já definido anteriormente, será a rede *Perceptron* Simples, de modo que sua função de transferência é *HARDLIM* (função degrau) e sua função de aprendizado é definida como *LEARNP* (tal função modifica os pesos sinápticos e o bias para um dado vetor de entrada.). Cada vez que a função *LEARNP* é executada, a rede *Perceptron* tem uma melhor chance de produzir as saídas corretas, de tal modo a convergir para uma solução em um número finito de iterações, caso a solução exista para esse tipo de rede. Com todos os parâmetros preenchidos clica-se em *Create* para finalizar a criação da RNA.

Com a rede criada e denominada como rede *Perceptron*, inicia-se o treinamento. Para se ter acesso a rede deve-se clicar duas vezes no seu nome ou clicar uma vez e pressionar o botão *Open* na parte inferior da janela do *NNTool*. Após a rede aberta é possível visualizar o seu formato, com números de entradas, quantidade de neurônios, função de transferência utilizada para o determinado neurônio e a saída. Na figura 25 podemos visualizar o formato da RNA.

Figura 25 – Diagrama da rede.



Fonte: Próprio Autor.

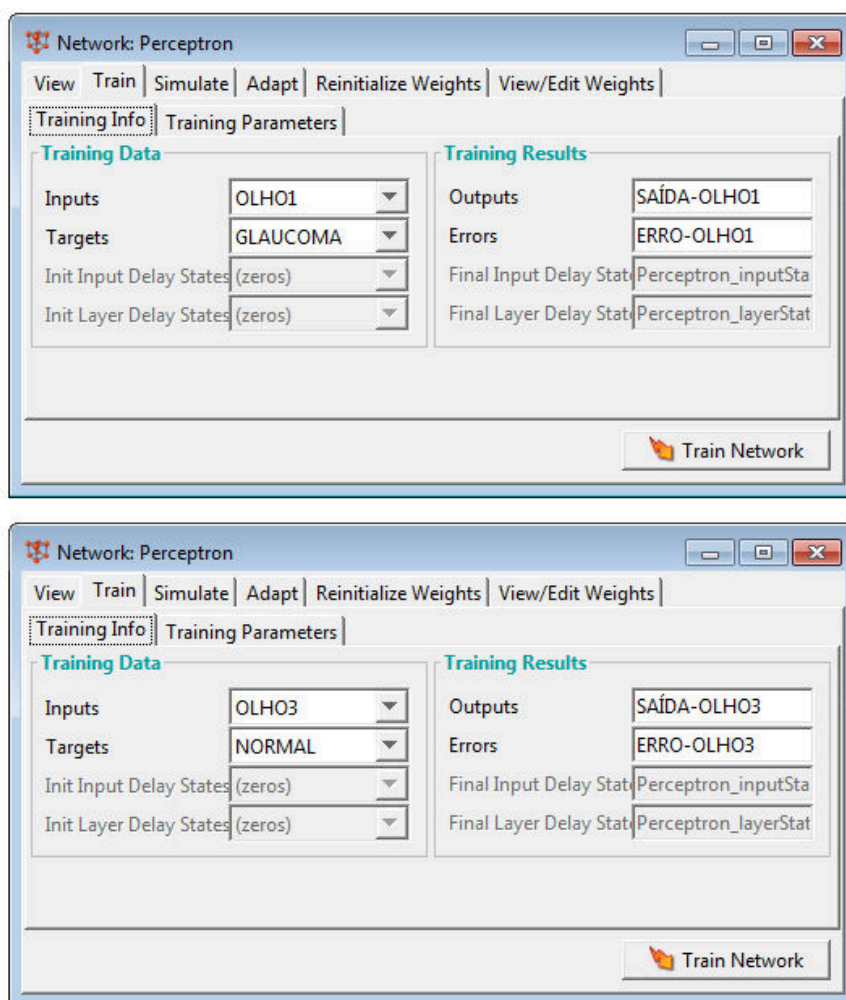
Para treinar a RNA foi aberta a opção *Train*, onde devemos passar as informações para treinamento e definir os nomes das variáveis de saída da rede. Para a rede criada foram inseridos os valores de treinamento, no qual foram definidos o OLHO1 e OLHO3 para serem os parâmetros de treinamento da rede.

O treinamento com o OLHO1, baseou-se na utilização de um vetor que continha características de um olho que possui o Glaucoma, de modo que foi colocado como a saída desejada para esse conjunto a variável GLAUCOMA, logo para um vetor de entrada que

possui características semelhantes com a do OLHO1 a rede vai acusar que aquele conjunto de entradas a saída vai ser o bit corresponde ao da variável GLAUCOMA que no caso é o bit “1”.

Já o treinamento com o OLHO3 consistiu-se em na utilização de um vetor de entrada que continha características correspondentes a um olho normal, ou seja, possuía a escavação do nervo óptico dentro dos padrões normais. Junto esse vetor de entrada foi apresentado à rede como saída desejada a variável NORMAL, de modo que a rede conseguisse estabelecer um padrão para esse conjunto de entrada e com isso aprender a reconhecer um olho normal e fornecer a saída correspondente a variável NORMAL, ou seja, fornece como saída o bit “0”.

Figura 26 – Configuração dos dados para treinamento da RNA.

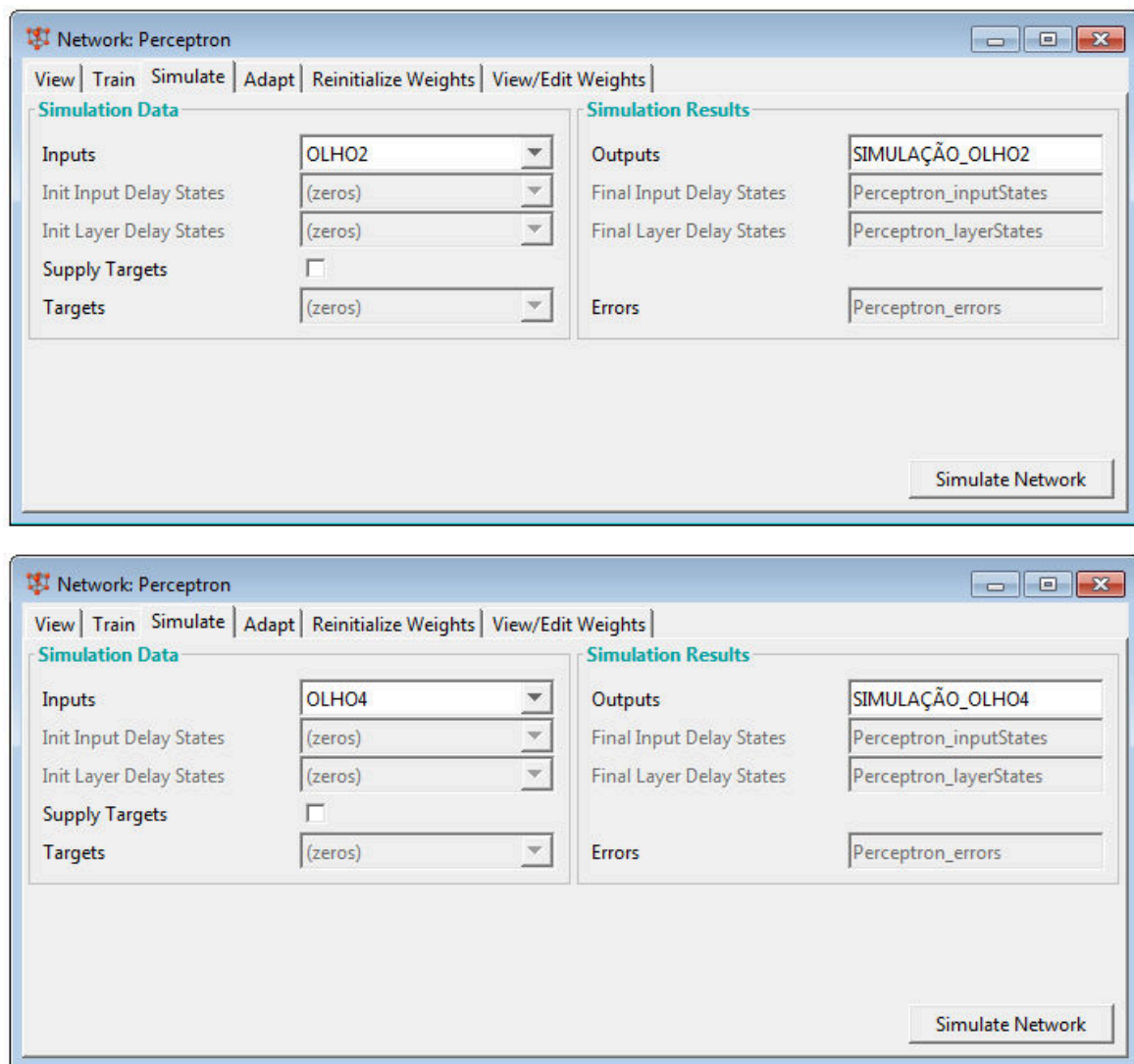


Fonte: Próprio Autor.

Para cada etapa do treinamento foram fixados variáveis representando o valor da saída gerada e o erro gerado durante o treinamento.

Após cada treinamento da RNA são gerados 64 pesos sinápticos diferentes, de tal maneira que devem ser feitas simulações, para averiguar se a rede está funcionando como o esperado. Para realização de tais simulações foram escolhidas as amostras OLHO2 e OLHO4, de tal modo que, caso a saída correspondente para cada olho for a prevista, então a rede está pronta para uso. A figura 27 mostra a janela de simulação na ferramenta *NNTool*.

Figura 27 – Simulação da rede.



Fonte: Próprio Autor.

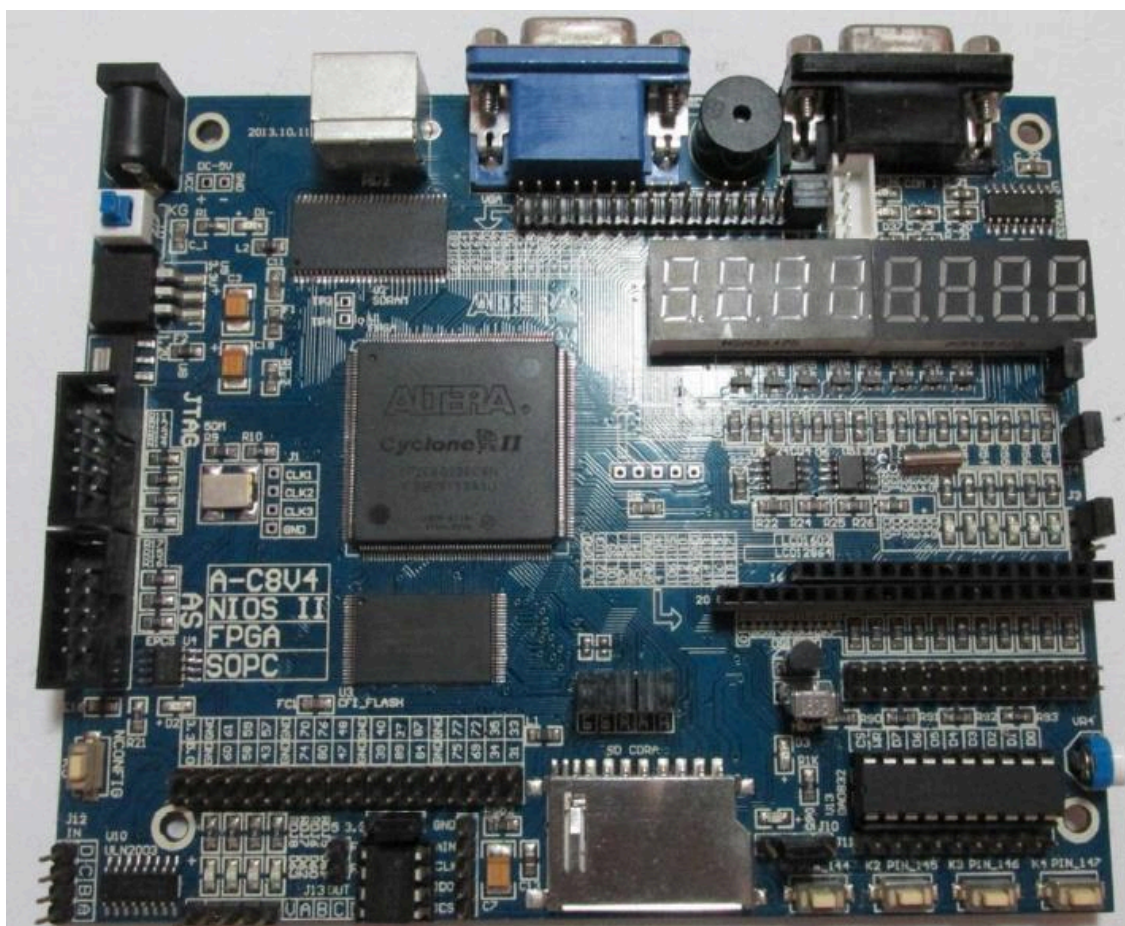
3.3. IMPLEMENTAÇÃO EM FPGA

3.3.1. Kit A – C8V4

O *kit* de desenvolvimento utilizado é o A-C8V4 possuindo, como principal componente, um FPGA Cyclone II, EPC2C8Q208C8, que possui 8.256 elementos lógicos, 165Kbits de memória interna, 18 multiplicadores embutidos e 138 pinos. Tais características estão muito além do necessário para aplicação desse trabalho, no entanto, a intercambiabilidade fornecida pelo VHDL permite que o circuito seja sintetizado em centenas de dispositivos diferentes.

Para síntese do projeto será utilizado o *software* Quartus II versão 13.0 da Altera, que está disponível gratuitamente no site da Altera, e para realizar a implementação no FPGA será utilizado um cabo USB *Blaster* que é conectado na entrada USB do computador e na entrada JTAG do FPGA. A figura 28 ilustra o *kit* utilizado.

Figura 28 – Kit A – C8V4.



Fonte: Próprio Autor.

3.3.2. Descrição da RNA em VHDL

O circuito da RNA foi descrito e sintetizado utilizando o *software* Quartus II, versão 13.0 da Altera utilizando o VHDL como linguagem de descrição de *hardware*.

A entidade, denominada como “rede”, descrita para o projeto pode ser observada na Figura 29, em que foram evidenciados os parâmetros **w1** a **w64** correspondentes aos pesos sinápticos da rede (inicialmente adotado como o vetor lógico “00”, até serem preenchidos com os valores obtidos no treinamento), o parâmetro **b1** representando o θ (inicialmente adotado como o vetor lógico “00”), as entradas **x1** a **x64** correspondendo as entradas da rede e a saída **y1**.

Figura 29 – Entidade da rede *Perceptron*.

```

6  ENTITY rede IS
7
8  GENERIC (
9
10
11  w1      : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
12  w2      : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
13  w3      : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
14  ⋮
15  ⋮
16  ⋮
17  ⋮
18  ⋮
19  ⋮
20  ⋮
21  ⋮
22  ⋮
23  ⋮
24  ⋮
25  ⋮
26  ⋮
27  ⋮
28  ⋮
29  ⋮
30  ⋮
31  ⋮
32  ⋮
33  ⋮
34  ⋮
35  ⋮
36  ⋮
37  ⋮
38  ⋮
39  ⋮
40  ⋮
41  ⋮
42  ⋮
43  ⋮
44  ⋮
45  ⋮
46  ⋮
47  ⋮
48  ⋮
49  ⋮
50  ⋮
51  ⋮
52  ⋮
53  ⋮
54  ⋮
55  ⋮
56  ⋮
57  ⋮
58  ⋮
59  ⋮
60  ⋮
61  ⋮
62  ⋮
63  w63     : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
64  w64     : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
65  b1      : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00"
66  );
67
68  PORT (
69
70  x1      : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
71  x2      : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
72  x3      : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
73  ⋮
74  ⋮
75  ⋮
76  ⋮
77  ⋮
78  ⋮
79  ⋮
80  ⋮
81  ⋮
82  ⋮
83  ⋮
84  ⋮
85  ⋮
86  ⋮
87  ⋮
88  ⋮
89  ⋮
90  ⋮
91  ⋮
92  ⋮
93  ⋮
94  ⋮
95  ⋮
96  ⋮
97  ⋮
98  ⋮
99  ⋮
100 ⋮
101 ⋮
102 ⋮
103 ⋮
104 ⋮
105 ⋮
106 ⋮
107 ⋮
108 ⋮
109 ⋮
110 ⋮
111 ⋮
112 ⋮
113 ⋮
114 ⋮
115 ⋮
116 ⋮
117 ⋮
118 ⋮
119 ⋮
120 ⋮
121 ⋮
122 ⋮
123 ⋮
124 ⋮
125 ⋮
126 ⋮
127 ⋮
128 ⋮
129 ⋮
130 ⋮
131 ⋮
132 ⋮
133 ⋮
134 ⋮
135 ⋮
136 ⋮
137 ⋮
138 ⋮
139 ⋮
140 ⋮
141 ⋮
142 ⋮
143 x63     : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
144 x64     : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
145
146 y1      : OUT BIT
147 );
148
149 END rede;
150

```

Fonte: Próprio Autor.

É de extrema importância ressaltar a praticidade resultante da criação de tipos genérico, pois permitem fácil acesso a valores de constantes no arquivo de descrição de *hardware*, bem como sua facilidade de alteração.

A arquitetura, denominada como “archp”, descrita para a rede pode ser observada na Figura 30. Inicialmente foi criado um sinal denominado **xw1** de modo que ele será o vetor resultante da operação aditiva do *Perceptron*, ou seja, esse sinal vai conter a somatória dos produtos das 64 entradas com seus respectivos 64 pesos sinápticos. Por fim foi estabelecido a saída **y1** do *Perceptron*, de modo que essa saída assume o valor do bit ‘0’ quando o resultado da função aditiva for menor ou igual a zero, e assume o bit ‘1’ quando esse resultado for maior que zero, fazendo assim o papel similar da função de ativação do tipo degrau.

Figura 30 – Arquitetura da rede *Perceptron*.

```

152  ARCHITECTURE archp OF rede IS
153
154  SIGNAL xw1 : STD_LOGIC_VECTOR (3 DOWNT0 0);
155
156  BEGIN
157
158  xw1 <= (x1*w1) + (x2*w2) + (x3*w3) + (x4*w4) + ( ... ) +
        (x62*w62) + (x63*w63) + (x64*w64) + b1;
159
160  y1 <= '0' when xw1 <= 0 else
161        '1' when xw1 > 0;
162
163  END archp ;

```

Fonte: Próprio Autor.

As bibliotecas utilizadas e suas respectivas funções na síntese de circuitos, são listadas na Tabela 1.

Tabela 1 – Bibliotecas utilizadas.

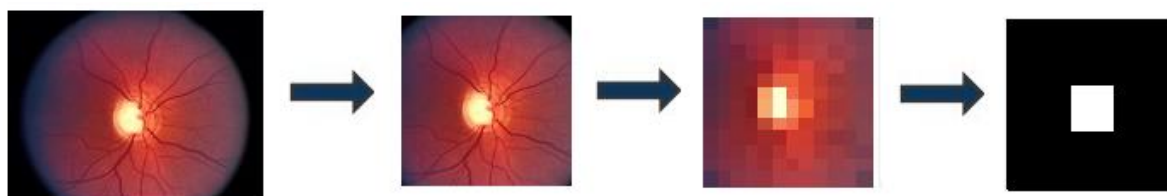
Nome da Biblioteca	Função
IEEE Std. 1164	Contém as demais bibliotecas
IEEE Std. Logic Unsigned	Contém o padrão numérico para a representação de um <i>bit</i> (<i>std_logic</i>) e de um vetor de <i>bits</i> (<i>std_logic_vector</i>).
IEEE Std. Logic Arith	Define operações aritméticas básicas (operados “+”, “-”, “*”, “/”).

4. RESULTADOS

4.1. PROCESSO DE CONSTRUÇÃO DE DADOS

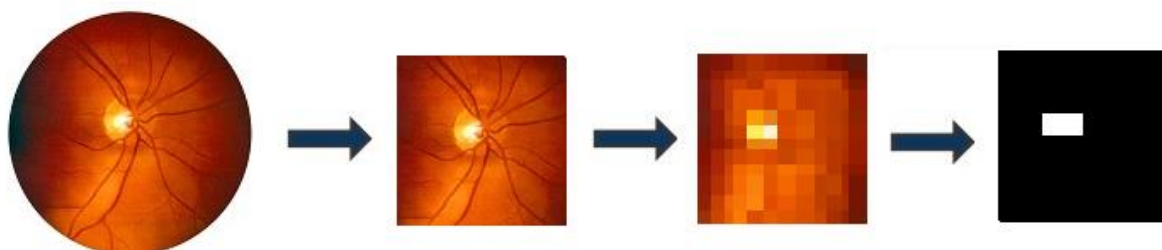
As Figuras 31 à 36 exibem o resultado do processamento de imagem digital realizado para cada uma das 6 amostras, onde, a partir do procedimento descrito na seção 3.1, foram geradas para cada amostra, uma imagem na escala preto e branco e sua respectiva matriz binária.

Figura 31 – Processamento de imagem realizado com o OLHO1.



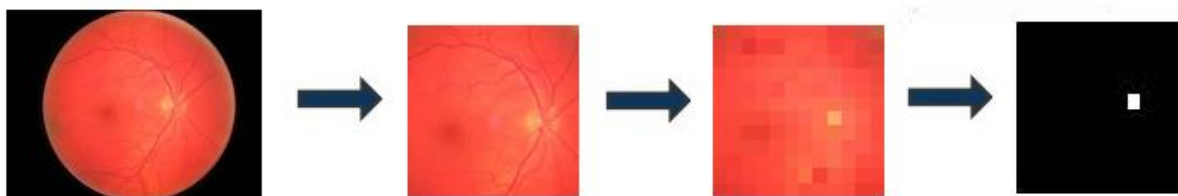
Fonte: Próprio Autor.

Figura 32 – Processamento de imagem realizado com o OLHO2.



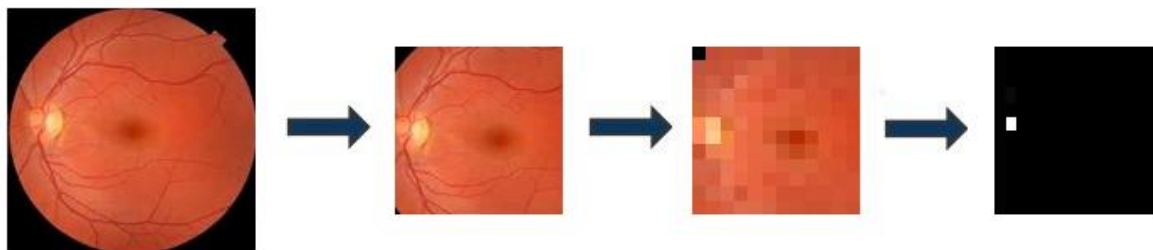
Fonte: Próprio Autor.

Figura 33 – Processamento de imagem realizado com o OLHO3.



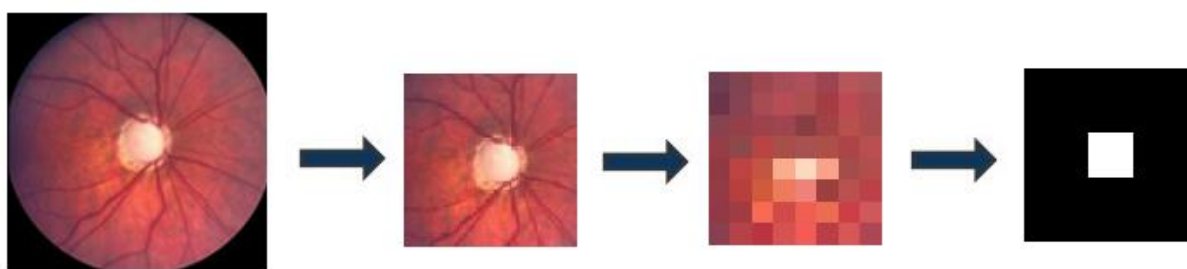
Fonte: Próprio Autor.

Figura 34 – Processamento de imagem realizado com o OLHO4.



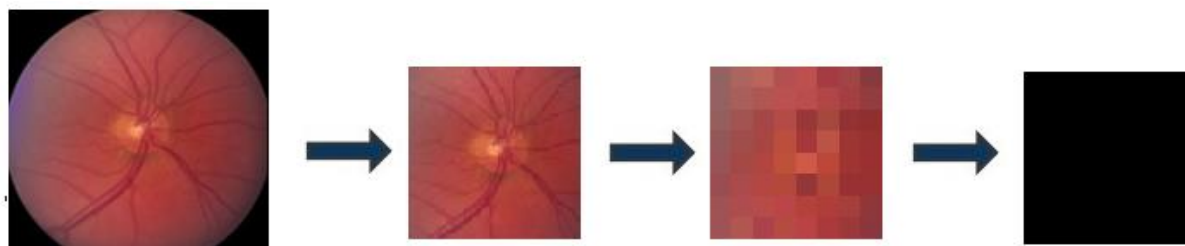
Fonte: Próprio Autor.

Figura 35 – Processamento de imagem realizado com o OLHO5.



Fonte: Próprio Autor.

Figura 36 – Processamento de imagem realizado com o OLHO6.



Fonte: Próprio Autor.

A partir do resultado do processamento de imagem realizado para cada uma das amostras, foi perceptível constatar a semelhança entre o OLHO1, OLHO2 e OLHO5, de modo que tais amostras possuem a escavação do nervo óptico (*pixel* branco) maior do que as demais amostras, constatando assim a presença do Glaucoma. Já o OLHO3 e OLHO4 possuem uma minúscula escavação do nervo óptico, confirmando assim que tais amostras representam olhos saudáveis.

O resultado obtido a partir do processamento do OLHO6, revelou que o processamento de imagem digital não deixa evidente o nervo óptico para qualquer amostra, entretanto nota-se que tal amostra corresponde a um olho normal, logo, podemos afirmar que

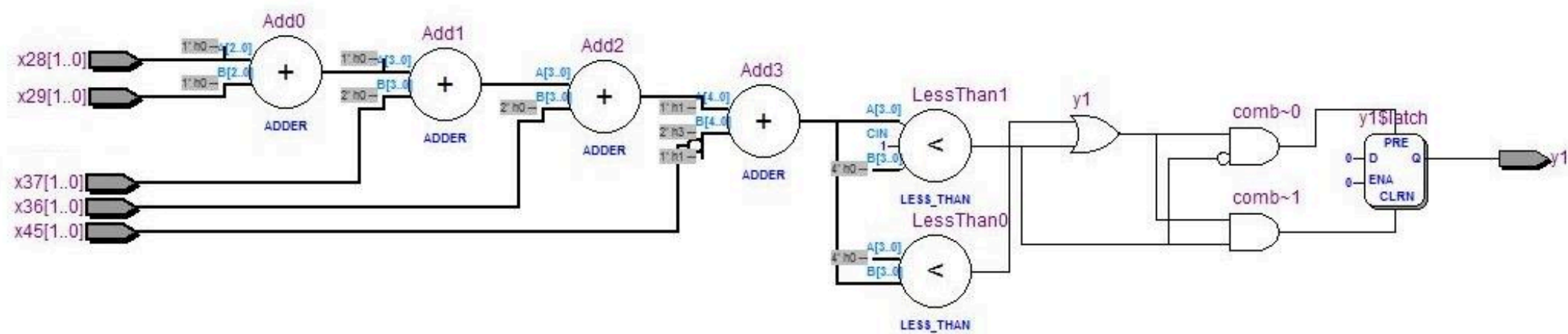


Figura 39 – Circuito equivalente à descrição da rede *Perceptron*.

Com a descrição do circuito da rede neural concluída, pode-se verificar se o FPGA, utilizado para tal implementação, possui os componentes internos necessários para a síntese da rede. Caso não possua, é necessário averiguar, no próprio *software*, os requisitos mínimos para o funcionamento do circuito que se deseja implementar, de modo a se encontrar um dispositivo adequado.

Pode-se, também, buscar alterações no projeto com base nos requisitos mínimos necessários, de modo a se obter um circuito equivalente com uma redução de custos. A Figura 40 exibe o número de componentes internos utilizados para implementação da rede *Perceptron*.

Figura 40 – Interface do *software* Quartus II, exibindo o número de componentes internos do FPGA utilizados para o projeto.

Flow Summary	
Flow Status	Successful - Thu May 05 10:18:29 2016
Quartus II 64-Bit Version	13.0.0 Build 156 04/24/2013 SJ Web Edition
Revision Name	rede
Top-level Entity Name	rede
Family	Cyclone II
Device	EP2C8Q208C8
Timing Models	Final
Total logic elements	15 / 8,256 (< 1 %)
Total combinational functions	15 / 8,256 (< 1 %)
Dedicated logic registers	0 / 8,256 (0 %)
Total registers	0
Total pins	129 / 138 (93 %)
Total virtual pins	0
Total memory bits	0 / 165,888 (0 %)
Embedded Multiplier 9-bit elements	0 / 36 (0 %)
Total PLLs	0 / 2 (0 %)

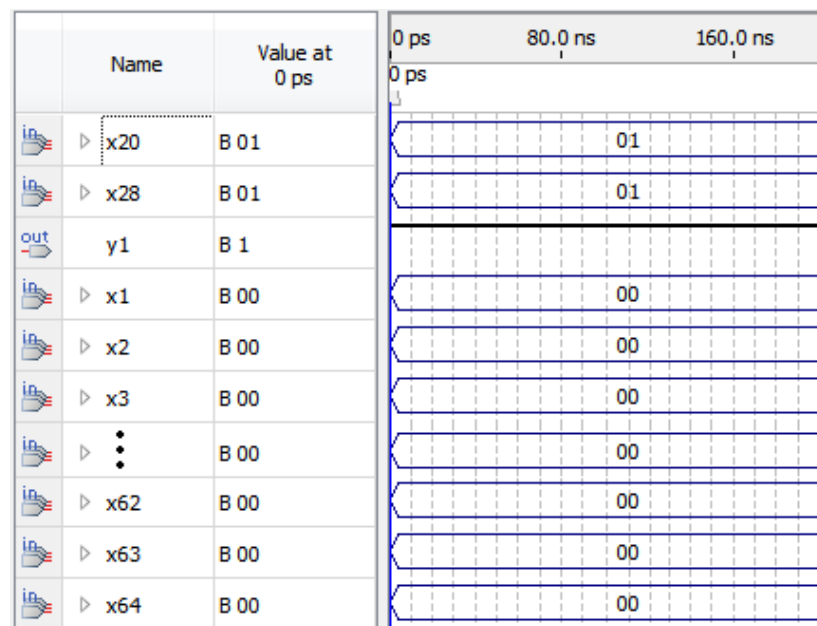
Fonte: Próprio Autor.

A utilização de pinos para representação das entradas da rede neural é utilizada apenas para fins de teste neste trabalho, pois o funcionamento real desse tipo de processo é realizado a partir de uma única memória que compõe todo o conjunto de entradas para cada amostra, de modo que o FPGA consegue em apenas um pulso de clock interpretar esse vetor memória de forma paralela. Essa característica é de grande relevância, pois isso o diferencia dos processadores atuais, que realizam tais etapas de forma puramente sequencial, acarretando assim um maior consumo de memória e uma diminuição do tempo de resposta, quando comparado com o FPGA.

4.3.1. Simulação e Teste

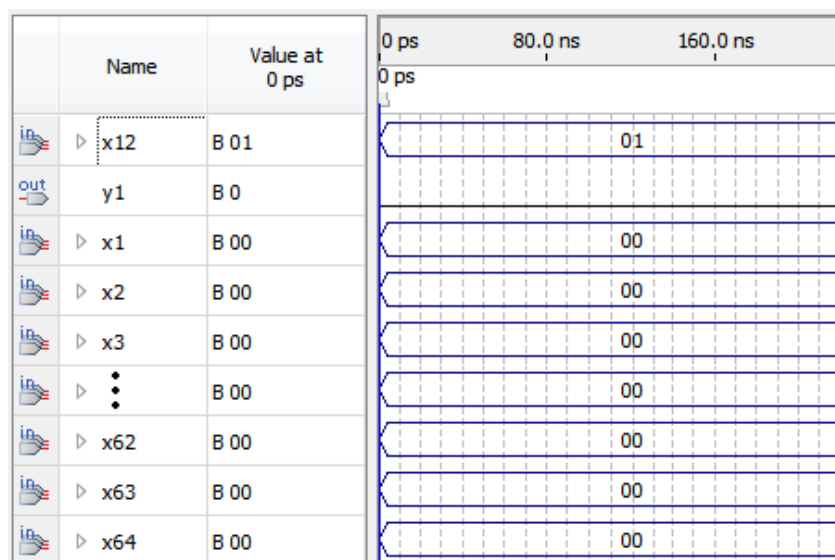
Para averiguação do funcionamento da rede *Perceptron* antes da sua implementação em FPGA, foi utilizado um recurso do *software* Quartus II, para realizar simulações o VWF (*WaveForm*) *Simulation*. As Figuras 41 à 44 exibem a saída **y1** a partir da aplicação das entradas da rede correspondente a cada uma das amostras: OLHO2, OLHO4, OLHO5 e OLHO6.

Figura 41 – OLHO2.



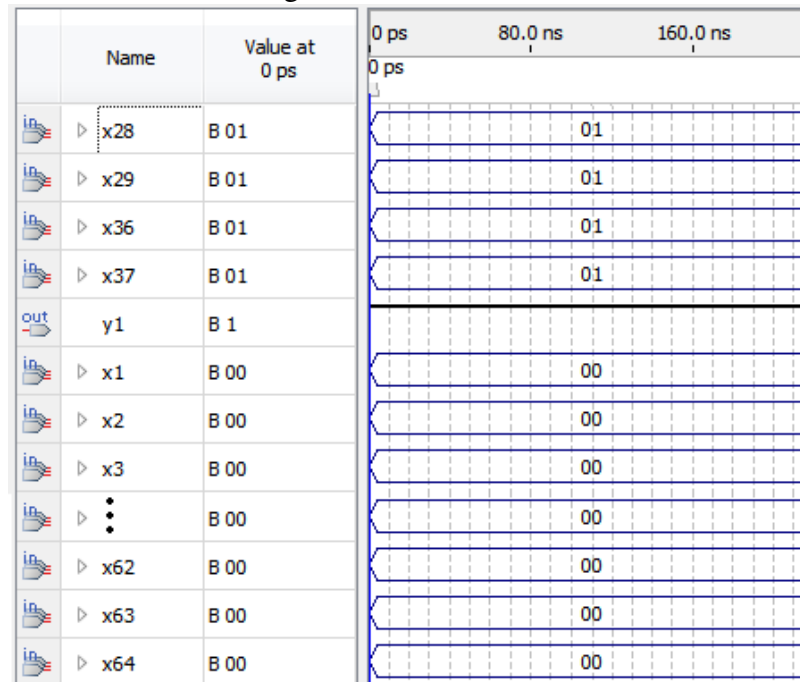
Fonte: Próprio Autor.

Figura 42 – OLHO4.



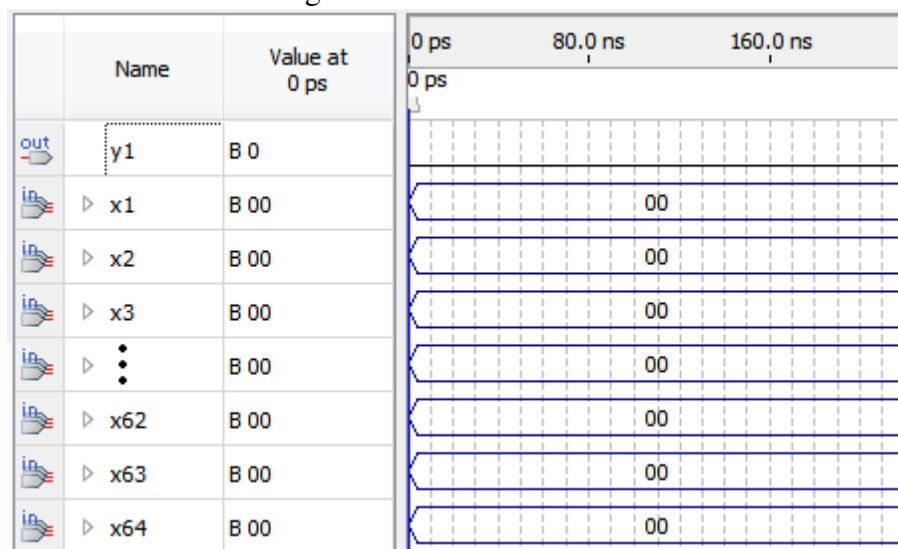
Fonte: Próprio Autor.

Figura 43 – OLHO5.



Fonte: Próprio Autor.

Figura 44 – OLHO6.



Fonte: Próprio Autor.

A saída **y1** corresponde a saída da rede *Perceptron* criada, de modo que quando assume valor do bit '1' indica que para o dado conjunto de amostra, foi identificado a presença do Glaucoma e quando assumi o valor do bit '0' representa que a amostra em questão não possui

uma escavação do nervo óptico acima do normal, logo há grande probabilidade dessa amostra não possuir o Glaucoma.

Os resultados obtidos com as simulações indicam que a rede está se comportando como o esperado para os quatro conjunto de amostras. A Tabela 2 compara as saídas obtidas no circuito sintetizado com os valor das saídas teóricas.

Tabela 2 – Verificação da rede *Perceptron* sintetizada.

Amostra	Saída y1 Teórica	Saída y1 Obtida
OLHO2	1	1
OLHO4	0	0
OLHO5	1	1
OLHO6	0	0

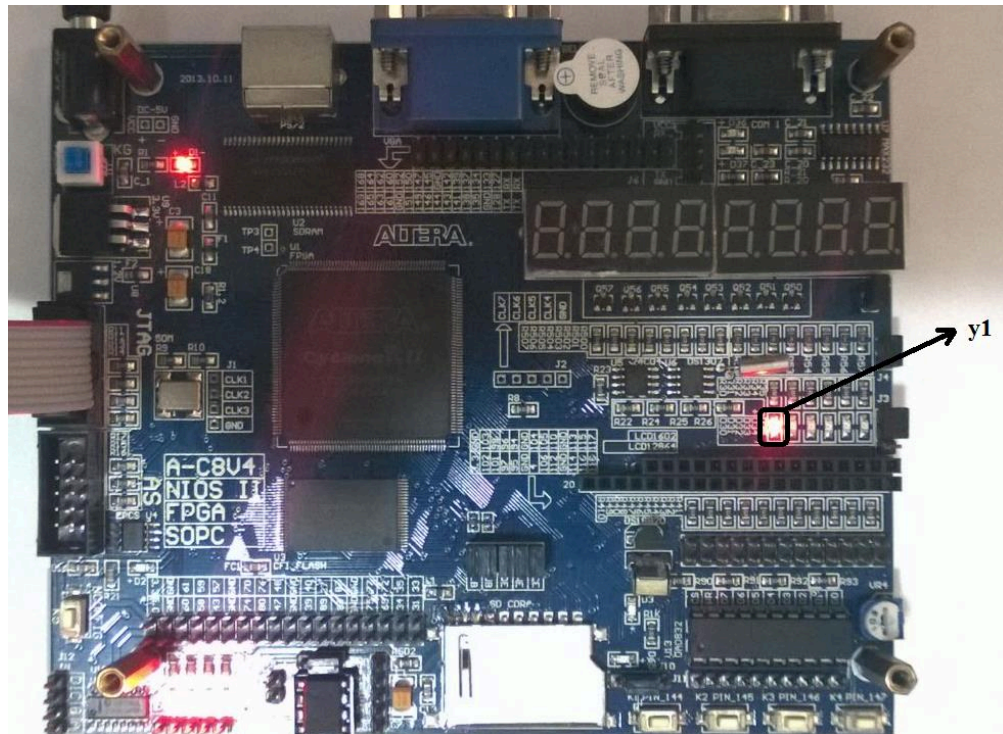
A Figura 45 exhibe o *kit* A – C8V4 da Altera com o circuito da rede *Perceptron* sintetizado, durante aplicação das entradas correspondentes à (das) amostras OLHO2 e OLHO5. É possível observar o valor da saída **y1** através do LED vermelho aplicado como provador lógico, de modo que o LED, para tal conjunto de entradas, se encontra no estado lógico alto, ou seja, ligado.

A Figura 46 exhibe o circuito sintetizado com as entradas correspondentes das amostras OLHO4 e OLHO6. Neste caso a saída **y1** se encontra no seu estado lógico baixo, portanto o LED que representa essa saída se encontra desligado.

Tal provador lógico é de grande importância, pois fornece ao usuário uma resposta visual e em tempo real da saída da rede, eliminando a necessidade de outros meios para observação da saída.

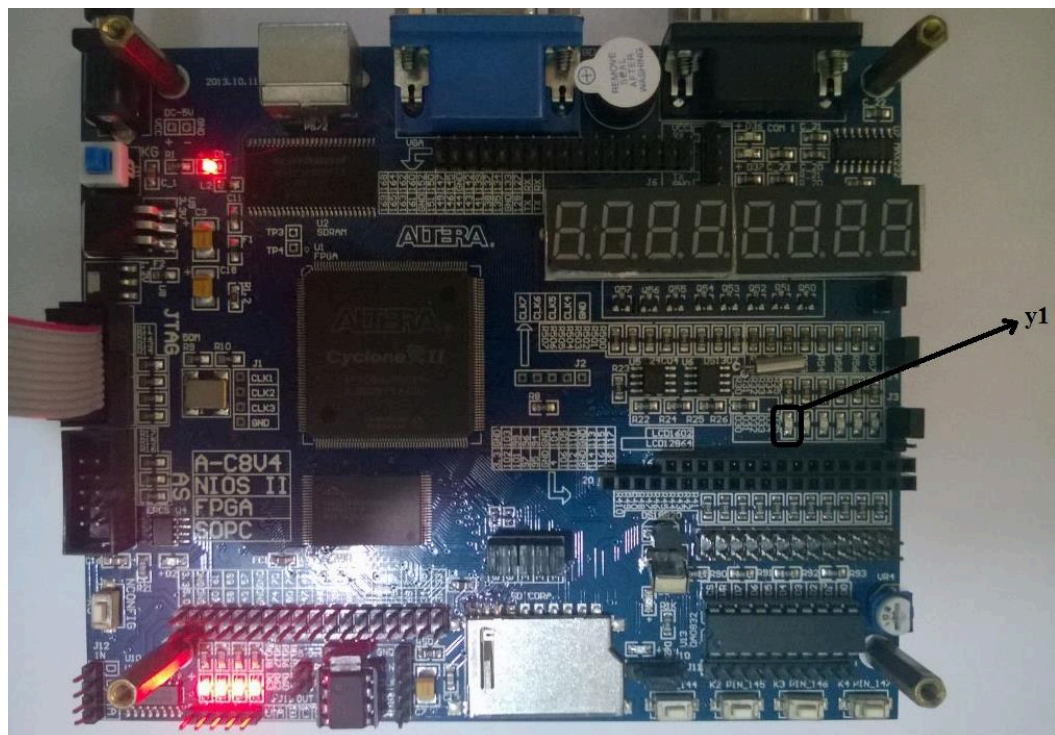
O restante dos LEDs que se encontram ligados, fazem parte da própria configuração do dispositivo de modo que toda vez que o FPGA é ligado tais LEDs permanecem acessos.

Figura 45 – Circuito da rede *Perceptron* implementado no *kit A – C8V4*, resultando a saída y_1 , correspondente as amostras OLHO2 e OLHO5.



Fonte: Próprio Autor.

Figura 46 – Circuito da rede *Perceptron* implementado no *kit A – C8V4*, resultando a saída y_1 , correspondente as amostras OLHO4 e OLHO6.



Fonte: Próprio Autor

5. CONCLUSÃO

Neste trabalho foi criada uma rede neural do tipo *Perceptron* Simples para identificar o Glaucoma a partir do processamento de imagens intraoculares. Para tal feito, foi utilizado a linguagem de descrição de *hardware* VHDL implementada no FPGA A-C8V4 da fabricante Altera. O treinamento da rede foi realizado de forma *off-line*, utilizando-se a ferramenta *Neural Network Toolbox* do *software* MATLAB.

A utilização do FPGA tornou possível a realização de diversas operações como adição e multiplicação em paralelo em nível de *hardware*, como alternativa às redes neurais executadas por *software* em máquinas sequenciais, que consistem em redes que deveriam operar em paralelo mas são processados por *hardwares* que executam tarefas de forma sequencial, como os computadores microprocessados.

A Tabela 2 permite observar que a rede *Perceptron* sintetizada no FPGA apresentou na saída y_1 obtida, os mesmos resultados da saída y_1 teórica, provando assim que a rede é apta a identificar o Glaucoma a partir de imagens intraoculares.

A utilização da linguagem de descrição de *hardware* facilitou bastante o processo de desenvolvimento da rede, pois essa linguagem é relacionada ao comportamento desejado do circuito a ser sintetizado, não exigindo, necessariamente, o conhecimento da implementação física do circuito.

A declaração de tipos genéricos durante a descrição da rede em VHDL, permitiu alteração de valores constantes presentes ao longo da descrição do circuito, através de uma simples mudança do valor declarado no início do código. Além de ser a alternativa mais rápida para substituição de valores, a sua aplicação diminui consideravelmente a probabilidade de erro durante a descrição, pois este recurso evitou que valores que deveriam ser substituídos em diferentes etapas do trabalho fossem esquecidos.

A possibilidade de obtenção dos componentes internos requisitados pelo projeto torna mais eficiente a escolha de um dispositivo ideal para o mesmo, considerando a sua capacidade de processamento e custo, fatores fundamentais para um projeto de engenharia.

5.1. TRABALHOS FUTUROS

O Trabalho apresentado pode ser estendido das seguintes formas:

- Aumentar o número de entradas para cada amostra, utilizando um dispositivo com uma maior quantidade de pinos de entrada;
- Implementação de um processo de treinamento para rede *Perceptron*;
- Elaborar um método que possa extrair as informações da imagem obtida simultaneamente com o exame de Fundoscopia, processá-las e aplicá-las como entrada da rede;
- Utilizar uma rede *Perceptron* de múltiplas camadas para realizar a identificação do Glaucoma, de modo a comparar com a rede *Perceptron* simples;
- Sintetizar outros tipos de topologias de RNA's e fazer uma análise comparativa com a rede *Perceptron* simples.

6. REFERÊNCIAS

BONFADINI, Gustavo. (Rio de Janeiro). **GLAUCOMA**. 2013. Disponível em: <<http://www.iorj.med.br/glaucoma-2/>>. Acesso em: 8 out. 2015.

BRAGA, A.; CARVALHO, A. P. L. F.; LUDEMIR, T. R. **Redes neurais artificiais: teoria e aplicação**. 2 ed. Rio de Janeiro: LTC, 2007. 226p

BRASIL. INSTITUTO BENJAMIN CONSTANT. **Um Olhar Sobre a Cegueira - Glaucoma**. 2005. Disponível em: <<http://www.ibc.gov.br/?itemid=118>>. Acesso em: 10 out. 2015.

CORIC, S.; LATINOVIC, I.; PAVASOVIC, A.; **A Neural Network FPGA Implementation**. *5th seminar on Neural Network Applications in Electrical Engineering*. (2000).

D'AMORE, R. **VHDL, Descrição e síntese de Circuitos Digitais**. 2 ed. São Paulo: LTC, 2005.

DELAÍ, André Luiz. **Softcores em FPGAs: microprocessadores do futuro?** 2009. Disponível em: <<http://www.hardware.com.br/artigos/softcores-fpga/>>. Acesso em: 7 dez. 2015.

DU, C.; SUN, W. Learning techniques used in computer vision for food quality evaluation: a review. **Journal of Food Engineering**, v.72, p.39-55, 2006.

FACELI, K.*et al.* **Inteligência Artificial: Uma abordagem de aprendizado de Máquina**. Rio de Janeiro: Editora LTC, 378p, 2011.

FACON, J. **Processamento e análise de imagem**. Córdoba: Universidad Nacional, 1993.

FILHO, O. M.; NETO, H.V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

GONZALEZ, Rafael C. **Processamento de imagens digitais**. São Paulo: Edgar Blucher, 2003.

HARISPRASATH, S.; PRABAKAR, T. N. **FPGA implementation of multilayer feed forward neural network architecture using VHDL**. *International Conference on Computing Communication and Application (ICCCA)*, pp 1-6. 2012.

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

HOLANDA, Priscila. C. **Desenvolvimento de uma rede neural multilayer Perceptron embarcada em FPGA**. 2013. 68f. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal do Ceará, Fortaleza, 2013.

INSTITUTO DE OFTALMOLOGIA DO RIO DE JANEIRO. **Glaucoma**. Disponível em: <<http://www.iorj.med.br/glaucoma-2/>>. Acesso em: 07 jan. 2016.

INSTITUTO DE OLHOS CONTINENTE. **Glaucoma**. Disponível em: <<http://www.iocflorianopolis.com.br/consulta/27/glaucoma>>. Acesso em: 15 fev. 2016.

KOVÁCS, L. Z. **Redes Neurais Artificiais: Fundamentos e Aplicações**. Livraria da Física. 2006.

MARIA, L. G. F. **Processamento Digital de Imagens**. INPE, Junho de 2000.

MATSUNAGA, Victoria Yukie. **Curso de Redes Neurais utilizando o MATLAB**. Belém. 2012.

MELLO, Paulo Augusto de Arruda. (São Paulo). **TUDO SOBRE O GLAUCOMA**. 2013. Disponível em: <<http://www.cbo.net.br/novo/publico-geral/tudosobreoglaucoma.php>>. Acesso em: 8 dez. 2015.

MENDONÇA, Marcelo. **Pesquisa inédita: pacientes com glaucoma procuram médico apenas quando estão quase cegos**. 2013. Disponível em: <<http://glaukos.com.br/2013/10/>>

pesquisa-inedita-pacientes-com-glaucoma-procuram-medico-apenas-quando-estao-quase-cegos/>. Acesso em: 8 dez. 2015.

PIÃO, Silvano S. **Implementação de uma Rede Neural Artificial em FPGA utilizando VHDL**. 2012. 48f. Monografia (Graduação em Engenharia Elétrica) -Escola de Engenharia de São Carlos, São Carlos, 2012.

PÓVOA, Cristiane Araújo *et al.* **Prevalência de glaucoma identificada em campanha de detecção em São Paulo**. 2001. Disponível em:< http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492001000400006>. Acesso em: 11 jan. 2016

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. **Redes neurais artificiais: para engenharia e ciências aplicadas**. São Paulo: Artliber, 2010.

SOARES, Pablo L. B. **Aplicação de rede neurais em um conjunto de wavelets na detecção e atenuação de ruídos em imagens digitais e sísmicas**. 2013. 79f. Dissertação (Mestrado em Ciências da Computação) – Universidade Federal Rural do Semiárido, Mossoró, 2013.

SOUSA, Miguel Angelo de Abreu de; TORRES, Thiago Felipe de Jesus. Implementação direta de uma Rede Neural Artificial em hardware e sua aplicabilidade no reconhecimento de padrões para separação de frutas. **Sinergia**, São Paulo, v. 12, n. 1, p.50-58, jan. 2011. Disponível em: <http://www.cefetsp.br/edu/prp/sinergia/complemento/sinergia_2011_n1/pdf_s/segmentos/artigo_06_v12_n1.pdf>. Acesso em: 7 out. 2015.

SOUZA, Lenivaldo R. **Algoritmo para reconhecimento e acompanhamento de trajetórias de padrões em vídeos**. 2011. Monografia (Graduação em Engenharia da Computação) – Universidade Federal do Vale do São Francisco, Juazeiro, Bahia, 2011.

URBANO, Andréia Peltier *et al.* **Avaliação dos tipos de glaucoma no serviço de oftalmologia da UNICAMP**. 2003. Disponível em:<http://www.scielo.br/scielo.php?pid=S0004-27492003000100012&script=sci_arttext>. Acesso em: 9 out. 2015