

UNIVERSIDADE FEDERAL DE ITAJUBÁ - CAMPUS ITABIRA

TAMIRIS SOUZA FONSECA
CHARLY BRAGA VENTURA

RECONHECIMENTO DE PLACAS DE AUTOMÓVEIS UTILIZANDO
PROCESSAMENTO DIGITAL DE IMAGENS E REDES NEURAIAS
ARTIFICIAIS

ITABIRA
2017

TAMIRIS SOUZA FONSECA
CHARLY BRAGA VENTURA

RECONHECIMENTO DE PLACAS DE AUTOMÓVEIS UTILIZANDO
PROCESSAMENTO DIGITAL DE IMAGENS E REDES NEURAIIS
ARTIFICIAIS

Monografia apresentada à Universidade Federal
de Itajubá – Campus Itabira como requisito fi-
nal para aprovação no curso de graduação em
Engenharia de Computação.

Professor Orientador: Prof. Dr. Sandro Carva-
lho Izidoro.

ITABIRA
2017

TAMIRIS SOUZA FONSECA
CHARLY BRAGA VENTURA

RECONHECIMENTO DE PLACAS DE AUTOMÓVEIS UTILIZANDO
PROCESSAMENTO DIGITAL DE IMAGENS E REDES NEURAIAS
ARTIFICIAIS

Pontuação obtida: _____

Itabira, ____ de _____ de 2016.

Prof. Dr. Sandro Carvalho Izidoro.
Professor Orientador

BANCA EXAMINADORA

RESUMO

O reconhecimento de placas de automóveis por sistemas computacionais consiste em um conjunto de técnicas que vão desde a captura da imagem da placa de um carro até a identificação correta da informação nela contida. Redes Neurais Artificiais (RNAs) são estruturas que simulam algumas características do cérebro humano e podem ser uma ótima técnica para reconhecimento de padrões, pois diferentemente dos algoritmos padrões, têm a capacidade de realizar tarefas e tomar decisões de forma autônoma. A união das técnicas de Processamento Digital de Imagens (PDI) e RNAs é uma poderosa ferramenta para o tratamento e reconhecimento de imagens. A metodologia de trabalho consistiu em utilizar procedimentos de limiarização, análise de conectividade de pixels, análise de padrões e geração de ruídos para o PDI. Foram feitos testes e análises de variações de parâmetros na construção das duas RNAs utilizadas: na RNA que reconhece números (RNA 1) e na RNA que reconhece letras (RNA 2). No resultados, a parte de PDI reconheceu um total 78% das imagens utilizadas para treinamento e 60% das utilizadas para a generalização. A RNA 1, no treinamento, obteve acurácia de 98,17% e erro médio quadrado (ϵ_{med}) de 0,0024. Na generalização ela obteve desempenho de 81,25% (acertou 13 de 16 caracteres). A RNA 2, no treinamento, obteve acurácia de 92,70% e erro médio quadrado (ϵ_{med}) de 0.0066. Na generalização ela obteve desempenho de 58,33% (acertou 7 de 12 caracteres). Devido à limitação do tamanho do conjunto de dados utilizado para o desenvolvimento do projeto (total de 28 imagens de placas de carros) o sistema em conjunto reconheceu todos os caracteres em apenas 1 de 7 placas (14,29%) durante o processo de generalização.

Palavras-chave: Redes Neurais Artificiais, Processamento Digital de Imagens, Reconhecimento de Padrões

ABSTRACT

Car license plates recognition by computational systems consists in a set of techniques, ranging from car plate's image capture to correct recognition of information contained on it. Artificial Neural Networks (ANNs) are structures which simulate some characteristics of the human brain and may be a great technique for pattern recognition since, differently than default algorithms, it has the capacity to do tasks and make decisions autonomously. The union between Digital Image Processing (DIP) techniques and ANNs is a powerful tool for the treatment and recognition of images. The work methodology consisted in using procedures of thresholding, pixel connectivity analysis, pattern analysis and noise generation for the DIP. Tests and analysis of parameter variation in the two ANNs built were made: in the ANN which recognized numbers (ANN 1) and in the ANN which recognizes letters (ANN 2). In the results, the DIP part recognized a total of 78% of images used for training and 60% of the images used for generalization. ANN 1, in training, obtained an accuracy of 98.17% and average quadratic error (ε_{med}) of 0.0024. In generalization, it obtained a performance of 81.25% (hitting 13 out of 16 characters). ANN 2, in training, obtained an accuracy of 92.70% and average quadratic error (ε_{med}) of 0.0066. In generalization, it obtained a performance of 58.33% (hitting 7 out of 12 characters). Due to the limitation of the size of the data set used for project development (a total of 28 images of car license plates), the system in group recognized all characters in only 1 out of 7 plates (14.29%) during generalization process.

Keywords: Artificial Neural Networks, Digital Image Processing, Pattern Recognition

LISTA DE ILUSTRAÇÕES

Figura 1 – Imagem original à esquerda e imagem transformada para escala de cinzas à direita	14
Figura 2 – Imagem original à esquerda e imagem limiarizada à direita	14
Figura 3 – Vizinhança-de-4	14
Figura 4 – Vizinhança-de-8	15
Figura 5 – Partes de um neurônio.	16
Figura 6 – Modelo de um neurônio artificial.	17
Figura 7 – Gráfico da função limiar.	20
Figura 8 – Gráfico da função linear saturada.	20
Figura 9 – Gráfico da função sigmoide.	21
Figura 10 – Gráfico da função tangente hiperbólica.	22
Figura 11 – Rede neural com uma camada.	22
Figura 12 – Rede neural MLP	23
Figura 13 – Grafo de fluxo de sinal ressaltando os detalhes do neurônio e saída j	24
Figura 14 – Representação da convergência do valor do peso, começando com um valor peso inicial para chegar a um valor peso final. As setas ao lado da curva representam a direção do gradiente dando o sentido do mínimo erro global .	25
Figura 15 – Gráfico Erro x Peso demonstrando a presença de um mínimo local além de um mínimo global na superfície de erro. Dependendo da taxa de aprendizagem, o algoritmo da descida de gradiente pode ficar preso no mínimo local	27
Figura 16 – Grafo de fluxo de sinal ilustrando o efeito da constante de momento α	28
Figura 17 – (I) - Imagem de entrada; (II) - Caracteres segmentados; (III) - Rótulo de cada caracter	30
Figura 18 – Exemplo de caracteres destacados após limiarização	31
Figura 19 – Caracteres das placas nacionais são mono-espaçados	32
Figura 20 – Estrutura básica da RNA 1	33
Figura 21 – Estrutura básica da RNA 2	34
Figura 22 – Fluxograma do sistema	35
Figura 23 – Exemplo de resultado do PDI para uma das imagens de entradas	36
Figura 24 – Exemplos de sucesso do PDI	36
Figura 25 – Exemplo de caracteres com perda de informação	37
Figura 26 – Matriz de Confusão e Curva ROC para a RNA 1	43
Figura 27 – Matriz de Confusão e Curva ROC para a RNA 2	44
Figura 28 – Testes feitos durante a generalização	45

LISTA DE TABELAS

Tabela 1 – Características das RNAs	33
Tabela 2 – Comparação de qualidade de uma RNA de 4 saídas e uma RNA de 10 saídas	39
Tabela 3 – Comparação da qualidade de uma RNA em função das diferentes funções de ativação	39
Tabela 4 – Comparação de desempenho em uma RNA com uma e com duas camadas ocultas	40
Tabela 5 – Desempenho de uma RNA de reconhecimento de números em função dos neurônios nas camadas ocultas	41
Tabela 6 – Desempenho de uma RNA de reconhecimento de letras em função dos neurônios nas camadas ocultas	41
Tabela 7 – Desempenho de uma RNA em função das variações dos passos e momentos - semi-final	42
Tabela 8 – Desempenho da RNA 1 em função das variações dos passos e momentos - final	42
Tabela 9 – Ciclo de treinamento x Conjunto de treinamento de uma RNA	43
Tabela 10 – Variações dos passos e momentos	49

LISTA DE ABREVIATURAS E SIGLAS

RNA	Rede Neural Artificial
PDI	Processamento Digital de Imagens

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.1.1	Objetivo Geral	11
1.1.2	Objetivos Específicos	11
1.2	Organização do documento	12
2	FUNDAMENTAÇÃO CONCEITUAL	13
2.1	PDI	13
2.2	RNAs	15
2.2.1	Conceitos básicos	15
2.2.2	Neurônio Artificial	17
2.2.2.1	Entradas	18
2.2.2.2	Pesos	18
2.2.2.3	Bias	18
2.2.2.4	Junção aditiva	18
2.2.2.5	Função de ativação	19
2.2.2.6	Saída	19
2.2.3	Tipos de Função de ativação	19
2.2.3.1	Função Limiar	19
2.2.3.2	Função Linear por Partes	20
2.2.3.3	Função Sigmoide	20
2.2.3.4	Função Tangente Hiperbólica	21
2.2.4	Camadas de RNA <i>Feedforward</i> Totalmente Conectada	21
2.2.5	Aprendizagem de Retropropagação de Erro (<i>Backpropagation</i>)	22
2.2.6	Taxa de Aprendizagem	27
2.3	Reconhecimento de padrões em imagens	28
2.3.1	Identificação de letras, números e outros símbolos	28
2.3.2	Identificação de rostos	28
2.3.3	Identificação de formas	29
2.3.4	Identificação de placas de carro	29
3	PROJETO DESENVOLVIDO	30
3.1	Materiais e Programas	30
3.2	Métodos	30
3.2.1	Obtenção dos dados de entrada	31

3.2.2	Parte 1 - PDI e segmentação	31
3.2.3	Parte 2 - Identificação utilizando redes neurais	33
4	RESULTADOS E DISCUSSÕES	36
4.1	PDI	36
4.2	RNAs	38
4.2.1	Comparação entre tipos de redes	38
4.2.1.1	Comparação entre uma rede de quatro saídas e uma rede com dez saídas	38
4.2.1.2	Comparações entre redes com variações das funções de ativação	39
4.2.1.3	Comparações entre duas RNAs com diferentes camadas ocultas	40
4.2.1.4	Variação de neurônios	40
4.2.1.5	Comparações entre RNAs com taxas de aprendizado ou passos (η) e constantes de momento (α) diferentes	41
4.2.1.6	Comparação entre conjunto de treinamento e épocas	42
4.3	Resultados das RNAs escolhidas	43
4.3.1	Resultados do treinamento da RNA 1	43
4.3.2	Resultados do treinamento da RNA 2	44
4.3.3	Generalização das RNAs	44
4.4	Sistema em conjunto	45
5	CONSIDERAÇÕES FINAIS	46
	REFERÊNCIAS	47
	APÊNDICE A – TABELA DE COMPARAÇÃO DE PARÂMETROS	49

1 INTRODUÇÃO

Aplicações utilizando técnicas de inteligência artificial tem sido cada vez mais comuns nos dias de hoje. As RNAs são estruturas que simulam algumas características do cérebro humano. Muito utilizadas para o reconhecimento de padrões em imagens, elas se encontram em diversas aplicações utilizadas frequentemente por diversas pessoas, como por exemplo, o reconhecimento de um rosto em uma foto, sendo possível a identificação de uma pessoa.

As RNAs podem ser uma ótima técnica para simular o processamento visual feito por nossos cérebros todos os dias. Apesar de uma pessoa reconhecer uma imagem comum com pouco esforço cognitivo, o reconhecimento de padrões é um trabalho complexo, tornando assim um processo difícil de ser reproduzido por um computador.

O PDI é um conjunto de técnicas computacionais que tem o objetivo de processar a imagem para sua devida finalidade. Por exemplo, pode-se converter escalas de cores, contrastes, eliminar ruídos e etc. Neste trabalho, o PDI é a primeira etapa, servindo de base para a RNA nas próximas etapas.

O reconhecimento de placas de automóveis por sistemas computacionais consiste em um conjunto de técnicas que capta a imagem da placa de um carro e identifica corretamente a informação nela contida para tomar alguma decisão. Esse sistema é dividido em algumas etapas, como obtenção da imagem pela câmera, PDI, extração dos caracteres de interesse e leitura e reconhecimento desses caracteres através de uma RNA. A velocidade e confiabilidade desse sistema são influenciados pelos *hardwares* e algoritmos utilizados.

Sistema reconhecedores de placas são amplamente difundidos no mundo, principalmente em radares de trânsito, que identificam veículos automotores em altas velocidades, acima do permitido. Como na maioria das vezes esses sistemas são soluções locais, a garantia de funcionamento correto é apenas para a região para o qual foi criado, devido aos diferentes padrões de placas em diferentes países. Na Grécia, por exemplo, enquanto a placa contém um contorno escuro e destacável, tornando a placa distinguível em carros de qualquer cor (ANAGNOSTOPOULOS et al., 2006), no Brasil as placas não possuem contorno, dificultando sua identificação em carros de cor clara (prata ou branco).

Estacionamentos que utilizam controle manual do fluxo de veículos exigem a presença de pessoas nas portarias para monitoramento da entrada e saída de automóveis, sendo necessário muitas vezes a assinatura do indivíduo que entra no local. O sistema de reconhecimento de placas de automóveis tem o objetivo de solucionar esses problemas, dispensando inspeção manual de entrada e saída de veículos, maior controle e segurança. Um controle automatizado pode auxiliar na tomada de decisões e melhoria nos procedimentos de acesso.

Além de controle de entrada de veículos, esse projeto pode servir como base para projetos diferentes, que desejam reconhecer algum tipo de placa, como de trânsito, por exemplo. Um exemplo são os projetos em andamentos de carros autônomos em diversas universidades por todo o país. Nesse tipo de carro, a presença do motorista é dispensada e o veículo passa a ser capaz de tomar decisões a partir dos eventos que acontecem no ambiente onde ele se localiza. É necessário que o sistema seja capaz de agir como um motorista humano agiria, respondendo a estímulos visuais, como parar, ao visualizar uma placa que sinalize a parada.

1.1 Objetivos

Nesta seção serão abordados os objetivos da pesquisa que será realizada.

1.1.1 Objetivo Geral

Desenvolver uma solução computacional que reconheça placas de automóveis no padrão brasileiro.

1.1.2 Objetivos Específicos

- Coletar imagens com altura e distância padrão de automóveis
- Analisar as imagens capturadas pela câmera;
- Aumentar conjunto de treinamento através de transformações das imagens utilizando técnicas de processamento digital;
- Desenvolver algoritmo de processamento digital para extrair caracteres de interesse de imagens de placas;
- Padronizar caracteres das imagens para o conjunto de treinamento das redes;
- Construir uma RNA para reconhecer números e uma rede para reconhecer letras de placas;
- Criar e definir parâmetros de configuração das redes;
- Treinar as redes com os dados das placas;
- Testar e validar as redes;
- Redefinir ou remodelar as RNAs, de acordo com os resultados dos testes.

1.2 Organização do documento

Na organização desse trabalho se encontram capítulos e documentos pós-textual. O capítulo Fundamentação Conceitual referencia trabalhos encontrados em livros e artigos científicos, que contribuíram com o desenvolvimento desse projeto. O capítulo Projeto Desenvolvido demonstra os métodos, ferramentas e materiais utilizados, possibilitando que o leitor reproduza empiricamente esse projeto. O próximo capítulo, Resultados e Discussões, demonstra os resultados desse projeto através de explicação textual e imagens de saída do código desenvolvido. No capítulo Considerações Finais é relatado a conclusão do trabalho, juntamente com as dificuldades encontradas durante o desenvolvimento do projeto.

2 FUNDAMENTAÇÃO CONCEITUAL

Nesta seção serão apresentados alguns conceitos sobre técnicas de processamento digital de imagens, além de materiais teóricos sobre RNAs. Também serão apresentados alguns trabalhos já realizados cujas técnicas unem ambas as áreas.

2.1 PDI

Com a crescente expansão da tecnologia, o uso de imagens digitais atualmente tem se tornado cada vez maior. Mídias visuais carregam um grande número de informações e seu uso tem sido explorado de diversas maneiras, desde o simples compartilhamento de fotos em redes sociais até em sistemas capazes de identificar determinadas características num grupo de imagens.

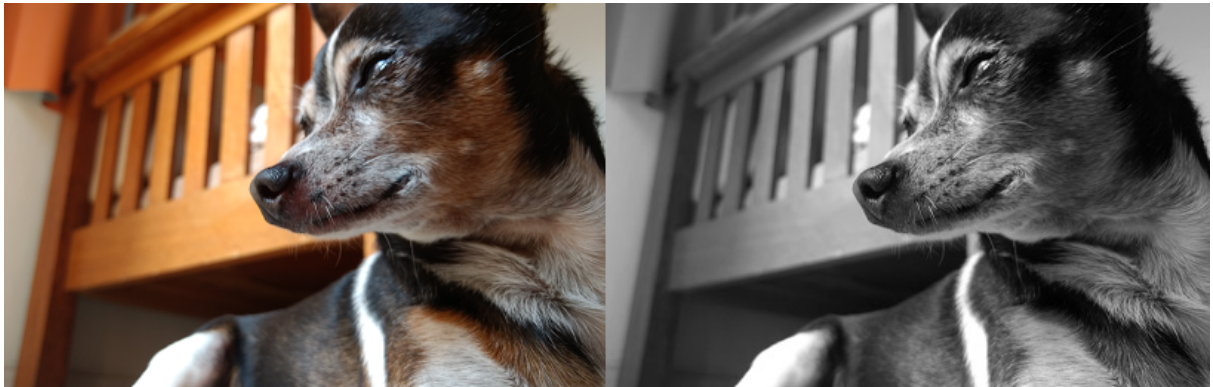
O PDI surgiu da necessidade de se alterar certos aspectos numa imagem digital afim de extrair dela informações ou apenas deixá-la mais harmoniosa e agradável aos olhos (GONZALES; WOODS, 2012). Nem sempre o dispositivo utilizado para obter a imagem reproduz a cena da maneira adequada à situação, sendo assim possível a utilização de diversas técnicas na tentativa de alcançar as características desejadas.

Sistemas computacionais que reconhecem automaticamente padrões em uma imagem fazem, em sua grande maioria, uso das técnicas de processamento de imagens para realçar as características que determinam esses padrões (GONZALES; WOODS, 2012). Um leitor biométrico, por exemplo, precisa que a imagem da impressão digital a ser analisada reproduza as linhas dos dedos, de forma que elas fiquem diferenciadas do nível mais baixo da pele. Para isso, técnicas como aumento do contraste ou limiarização podem ser utilizadas.

Grande parte dos dispositivos existentes hoje são capazes de capturar imagens com cores. Portanto, para certas situações de uso do tratamento de imagens, as cores não trazem informações relevantes, apenas aumentam o custo de processamento computacional. Nesses casos, a transformação para escala de cinzas é uma técnica eficiente. A transformação para escala de cinzas consiste em transformar uma imagem colorida para uma imagem que contém apenas tons de cinza, que variam entre o valor 0 (preto) e o valor 255 (branco). A Figura 1 representa um exemplo dessa transformação.

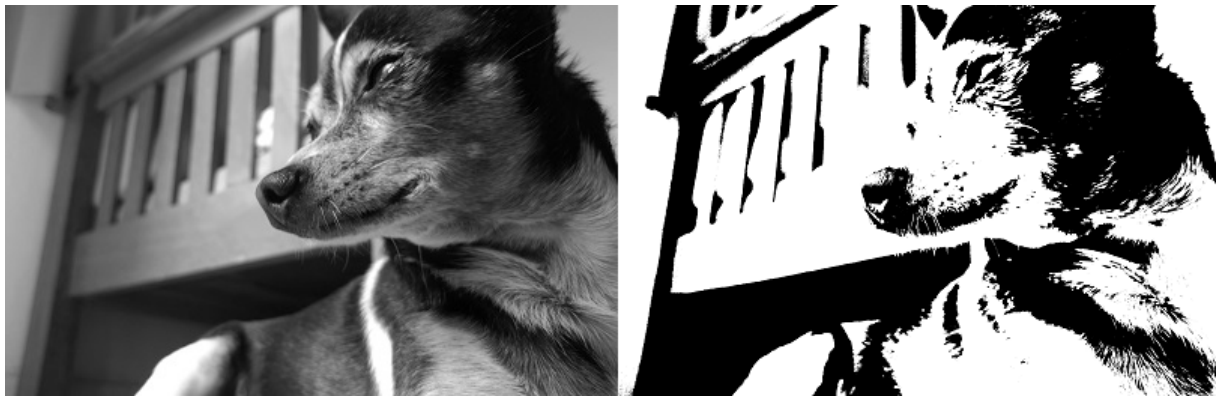
No processo de limiarização, uma imagem passa a ter apenas dois níveis de cores, usualmente preto e branco. Nesta transformação, é escolhido um valor entre 0 e 255 como limiar. Pixels com valores abaixo do limiar passam a ter um nível e pixels acima do limiar, o outro nível. A Figura 2 representa um exemplo de limiarização.

Figura 1 – Imagem original à esquerda e imagem transformada para escala de cinzas à direita



Fonte: autores, 2017.

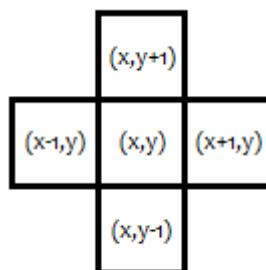
Figura 2 – Imagem original à esquerda e imagem limiarizada à direita



Fonte: autores, 2017.

Entre as técnicas de PDI há a análise de relacionamentos entre pixels. Segundo Gonzales e Woods (2012, p.26), uma vizinhança-de-4 é o nome dado ao conjunto de pixel onde um pixel central possui 4 pixels conectados horizontalmente e verticalmente a ele como é mostrado na Figura 3.

Figura 3 – Vizinhança-de-4

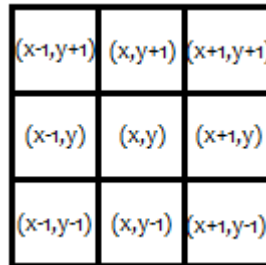


Fonte: autores, 2017.

Ainda segundo Gonzales e Woods (2012, p.26), uma vizinhança-de-8 é a vizinhança-de-4

mais os pixels vizinhos conectados diagonalmente, assim como é mostrado na Figura 4

Figura 4 – Vizinhança-de-8



Fonte: autores, 2017.

Esse tipo de análise é amplamente utilizada em algoritmos que procuram por objetos numa imagem limiarizada ou em operações de morfologia matemática como erosão e dilatação de formas.

2.2 RNAs

Um RNA é um sistema computacional que tem a capacidade de executar tarefas de forma a simular um cérebro humano.

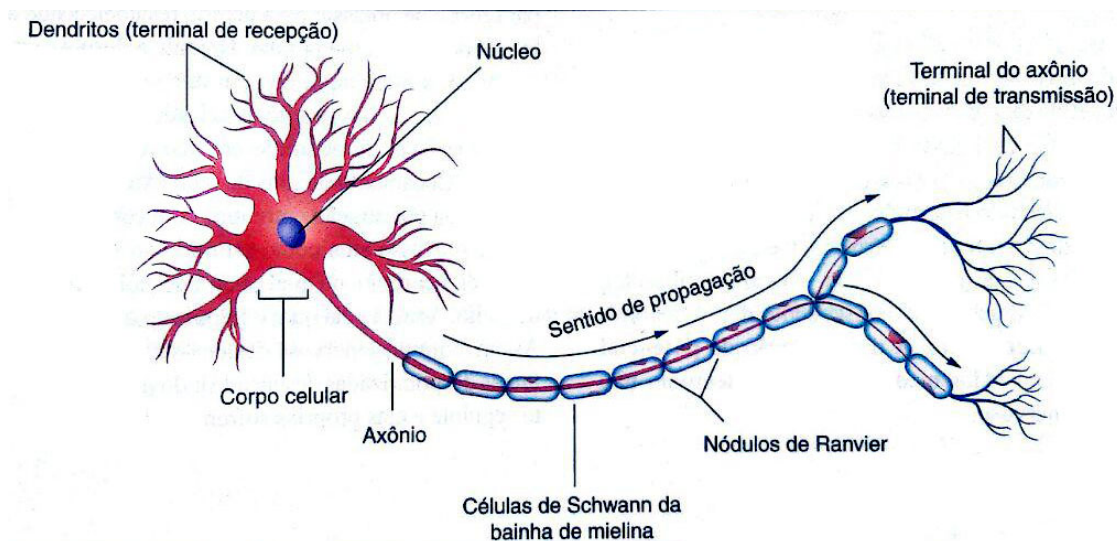
2.2.1 Conceitos básicos

As RNAs são técnicas baseadas no funcionamento do cérebro (ou seja, no funcionamento das redes neurais biológicas) numa tentativa de simular sua complexidade de processamento. Possuindo cerca de 86 bilhões de neurônios (D'ANGELO, 2016), o cérebro humano tem a capacidade de realizar diferentes tarefas todos os dias, como processar sinais ópticos ou responder a outros tipos de estímulos externos recebidos.

Os neurônios possuem uma estrutura bem característica (Figura 5) que lhes possibilitam se conectarem e formarem uma rede. Suas principais partes são o axônio, os dendritos e o núcleo celular. Os dendritos são responsáveis por receber sinais vindos de outros neurônios e o seu núcleo faz o processamento necessário. Os axônios enviam o sinal processado para os próximos neurônios. Conexões entre neurônios são chamadas de sinapses.

Um conjunto de neurônios interligados forma uma RNA. Essa rede é capaz de adquirir experiência ao longo do tempo, durante um processo chamado **aprendizagem**. Uma pessoa é capaz de reconhecer um rosto familiar no meio de uma multidão, pois ela já viu essa face algumas vezes em algum momento anterior e aprendeu sobre as suas características. O mesmo

Figura 5 – Partes de um neurônio.



Fonte: (BRITO, 2015)

funciona para o reconhecimento de um cheiro, para resolver um problema matemático ou para a escrita.

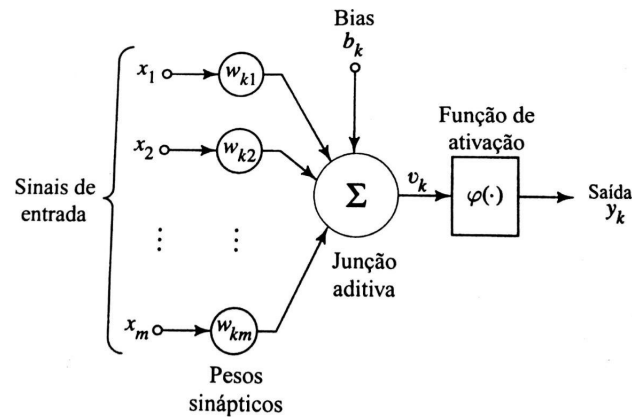
Segundo Haykin (2001, p. 28), uma RNA pode ser definida como um processador paralelo e distribuído por possuir unidades simples que realizam processamento (neurônios artificiais) que possuem a capacidade de armazenar conhecimento adquirido com experiências, tornando-se disponível para uso. As RNAs têm como objetivo simular algumas propriedades do cérebro, como a não-linearidade e o paralelismo no processamento, além de serem capazes de aprender.

O aprendizado numa RNA faz com que ela adquira a experiência necessária para que possa posteriormente responder a entradas desconhecidas. A rede é treinada com entradas e saídas conhecidas. Este treino modifica os pesos sinápticos da rede fazendo com que ela se aproxime do objetivo desejado (HAYKIN, 2001).

Um neurônio artificial pode ser modelado como uma função aditiva que recebe uma ou mais entradas, envia a saída para uma função de ativação que fornece o resultado final do neurônio (Figura 6). Essa saída poderá ser uma entrada para um próximo neurônio. As RNAs possuem diversas modelagens diferentes, porém todas utilizam neurônios e necessitam passar por um processo de aprendizagem.

As redes possuem propriedades como adaptabilidade e tolerância à falhas que as fazem ideais para determinadas situações. A primeira é a possibilidade da rede trabalhar em ambientes que podem sofrer pequenas alterações ao longo do tempo enquanto a segunda pode ser aplicada a redes implementadas fisicamente, pois caso aconteça uma perda em algum de seus neurônios e sinapses, a rede pode continuar obtendo resultados satisfatórios, porque a informação aprendida

Figura 6 – Modelo de um neurônio artificial.



Fonte: (HAYKIN, 2001).

está distribuída por todo o conjunto de neurônios (HAYKIN, 2001).

Para que uma RNA possa ser corretamente modelada, é necessário o processo de aprendizagem. Este processo pode ser definido pela adaptação dos parâmetros livres da rede utilizando estimulação do ambiente onde ela está inserida. A maneira como estes parâmetros são alterados define o tipo de aprendizado que está sendo usado para que a rede adquira o conhecimento necessário (HAYKIN, 2001).

Uma das formas de aprendizado mais utilizadas no treinamento de redes neurais é a aprendizagem supervisionada em que, segundo Russell (2004, p.630), "envolve a aprendizagem de uma função a partir de exemplos de suas entradas e saídas". Dentre as formas de aprendizado supervisionado, a aprendizagem por retropropagação (também conhecida como *backpropagation*) é uma das formas mais eficientes de treinamento, pois ela utiliza o erro para fazer as devidas alterações sinápticas. Nesse treinamento a saída obtida pela rede é comparada com a saída desejada, resultando em um valor de erro, o que possibilita alterações nos pesos sinápticos (HAYKIN, 2001).

2.2.2 Neurônio Artificial

Um neurônio, exibido na Figura 6, possui entradas, pesos, uma junção aditiva, uma função de ativação e uma saída. Os sinais de entrada fornecem os dados de interesse para a rede (HAYKIN, 2001). A seguir são exibidos cada componente de um neurônio.

2.2.2.1 Entradas

As entradas fornecem os dados de interesse para a rede. Podem consistir, por exemplo, de pixels de uma imagem. Para cada neurônio, existe uma entrada chamada bias, cujo valor, é definido pelo projetista da máquina de aprendizagem.

2.2.2.2 Pesos

Um peso sináptico é ligado entre uma entrada e uma saída de um neurônio. Assim, por exemplo, se um neurônio, em qualquer camada, possui dez entradas e apenas uma saída, ele terá dez pesos. O valor ajustado no peso pela rede define sua aprendizagem local. O conjunto de ajustes de todos pesos da rede define a aprendizagem global.

2.2.2.3 Bias

O bias possibilita um neurônio apresentar uma saída não-nula. Um exemplo da importância do bias seria se todas as entradas de um rede fossem nulas. Dessa forma, se a função de ativação fosse, por exemplo, uma tangente hiperbólica, mostrada na seção 2.2.3, a saída seria nula. Assim, como exemplo, a rede não funcionaria para o problema de lógica booleana "ou exclusivo", com a relação $x = [0 \ 0]^T$ e $y = 1$ (JR.; MONTGOMERY, 2007, p.14).

2.2.2.4 Junção aditiva

A função aditiva é o núcleo do neurônio, responsável por fazer o somatório do produto da entrada pelo peso correspondente, conforme demonstrado na Figura 6. A saída da função aditiva é o campo induzido, representado pela equação 2.1.

$$v_k = \sum_{j=1}^m w_{kj}x_j + b_k \quad (2.1)$$

Onde:

v_k = campo induzido do neurônio k

w_{kj} = peso sináptico ligado a uma entrada j e saída do neurônio k

x_j = entrada j

b_k = bias do neurônio k

2.2.2.5 Função de ativação

A função de ativação funciona como um amplificador ou atenuador do sinal do campo induzido, a qual é discutida detalhadamente na seção 2.2.3.

2.2.2.6 Saída

A saída fornece a resposta da rede para determinados estímulos ou entradas. O seu valor pode variar entre -1 e 1, dependendo da função de ativação utilizada e do somatório do campo induzido. A saída de um neurônio é representada pela equação 2.2.

$$y_k = \varphi(v_k) \quad (2.2)$$

Onde:

y_k = saída do neurônio k

$\varphi(v_k)$ = função de ativação do neurônio k

v_k = campo induzido do neurônio k (somatório das entradas)

2.2.3 Tipos de Função de ativação

A função de ativação define a saída de um neurônio, a qual pode variar entre -1 e 1, dependendo da função transferência utilizada e do resultado do campo induzido v_k , o qual corresponde ao somatório do produto dos pesos sinápticos pelas entradas, conforme ilustra a Figura 6. Assim a função de ativação restringe ou limita a amplitude do sinal de saída a um valor finito. As quatro principais funções de ativação são: Limiar, Linear por Partes, Sigmóide e Tangente Hiperbólica (HAYKIN, 2001, p.38-41, p.37,p.206-207).

2.2.3.1 Função Limiar

Neste modelo, usado para sistemas lineares, a saída do neurônio assume o valor 0 quando a saída induzida v_k é negativa, e assume 1 caso contrário, conforme a equação 2.3.

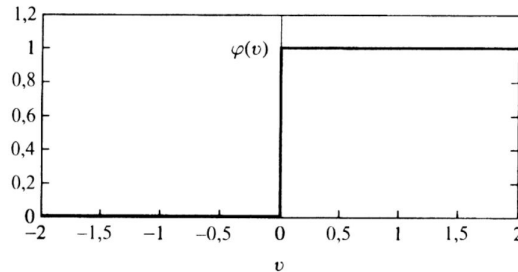
$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (2.3)$$

Onde:

$\varphi(v)$ = função de ativação

O gráfico da função Limiar corresponde à uma função degrau, o que é demonstrado na Figura 7.

Figura 7 – Gráfico da função limiar.



Fonte: (PERALES, 2011, p.26).

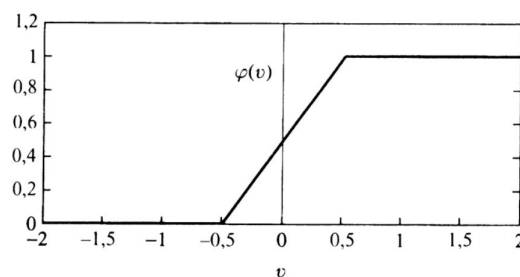
2.2.3.2 Função Linear por Partes

Corresponde à uma aproximação de um amplificador não linear, demonstrado na equação 2.4.

$$\varphi(v) = \begin{cases} 1, & v \geq \frac{1}{2} \\ v, & \frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (2.4)$$

O seu gráfico é uma função linear e poder ser visto na Figura 8.

Figura 8 – Gráfico da função linear saturada.



Fonte: (PERALES, 2011, p.26).

2.2.3.3 Função Sigmoide

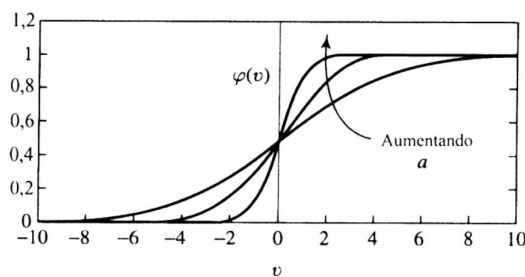
Quando se constrói uma rede com mais de uma camada, o ajuste dos pesos será resolvido por uma técnica chamada Descida de Gradiente em busca do menor erro global, o que será retratado detalhadamente na seção 2.2.5. Se a função de transferência fosse linear, a derivada seria constante e os valores dos pesos não seriam atualizados (JR.; MONTGOMERY, 2007,

p.13). A sigmoide é a função de ativação mais utilizada em RNAs para sistemas lineares e não-lineares (HAYKIN, 2001, p.40). A função de transferência da sigmoide é mostrada na equação 2.5.

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (2.5)$$

O gráfico que ilustra a equação 2.5 está disposto na forma de S e sua inclinação depende do fator de inclinação a , o que é exibido na Figura 9.

Figura 9 – Gráfico da função sigmoide.



Fonte: (PERALES, 2011, p.26).

2.2.3.4 Função Tangente Hiperbólica

Essa função de ativação corresponde à uma sigmoide anti-simétrica, ou seja, que varia entre -1 e 1. Duas vantagens sobre a sigmoide é que possui maior intervalo de valores e permite um treinamento da RNA mais rápido, o que pode ser mais útil para determinados tipos de problemas. A função de transferência da tangente hiperbólica é mostrada na equação 2.6 (HAYKIN, 2001, p.40, p.206).

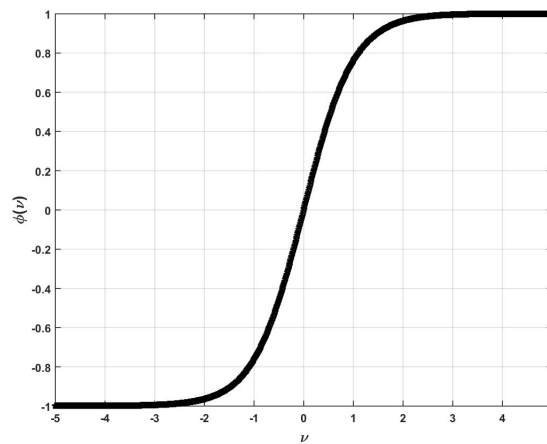
$$\varphi(v) = \tanh(v) \quad (2.6)$$

A equação 2.6 é exibida graficamente na Figura 10.

2.2.4 Camadas de RNA *Feedforward* Totalmente Conectada

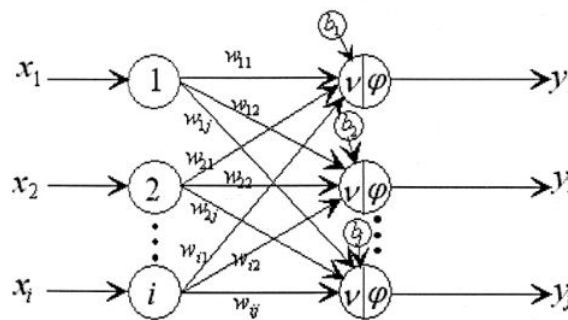
O termo *Feedforward* se refere ao sinal que segue da entrada em direção à saída de uma RNA. O termo totalmente conectada está relacionado aos nós de uma camada de rede ligado a todos os nós adjacentes. Uma rede parcialmente conectada possui a ausência de alguns elos de comunicação. Uma RNA de camada única, usada para resolver problemas de natureza simples, possui uma camada de entrada e uma de saída. Apesar de ter duas camadas, a camada de entrada não é contada por não realizar qualquer computação. Na Figura 11 é demonstrado uma rede *feedforward* totalmente conectada de camada única (JR.; MONTGOMERY, 2007, p.27-29).

Figura 10 – Gráfico da função tangente hiperbólica.



Fonte: autores, 2017

Figura 11 – Rede neural com uma camada.



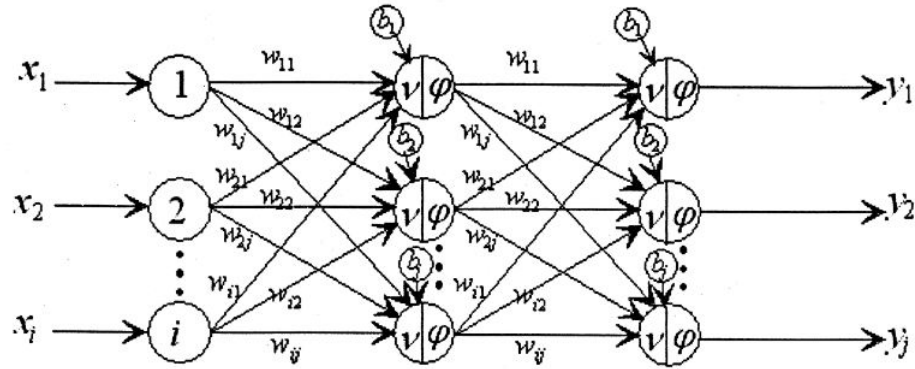
Fonte: (JR.; MONTGOMERY, 2007, p.28).

Uma RNA Perceptron de Múltiplas Camadas ou *MultiLayer* Perceptron (MLP) possui uma ou mais camadas ocultas, que ficam entre a camada de entrada e camada de saída. Esse tipo de rede é usado geralmente para resolver problemas complexos. Na Figura 12 é exibido uma MLP de apenas uma camada oculta (JR.; MONTGOMERY, 2007, p.45-46).

2.2.5 Aprendizagem de Retropropagação de Erro (*Backpropagation*)

O algoritmo de retropropagação é uma técnica usada em redes MLP que implementa uma otimização por descida de gradiente. Esse algoritmo calcula o menor erro no espaço de pesos, realizando as atualizações dos pesos da saída em direção à entrada, através de um processo de retropropagação. A tendência é que o erro diminua gradativamente a cada época (ciclo de treinamento), até que seja pequeno suficiente para que a rede esteja apta a desempenhar as tarefas para as quais foi criada e os pesos se tornem estabilizados. O erro local (apenas um neurônio) é gerado por meio da diferença entre a saída desejada apresentada a rede pelo projetista e a saída

Figura 12 – Rede neural MLP



Fonte: (JR.; MONTGOMERY, 2007, p.46).

real de um neurônio da camada de saída de uma RNA, conhecido como método Supervisionado. A equação do erro local é demonstrada na equação 2.7 (MEDEIROS, 2003, p.13-16).

$$e_j(n) = d_j(n) - y_j(n) \quad (2.7)$$

Onde:

$e_j(n)$ = erro do neurônio j para um exemplo n no conjunto de treinamento, em uma época

$d_j(n)$ = saída desejada do neurônio j da camada de saída

$y_j(n)$ = saída real do neurônio j da camada de saída

A Figura 13 exibe a equação 2.7 em forma de grafo, em que d_j representa a saída desejada, e_j é o erro e y_i representa a saída do neurônio anterior e entrada para o neurônio j da camada de saída.

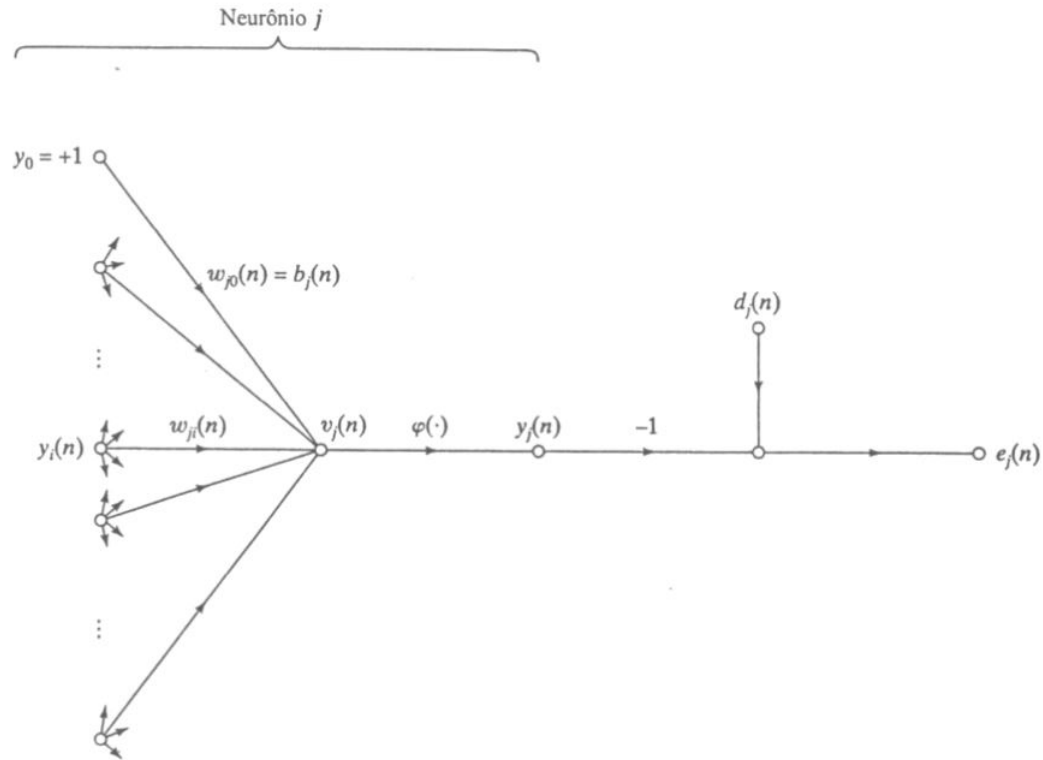
Conhecida como Regra Delta, baseando-se no erro, os ajustes e correções dos pesos locais são feitos através da função de custo instantâneo local, índice de desempenho instantâneo local ou energia instantânea de erro local ($\varepsilon_j(n)$), o qual é calculado para apenas um neurônio da camada de saída da rede, o que é demonstrado na equação 2.8 (HAYKIN, 2001, p.77).

$$\varepsilon_j(n) = \frac{1}{2} e_j^2(n) \quad (2.8)$$

Onde:

$\varepsilon_j(n)$ = custo local do neurônio j da camada de saída

A soma da energia instantânea de erro local de todos os neurônios de saída da rede para

Figura 13 – Grafo de fluxo de sinal ressaltando os detalhes do neurônio e saída j 

Fonte: (PERALES, 2011, p.16).

cada exemplo n em uma época, resulta na energia instantânea de erro global para cada exemplo n , demonstrado na equação 2.9.

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2.9)$$

Onde:

$\varepsilon(n)$ = energia instantânea de erro global para cada exemplo, dentro de uma época

C = conjunto de todos os neurônios da camada de saída da rede

A energia média de erro global (ε_{med}), que inclui todos os N exemplos de um conjunto de treinamento dentro de uma época para todos os neurônios da camada de saída, utiliza-se a equação 2.10.

$$\varepsilon_{\text{med}} = \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \quad (2.10)$$

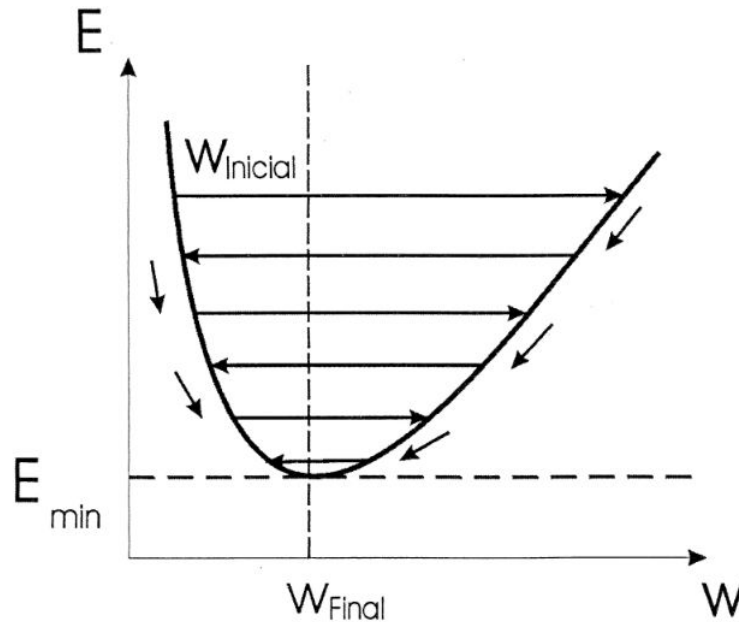
Onde:

ε_{med} = energia média de erro global

N = todos os exemplos ou amostras do conjunto de treinamento de uma RNA

O erro de cada neurônio é elevado ao quadrado (equação 2.8) para evitar que um erro negativo compense um erro positivo, o que criaria um erro falso. O mais importante para o custo global é a distância euclidiana entre saída desejada e saída real. Na Figura 14 é representada graficamente o processo da descida de gradiente para encontrar o energia média de erro global (ε_{med}) mínima, a qual possui seus valores no eixo das ordenadas. Os valores dos pesos, encontrados no eixo das abcissas, são ajustados como consequência da descida do gradiente (JR.; MONTGOMERY, 2007, p.47).

Figura 14 – Representação da convergência do valor do peso, começando com um valor peso inicial para chegar a um valor peso final. As setas ao lado da curva representam a direção do gradiente dando o sentido do mínimo erro global



Fonte: (MEDEIROS, 2003, p.15).

Conhecida como regra delta, a correção dos pesos locais são realizadas através da derivada do custo global em relação ao peso local, sendo realizado em uma velocidade que depende da taxa de aprendizagem e utilizando um sinal negativo para indicar a busca de um mínimo global. Esse processo é demonstrado na equação 2.11, para cada amostra n em uma época.

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} \quad (2.11)$$

Onde:

$\Delta w_{ji}(n)$ = correção de um peso local (entre um neurônio i e um neurônio j) para cada amostra n em uma época

η = taxa de aprendizagem

sinhal negativo = indica descida de gradiente no espaço de pesos para redução de $\varepsilon(n)$

A regra delta da equação 2.11 produz a equação 2.12.

$$\Delta w_{ji}(n) = \eta \delta_j y_i(n) \quad (2.12)$$

Onde:

δ_j = gradiente local do neurônio j

y_i = saída do neurônio da camada anterior e entrada i para o neurônio j

Quando o neurônio j se encontra na camada de saída da rede, o gradiente local é calculado através da equação .

$$\delta_j = e_j(n) \varphi'(v_j(n)) \quad (2.13)$$

Onde:

$v_j(n)$ = saída induzida do neurônio j para n amostras

$\varphi'(\cdot)$ = função de ativação diferenciável em relação $v_j(n)$

Por outro lado, quando o neurônio j está em uma camada oculta, utiliza-se a equação .

$$\delta_j(n) = \varphi'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (2.14)$$

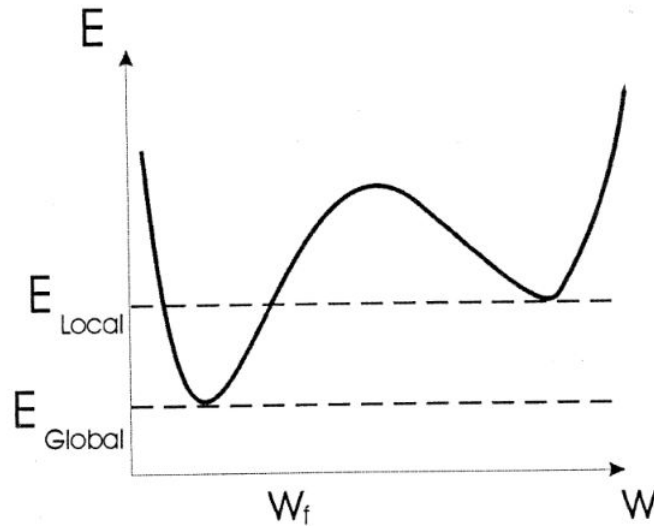
Onde:

$\delta_j(n)$ = gradiente local do neurônio j da camada oculta

$\delta_k(n)$ = gradiente local de um neurônio k da camada adiante

É desejável que a rede convirja para o menor erro possível, conhecido como erro global. Porém pode acontecer, a depender da natureza do problema, que o gradiente encontre o mínimo local, o que é influenciado pela taxa de aprendizado. Esse problema é ilustrado na Figura 15.

Figura 15 – Gráfico Erro x Peso demonstrando a presença de um mínimo local além de um mínimo global na superfície de erro. Dependendo da taxa de aprendizagem, o algoritmo da descida de gradiente pode ficar preso no mínimo local



Fonte: (MEDEIROS, 2003, p.16).

2.2.6 Taxa de Aprendizagem

A taxa de aprendizagem ou simplesmente passo de aprendizagem, representada por η , fornece a velocidade com a que os pesos sinápticos são ajustados, e consequentemente, a velocidade em que uma RNA aprende. Quanto menor os passos, menor será o ajuste dos pesos entre as iterações e, assim, mais suave será a trajetória do espaço de pesos. Por outro lado, se o passo da rede for grande, a rede aprenderá de forma mais rápida, porém as grandes modificações dos pesos em cada iteração pode tornar a rede instável, como exemplo oscilatória. Para evitar a instabilidade da rede, utiliza-se o termo constante de momento (α) demonstrado na equação 2.15, o qual é um parâmetro da RNA cujo o valor varia entre 0 e 1, igualmente à taxa de aprendizagem. (HAYKIN, 2001, p.196-198).

$$\Delta w_{ji} = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad (2.15)$$

Onde:

Δw_{ji} = correção de peso local

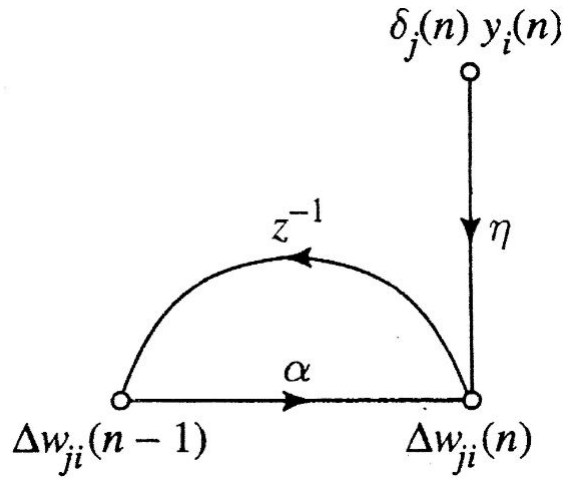
η = taxa de aprendizagem

δ_j = gradiente local que minimiza o erro local

y_i = saída do neurônio da camada anterior e entrada i para o neurônio j

A equação 2.15 é demonstrada em forma de grafo na Figura 16, cujo Z^{-1} representa atraso unitário.

Figura 16 – Grafo de fluxo de sinal ilustrando o efeito da constante de momento α .



Fonte: (HAYKIN, 2001, p.197).

2.3 Reconhecimento de padrões em imagens

Muito utilizadas para reconhecimento de padrões em imagens, as RNAs muitas vezes possuem alta eficiência na identificação de caracteres de texto, símbolos, objetos específicos e até mesmo reconhecimento facial em imagens. Nas próximas subseções serão apresentados trabalhos já realizados em algumas dessas áreas.

2.3.1 Identificação de letras, números e outros símbolos

Yang et al. (2016) apresentou uma RNA capaz de fazer o reconhecimento de caracteres chineses escritos à mão. Com a utilização de uma rede neural convolucional, os resultados obtidos tiveram uma média de 97% de eficiência.

Guingo, Thomé e Rodrigues (2002) desenvolveram um trabalho de reconhecimento automático de placas de veículos automotores brasileiros. O problema foi dividido em 6 partes, desde o a captura da imagem até o reconhecimento de cada um dos caracteres que compõem a placa. O desempenho foi abaixo do esperado devido ao uso de câmeras que capturam imagens de baixa qualidade (eficiência próxima de 71%).

2.3.2 Identificação de rostos

Babu et al. (2015) apresentou um projeto utilizando a técnica de *backpropagation* no aprendizado de uma RNA e o *software* Matlab para identificar rostos de pessoas criminosas. O *hardware* para captar e processar a imagem foi uma câmera simples e um computador pessoal, possibilitando que o sistema seja usado por pessoas com menor poder aquisitivo. Segundo o

autor, a parte mais crítica do reconhecimento de face foi a extração de características de imagens, pois o sistema deveria captar características únicas no rosto das pessoas para diferenciá-las.

2.3.3 Identificação de formas

Segundo Neubauer (1998) em seu projeto “Avaliação de Rede Neural Convolucional para Reconhecimento Visual” que tem o intuito de reconhecer faces humanas e dígitos numéricos manuscritos, uma rede convolucional é completamente conectada, tem maior paralelismo neural, é menos suscetível à falhas e tem maior precisão comparada à uma rede não convolucional. Em seus diversos experimentos e resultados, obteve-se reconhecimento correto de até 87,6% em alguns casos.

2.3.4 Identificação de placas de carro

Tabrizi e Cavus (2016) mostram uma solução para o reconhecimento de caracteres de placas de carros utilizando técnicas de detecção de bordas, funções da morfologia matemática e filtragens, obtendo 97% de acerto na identificação.

Anagnostopoulos et al. (2006) elaboraram uma solução para a identificação de caracteres de placas de carros da Grécia utilizando o algoritmo SCW, técnica que utiliza janelas concêntricas deslizantes que percorrem a imagem procurando pela placa do carro (caracterizada por uma irregularidade na imagem), métodos de binarização e finalmente uma rede neural para identificação dos caracteres segmentados obtendo entre 80% e 90% de acerto (os valores variam de acordo com os parâmetros utilizados).

Rodrigues (2002) apresentou um modelo de identificação de placas de trânsito a partir de dados capturados por uma câmera num veículo em movimento. Foi utilizado um módulo de localização e um módulo de classificação que localizava a placa em cada quadro capturado e classificava o tipo de placa, respectivamente. As RNAs se mostraram eficientes para a realização do objetivo proposto, porém os resultados tiveram certas limitações devido à limitação recursos disponíveis.

Ciresan et al. (2012) apresentou uma técnica para reconhecimento das placas de trânsito que apresentou eficácia de mais de 98%. Foram utilizados filtros sobre as imagens capturadas e uma rede neural de multicamadas.

3 PROJETO DESENVOLVIDO

Neste capítulo serão apresentados os métodos de pesquisa, ferramentas e materiais utilizados.

3.1 Materiais e Programas

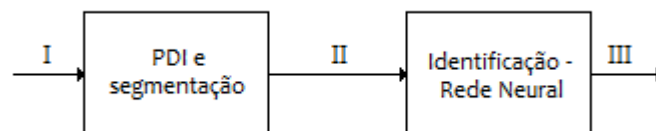
Nesta seção serão apresentados os recursos utilizados, os quais foram adquiridos previamente, não incluindo custos para esse projeto.

- Computadores pessoais;
- *Software* do Sistema Operacional Microsoft Windows;
- *Software* Matlab;
- Smartphone com câmera fotográfica com resolução de 8 megapixels

3.2 Métodos

O projeto apresentado consiste em duas partes principais, representadas pela Figura 17. A primeira parte consiste no processamento da imagem de entrada (parte frontal do carro), tendo como resultado os caracteres da placa segmentados. A segunda parte faz a identificação de cada um dos caracteres, atribuindo-lhes seus respectivos rótulos.

Figura 17 – (I) - Imagem de entrada; (II) - Caracteres segmentados; (III) - Rótulo de cada caracter



Fonte: o autor, 2017

Cada uma das partes possui seus padrões de entrada e saída, que serão discutidos nas próximas subseções.

3.2.1 Obtenção dos dados de entrada

As imagens que compõem a base de dados foram capturadas por um smartphone com câmera com capacidade de resolução de 8 megapixels. Para ampliar a quantidade de imagens na base de dados, foram utilizadas funções do MatLab que geram ruídos nas imagens. Alguns dos ruídos utilizados foram gaussiano, multiplicativo, *salt and pepper* e embaçamento.

3.2.2 Parte 1 - PDI e segmentação

O sistema recebe como entrada uma foto no formato *.jpg* com 490 pixels de largura por 368 pixels de altura. O dispositivo de captura da imagem foi posicionado a aproximadamente à 1 metro de distância da parte frontal do carro e 60 cm de altura. Todas as imagens foram captadas pelos próprios autores por meio de um smartphone.

A imagem capturada é colorida e possui 3 matrizes de dados correspondentes aos seus tons de vermelho, verde e azul. Transforma-se a imagem para a escala de cinzas, reduzindo assim para apenas uma matriz de dados, com cada pixel em um tom de cinza que varia de 0 a 255. Essa imagem então é limiarizada com o valor de 40.

Após a limiarização, espera-se que os caracteres se destaquem do fundo da placa como mostra o exemplo da Figura 18.

Figura 18 – Exemplo de caracteres destacados após limiarização



Fonte: autores, 2017

Com a imagem binária, é aplicada uma análise de conectividade de pixels, procurando por conjuntos de pixels de valor 1 (branco) unidos por vizinhança-de-8. Cada um desses conjuntos é tratado como um objeto envolvido por pixels de valor 0 (preto). Diversos objetos são encontrados,

porém apenas 7 deles são os objetos de interesse, sendo necessária então a utilização de filtros para eliminar aqueles que não contêm as letras e números da placa de identificação do automóvel.

O primeiro filtro utilizado é o de quantidade de pixels. Objetos com mais de 450 e menos de 50 pixels já são eliminados dos objetos candidatos a serem os de interesse pois, devido a distância entre a placa e a câmera, os caracteres sempre possuem uma quantidade de pixels entre esses valores. Esses valores foram definidos por meio de tentativa e erro.

O segundo filtro é uma análise de alinhamento horizontal e de altura dos objetos. Sabe-se que em uma placa de identificação, os caracteres se encontram todos alinhados, portanto objetos que estejam desalinhados e/ou possuem alturas diferentes são eliminados dos objetos candidatos. No momento da comparação de alinhamento há uma margem de erro de 10 pixels para que leves angulações na placa sejam consideradas além de possíveis ruídos.

O terceiro e último filtro procura por objetos cujos centroides sejam mono-espaçados, com exceção da terceira letra e do primeiro número, pois são separados por um hífen. O padrão atual de placas nacionais utiliza o mono espaçamento para os caracteres, como pode ser visto na Figura 19. No momento da comparação das distâncias entre os centroides foi utilizada uma margem de erro de 3 pixels para que objetos com ruídos não sejam excluídos.

Figura 19 – Caracteres das placas nacionais são mono-espaçados



Fonte: Desconhecido (2017)

Após o terceiro filtro, os 7 objetos de interesse são detectados. Cada um dos caracteres é centralizado numa imagem que contém largura igual a 20 pixels e altura igual a 20 pixels. Isso é feito completando a largura ou a altura (a dimensão que for menor) com pixels pretos e depois redimensionando a imagem para o valor desejado (20x20). Logo após, essa imagem é transformada para uma matriz coluna de 400 posições como padrão para entrada da RNA, a qual faz a identificação dos caracteres segmentados na segunda parte do sistema.

Em alguns casos onde o nível de iluminação no ambiente era alto, o limiar de valor 40 escolhido não resultava no realce dos caracteres sobre o fundo da placa. Nesse caso, o algoritmo incrementa em 10 o valor do limiar, retornando ao processo de limiarização. Isso é repetido até que o realce entre caracteres e fundo da placa aconteça ou o limiar alcance seu nível máximo. Este valor máximo é de 245, pois com o incremento de 10 ele se torna o maior tom que um pixel pode ter (255 - branco).

Quando o valor máximo do limiar é atingido e o realce entre caracteres e fundo da placa ainda não aconteceu, o algoritmo falha pois ele não foi capaz de encontrar os objetos de interesse.

3.2.3 Parte 2 - Identificação utilizando redes neurais

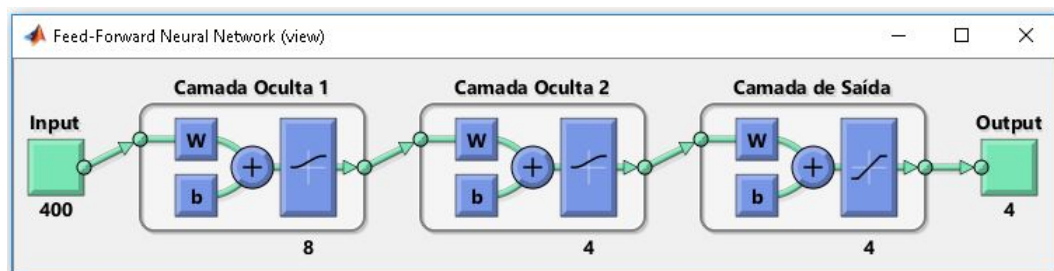
Para a segunda parte foram utilizadas duas RNAs, uma para fazer a identificação dos números e outra para fazer a identificação das letras. Esta escolha foi feita para eliminar a confusão na identificação feita pela rede nos casos dos números 0 e 1 e das letras O e I, pois, devido à fonte dos caracteres, o número 0 (zero) e a letra O são idênticos, assim como o número 1 e a letra I. As duas RNAs foram escolhidas baseando-se em testes empíricos conforme

Tabela 1 – Características das RNAs

Características	RNA 1	RNA 2
Entradas (pixels)	400	400
Conexão entre neurônios	total	total
Valor inicial de pesos	0	0
Valor de bias	1	1
Número de camadas ocultas	2	2
Função de ativação por camada	sigmoide-sigmoide-linear	sigmoide-sigmoide-linear
Neurônios por camada	8 - 4 - 4	10 - 5 - 5
Saídas	4	5
Taxa de aprendizado (η)	0.5	0.005
Momento (α)	0.001	0.005
Tempo de treino (hs)	13	2.3
Conjunto de treinamento	4480	2400
Épocas	600	200
Entradas de amostras	aleatório	aleatório

demonstrado no capítulo 4, resultados e discussão. Na Tabela 1 é demonstrado as características da RNA 1 (números) e RNA 2 (letras) em relação a estrutura e treinamento. Nas Figuras 20 e 21, tem-se, respectivamente, as estruturas básicas das RNAs para números e para letras.

Figura 20 – Estrutura básica da RNA 1

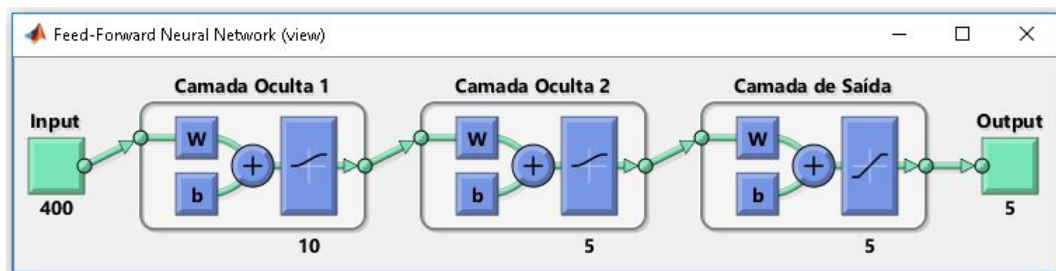


Fonte: autores, 2017

Para a realização do treinamento, os caracteres já segmentados passaram por um novo procedimento de geração de ruídos no intuito de aumentar ainda mais o conjunto de treinamento.

A quantidade total de amostras após utilização de ruídos nos caracteres segmentados foram 4480 para números e 2400 para letras. Uma observação importante é que quantidade de caracteres obtidos no treinamento de letras foi menor do que de números, consequentemente fazendo o conjunto de treinamento de letras ser menor. E no treinamento da RNA 2 foram usadas apenas 200 épocas porque se observou que no treinamento da RNA 1 após essa quantidade de ciclos de treinamento o erro médio quadrado (ε_{med}) não diminuía mais, ou seja, a RNA cessava o aprendizado.

Figura 21 – Estrutura básica da RNA 2

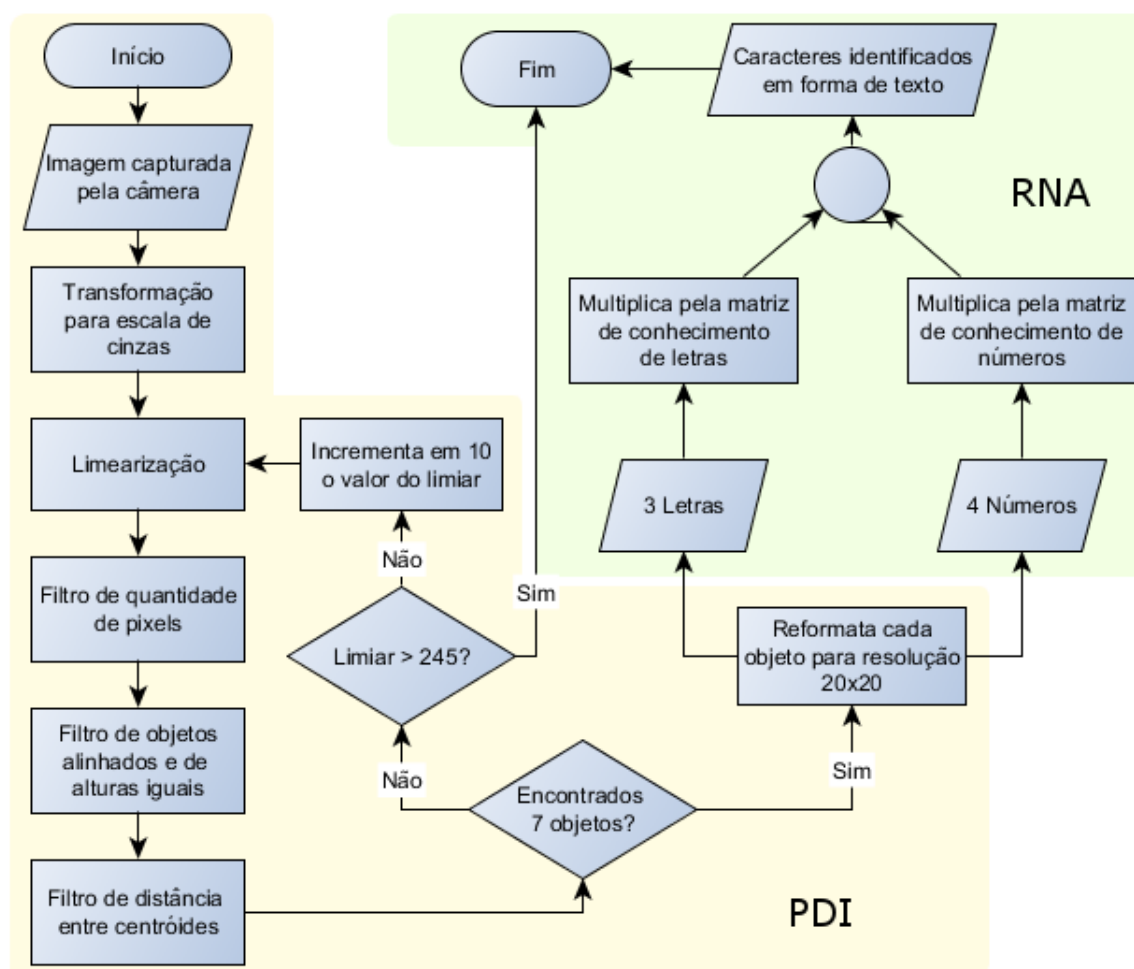


Fonte: autores, 2017

Na Figura 22 é possível visualizar o fluxograma do sistema indicando todos os passos descritos. No bloco PDI, o sistema se inicia na entrada da imagem da frente do carro, que foi capturada pela câmera. Esta imagem é transformada para a escala de cinzas e limerizada com o primeiro valor de limiar (40). Com a imagem em dois níveis de cores (preto e branco) são encontrados os objetos existentes na imagem, já passando pelo primeiro filtro que elimina objetos muito grandes ou muito pequenos. O segundo filtro elimina objetos que não estejam alinhados e não satisfazem às condições de proximidade de altura e o terceiro filtro elimina os objetos que não possuem centroides mono-espaçados. Se após os 3 filtros os 7 caracteres não tiverem sido encontrados, retorna-se ao processo de limiarização com o valor de limiar incrementado (desde que o valor máximo ainda não tenha sido atingido). Quando os 7 objetos são encontrados, eles são reformatados para uma imagem 20x20.

No Bloco RNA, os caracteres são separados em letras e números, sendo enviado as letras para a RNA que reconhece letras e o segundo para a RNA que reconhece números. Cada letra é convertida para uma matriz coluna e multiplicada pela matriz de pesos da rede de letras. O mesmo é feito para cada um dos números, na RNA de números. O resultado dessa multiplicação é convertido para a representação em texto correspondente, gerando assim a saída do sistema: a placa identificada.

Figura 22 – Fluxograma do sistema



Fonte: autores, 2017

4 RESULTADOS E DISCUSSÕES

Nesta seção serão discutidos os resultados obtidos no projeto desenvolvido. Os resultados serão demonstrados para a parte de PDI, para as RNAs e para o sistema em conjunto.

4.1 PDI

É possível ver na Figura 23 o resultado do PDI realizado em uma das imagens adquiridas. Na Figura 23(a), encontra-se a imagem de entrada em sua forma original com cores. Na 23(b),

Figura 23 – Exemplo de resultado do PDI para uma das imagens de entradas



(a) Imagem original



(b) Imagem binária



(c) Caracteres segmentados

já foram realizados os tratamentos de transformação para escala de cinzas e limiarização com o valor de limiar que faz os caracteres se destacarem do fundo da placa. Na 23(c) encontram-se os caracteres encontrados após passarem pelos três filtros, já segmentados e centralizados numa imagem 20x20, padronizados para a entrada na RNA.

Figura 24 – Exemplos de sucesso do PDI



Fonte: autores, 2017

Durante o processo de aquisição de imagens foram capturadas 28 imagens, 7 foram separadas para o processo de generalização e 21 para desenvolvimento e testes do PDI e

treinamento das RNAs. Com o objetivo de aumentar a quantidade de imagens para a realização dos testes no PDI, estas 21 imagens passaram por processos de adição de ruído, aumentando assim para 147 imagens. Os caracteres, depois de segmentados passaram por novos ruídos para aumentar o número de amostras que seriam utilizadas apenas no treinamento das RNAs.

Nas 147 imagens de placas de carro o PDI foi capaz de encontrar os 7 objetos de interesse em 78% delas. O tempo médio de execução do processamento para cada uma foi de 0,7321 segundos, sendo que o menor tempo registrado foi de 0,6719 segundos e o maior foi de 1,3750 segundos. É possível observar na Figura 24 outros exemplos de sucesso na utilização do PDI.

Em algumas das placas os objetos de interesse foram encontrados porém com perda de informação devido à distribuição de luz sobre ela não ser uniforme. Nestes casos a utilização de um valor de limiar global não é o ideal, ocasionando perda de informação como pode ser visto na Figura 25 (b) onde o número 9 perdeu parte de sua informação. Na maioria dos casos em que isso ocorreu a RNA ainda obteve êxito na identificação. Em outras, a perda de informação do caractere foi demasiada a ponto de não ser reconhecido pelo algoritmo como um objeto de interesse.

Figura 25 – Exemplo de caracteres com perda de informação



(a) Imagem original



(b) Caracteres segmentados com perda de informação

Durante o desenvolvimento e testes do algoritmo de PDI, nos casos em que o algoritmo não foi capaz de encontrar os objetos de interesse (em 22% das imagens), observou-se um principal motivo: a má escolha do valor de limiar. Quando não é escolhido o melhor valor de limiar podem restar pixels entre caracteres que acabam unindo-os, fazendo com que fossem tratados como um objeto só. Diferentes níveis de iluminação na placa também dificultam a escolha de um limiar global ideal, pois em alguns desses casos caracteres tiveram muita perda de informação a ponto de serem eliminados em um dos três filtros utilizados.

Imagens capturadas com mais de 1,5 metros de distância sofreram ainda mais com estas falhas, obtendo-se resultados não satisfatórios.

4.2 RNAs

Como já foram demonstrados os resultados para a parte de PDI, agora serão demonstrados os resultados apenas para a parte das RNAs.

4.2.1 Comparação entre tipos de redes

Nesta seção serão comparados os diferentes tipos de redes e serão demonstrados como foram escolhidas as configurações e parâmetros para a criação das redes. Os testes foram realizados apenas com a RNA para reconhecer números (RNA 1), pelo fato de possuir a mesma estrutura e tipo de entrada da RNA que reconhece letras (RNA 2), além de ter menor quantidade de caracteres para reconhecimento, facilitando os testes e diminuindo o tempo para execução dos algoritmos e comparações dos resultados. Em todos os testes se buscou escolher a melhor RNA baseada no Energia Média de Erro do último ciclo de treinamento (ε_{med}), que será chamado de Erro Médio Quadrado (ε_{med}), explicado na seção 2.2.5. Um outro parâmetro importante para a escolha de uma RNA é a Acurácia, que é o total de acertos (quando saída real e saída desejada são iguais) dividido pela quantidade de exemplos totais utilizados do conjunto de treinamento, obtida por testes de uma RNA durante o processamento de treinamento. Os valores de erro médio quadrado (ε_{med}) e da acurácia são gerados pelo próprio software de desenvolvimento após o treinamento. Em algumas tabelas, foi realizado o processo de generalização, que são testes (comparação entre saída desejada e saída real) com a rede utilizando amostras da base de dados que não foram usadas durante o treinamento. Na maior parte das tabelas foi exibido o tempo de treinamento, porém ele não é decisivo na escolha da melhor RNA. As linhas de todas as tabelas foram organizadas por ordem crescente do erro médio quadrado (ε_{med}) para facilitar a visualização dos melhores resultados.

4.2.1.1 Comparação entre uma rede de quatro saídas e uma rede com dez saídas

Na fase inicial da projeção das redes, uma das escolhas mais importante foi a de como implementar as saídas. Inicialmente buscou-se criar uma rede com dez saídas, uma para cada caracter numérico. Devido a grande quantidade de saídas e de neurônios, decidiu-se criar uma rede com apenas quatro saídas, o que diminuiria o processamento pela quantidade reduzida de neurônios. A rede funcionaria basicamente como uma lógica binária: quando uma caracter de entrada fosse o número três, por exemplo, a saída deveria ser 1100, equivalente a três em binário, sendo o bit menos significativo a esquerda. Os dois tipos de redes foram criados, quatro saídas e dez saídas, cuja a primeira se demonstrou com menor erro médio quadrado (ε_{med}), tempo de processamento menor e melhor generalização, conforme Tabela 2, sendo o motivo de escolha. Outra característica importante da estrutura da RNA com saída binária seria a maior simplicidade

Tabela 2 – Comparação de qualidade de uma RNA de 4 saídas e uma RNA de 10 saídas

Item	Saídas	Erro Médio Quadrado (ε_{med})	Acurácia (%)	Tempo (mim)	Generalização (%)
1	4	0.00501307	90.31	13.02	50
2	10	0.01060000	95.47	25.79	10

de implementação. Assim a RNA1 foi projetada com 4 saídas e a RNA2 com 5 saídas, valores equivalentes binários para conhecimentos dos números e letras, respectivamente.

4.2.1.2 Comparações entre redes com variações das funções de ativação

Na Tabela 3 podem ser visualizadas as comparações entre funções de ativação diferentes, para realizar a melhor escolha considerando o erro médio quadrado (ε_{med}) e a acurácia para cada RNA. Nesse teste foi usado um tipo de rede com duas camadas, com 10 neurônios na primeira e quatro neurônios na segunda camada, e o restante das configurações foram iguais para todos dos testes. Foram feitas combinações de funções de ativação nas camadas oculta e de saída. A RNA que demonstrou o melhor resultado foi o item 2, que utiliza funções sigmoide na camada oculta e linear na camada de saída, devido a melhor acurácia e segundo menor erro médio quadrado (ε_{med}). O item 1 não foi escolhido, pois apesar do erro médio quadrado (ε_{med}) ser um pouco melhor que do item 2, a acurácia é muito menor, conforme demonstrado na Tabela 3.

Tabela 3 – Comparação da qualidade de uma RNA em função das diferentes funções de ativação

Item	Camada 1	Camada 2	Erro Médio Quadrado (ε_{med})	Acurácia (%)	Tempo (mim)
1	tanh	sigmoide	0.001478	78.44	15.29
2	sigmoide	linear	0.001917	97.19	15.69
3	tanh	linear	0.002119	96.88	13.79
4	sigmoide	sigmoide	0.002561	58.75	16.73
5	limiar	linear	0.002619	91.41	7.59
6	limiar	sigmoide	0.003239	60.94	15.38
7	sigmoide	limiar	0.003373	67.66	15.29
8	sigmoide	tanh	0.003818	57.66	15.49
9	tanh	limiar	0.003967	67.97	15.29
10	linear	sigmoide	0.004459	85.47	15.30
11	tanh	tanh	0.006718	73.44	15.29
12	linear	linear	0.007993	89.84	15.46
13	limiar	limiar	0.011915	66.88	16.89
14	limiar	tanh	0.015400	68.91	16.73
15	linear	limiar	0.015752	77.03	16.30
16	linear	tanh	0.020958	68.13	15.51

4.2.1.3 Comparações entre duas RNAs com diferentes camadas ocultas

Nesta subseção foram feitas comparações entre duas RNA, uma com apenas uma camada oculta e outra com duas camadas ocultas, mantendo o restante de suas configurações idênticas, para validar a comparação. Foram usados 1280 exemplos no conjunto de dados e 100 épocas. Foram avaliados o erro médio quadrado (ε_{med}) e a acurácia, onde os resultados se encontram na Tabela 4. Apesar dos dois resultados serem muito próximos em todos os aspectos, o item 1, com duas camadas ocultas, foi escolhido por ter o erro médio quadrado (ε_{med}) um pouco menor, servindo com critério de desempate nesse caso.

Tabela 4 – Comparação de desempenho em uma RNA com uma e com duas camadas ocultas

Item	Nº de Camadas Ocultas	Erro Médio Quadrado (ε_{med})	Acurácia (%)	Tempo (min)
1	2	0.00329170	96.48	21.61
2	1	0.00333237	97.34	21.04

4.2.1.4 Variação de neurônios

Esta subseção tem o objetivo de demonstrar os resultados de desempenhos da RNA 1 (Tabela 5) e da RNA 2 (Tabela 6), onde em cada tabela os itens 1 foram escolhidos, pelo baixo erro médio quadrado (ε_{med}) e acurácia alta. Para a RNA 1 da Tabela 5, foram testados valores de potência de 2 nas duas camadas ocultas (camada 1 e camada 2), por meio de diferentes combinações, pelo fato da saída ter 4 neurônios, sendo potência de 2. Iniciou-se os testes com 2 neurônios e terminou com 16 neurônios. Para verificar o comportamento da rede quando a camada oculta 2 (adjacente a camada de saída) tivesse um número de neurônios que não fosse múltiplo ou divisor do número de neurônios da camada de saída, usou-se a camada intermediária com 6 neurônios, por ser um valor próximo da mesma quantidade de neurônios da camada de saída, combinado com os outros valores da camada oculta 1. Apesar dos testes empíricos terem sido realizados em ordem crescente do número de neurônios, a Tabela 5 foi organizada por ordem crescente do erro médio quadrado (ε_{med}), para facilitar a identificação do item de melhor resultado. Por ter menor erro médio quadrado (ε_{med}) e a acurácia alta (segunda maior), a estrutura do item 1 da Tabela 5 foi escolhida para a implementação da rede de reconhecimento de números.

Baseando-se na Tabela 5, os dois itens de melhores resultados (item 1 e 2) foram usados para compor os testes para a RNA2 da Tabela 6. Como o melhor resultado da RNA1 foi quando a camada oculta 1 tinha o dobro de neurônios e a camada oculta 2 tinha a mesma quantidade de neurônios da camada de saída, realizou-se mais um teste referente a quantidade de neurônios, porém desta vez para a RNA 2, com essa mesma proporção. Como esperado, a melhor estrutura para a RNA 2 foi o item 1, com melhor erro médio quadrado (ε_{med}) e melhor Acurácia (%), demonstrado na Tabela 6.

Tabela 5 – Desempenho de uma RNA de reconhecimento de números em função dos neurônios nas camadas ocultas

Item	Camada 1	Camada 2	Erro Médio Quadrado (ε_{med})	Acurácia (%)	Tempo (min)
1	8	4	0.00022669	98.28	93.63
2	8	6	0.00085030	96.56	97.61
3	8	16	0.00131664	98.52	101.31
4	16	16	0.00204071	97.50	114.05
5	16	6	0.00228341	97.03	107.91
6	16	8	0.00326614	96.48	107.91
7	4	6	0.00289035	93.91	25.36
8	8	8	0.00533291	96.02	94.62
9	4	16	0.01174950	89.61	26.96
10	2	6	0.01303707	87.11	7.60
11	4	4	0.02834597	93.52	23.63
12	8	2	0.03484084	82.50	93.83
13	4	2	0.03516104	75.23	24.47
14	16	4	0.05127787	71.17	102.78
15	16	2	0.05734660	94.14	106.05
16	2	2	0.08670277	40.47	7.20
17	2	4	0.08875692	67.89	7.67
18	2	8	0.12694901	78.75	8.22
19	2	16	0.12726131	79.22	9.20
20	16	4	0.13043920	59.53	74.67
21	4	8	0.13119829	79.14	24.81

Tabela 6 – Desempenho de uma RNA de reconhecimento de letras em função dos neurônios nas camadas ocultas

Item	Camada 1	Camada 2	Erro Médio Quadrado (ε_{med})	Acurácia (%)	Tempo (min)
1	10	5	0.00373303	92.50	74.33
2	8	6	0.01701124	87.40	71.55
3	8	4	0.02233091	80.10	63.64

4.2.1.5 Comparações entre RNAs com taxas de aprendizado ou passos (η) e constantes de momento (α) diferentes

Para escolher a melhor taxa de aprendizado (η) e momento (α), foram realizados diversos testes com combinações variadas dos dois parâmetros, conforme demonstrados na Tabela 10, que se encontra no apêndice deste documento. Foram usadas 50 épocas e 640 amostras no conjunto de treinamento. Para todos os casos, os parâmetros restantes foram idênticos.

Para realizar a escolha da RNA com base na taxa de aprendizagem (η) e momento (α), foram selecionados seis itens nas Tabelas 10. Os itens 1 e 2 foram escolhidos devido aos menores erro médio quadrado (ε_{med}), os itens 26 e 27 foram selecionados pela melhor acurácia e os itens 5 e 11 por serem intermediários. Um novo teste foi realizado, desta vez apenas com os seis melhores itens mencionados. A Tabela 7 mostra o resultado de desempenho dos seis itens,

utilizando 1280 amostras e 300 épocas.

Tabela 7 – Desempenho de uma RNA em função das variações dos passos e momentos - semi-final

Item	Passo (η)	Momento (α)	Erro Médio Quadrado (ε_{med})	Acurácia (%)
1	0.0500	0.0009	0.00163171	96.09
2	0.5000	0.0010	0.00178287	97.19
3	0.1000	1.0000	0.00280360	98.28
4	0.0100	0.0900	0.00316447	97.50
5	0.0900	1.0000	0.00414045	95.16
6	0.0050	0.0050	0.00518997	96.88

Finalmente foram escolhidos três itens da Tabela 7, um item com menor erro médio quadrado (ε_{med}), um item com maior acurácia e um item intermediário. Mais um teste de comparação foi feito entre esses três itens, utilizando um conjunto de treinamento com 1920 amostras e 600 épocas. O item 1, demonstrado na Tabela 8, foi selecionado como parâmetro para RNA 1 por ter o menor erro médio quadrado (ε_{med}) e maior acurácia. A última coluna mostra o total de acertos na generalização, para trinta testes. Porém, para a RNA 2, apesar de não está exibido em tabela, o item 2 foi o melhor resultado, sendo a escolha para a rede de letras.

Tabela 8 – Desempenho da RNA 1 em função das variações dos passos e momentos - final

Item	Passo (η)	Momento (α)	Erro Médio Quadrado (ε_{med})	Acurácia (%)	Generalização (%)
1	0.5	0.001	0.00239757	97.50	90.00
2	0.05	0.0009	0.00315749	96.61	93.33
3	0.1	1.000	0.00389597	95.73	76.77

4.2.1.6 Comparação entre conjunto de treinamento e épocas

Esta subseção tem o objetivo de comparar entre a escolha entre aumentar o conjunto de dados ou aumentar o ciclo de treinamento de uma RNA. Na Tabela 9 é demonstrada a comparação, para o treinamento de uma mesma RNA, onde o item 1 se tem um conjunto de treinamento menor que do item 2, porém um ciclo de treinamento maior. Observa-se que no item 2, onde houve a decisão de aumentar o conjunto de treinamento e diminuir o ciclo de treinamento, a acurácia foi melhor. Os erros médio quadrado (ε_{med}) dos dois itens foram quase idênticos. Os resultado geral dos dois itens foram muito próximos, porém, como dar prioridade ao conjunto de treinamento significa ensinar a RNA a reconhecer maior variação de exemplos, a prioridade de escolha seria do item 2 para o treino.

Tabela 9 – Ciclo de treinamento x Conjunto de treinamento de uma RNA

Item	Épocas	Exemplos	Erro Médio Quadrado (ε_{med})	Acurácia (%)
1	600	1280	0.00239757	95.73
2	300	2560	0.00240000	97.50

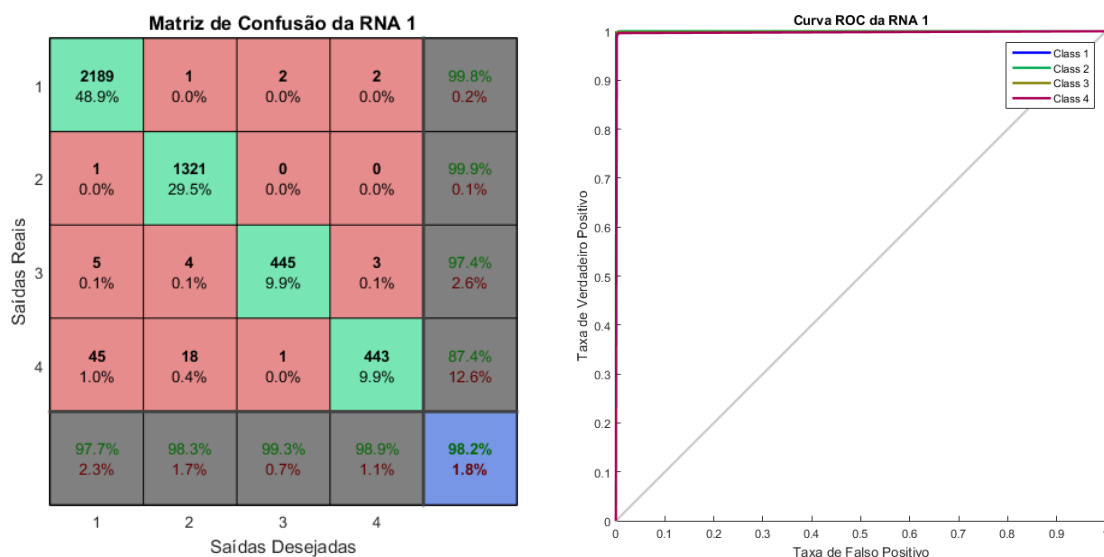
4.3 Resultados das RNAs escolhidas

Nesta seção serão demonstrados os resultados dos treinamentos e das generalizações para a RNA de números (RNA 1) e a RNA de letras (RNA 2), após a definição de seus parâmetros, demonstrado na subseção 4.2.1.

4.3.1 Resultados do treinamento da RNA 1

Nesta subseção estão sendo demonstrados os resultados do treinamento para a RNA 1. O tempo de treinamento foi de 13 horas, utilizando 4480 exemplos do conjunto de dados. O erro médio quadrado (ε_{med}) foi 0.0024. A acurácia, que define a quantidade de acertos utilizando todos os exemplos durante o treinamento da RNA, foi 98.17%, conforme demonstrado na matriz de confusão da Figura 26 (a). O objetivo dessa matriz é demonstrar quantidade de acertos e de erros da rede, comparando as saídas desejadas e saídas reais. Cada linha representa uma saída real e cada coluna uma saída desejada. Na diagonal da matriz se tem os acertos para cada saída e na última célula se tem a acurácia.

Figura 26 – Matriz de Confusão e Curva ROC para a RNA 1



(a) Matriz de Confusão para a RNA de números

(b) Curva ROC para a RNA de números

Fonte: autores, 2017

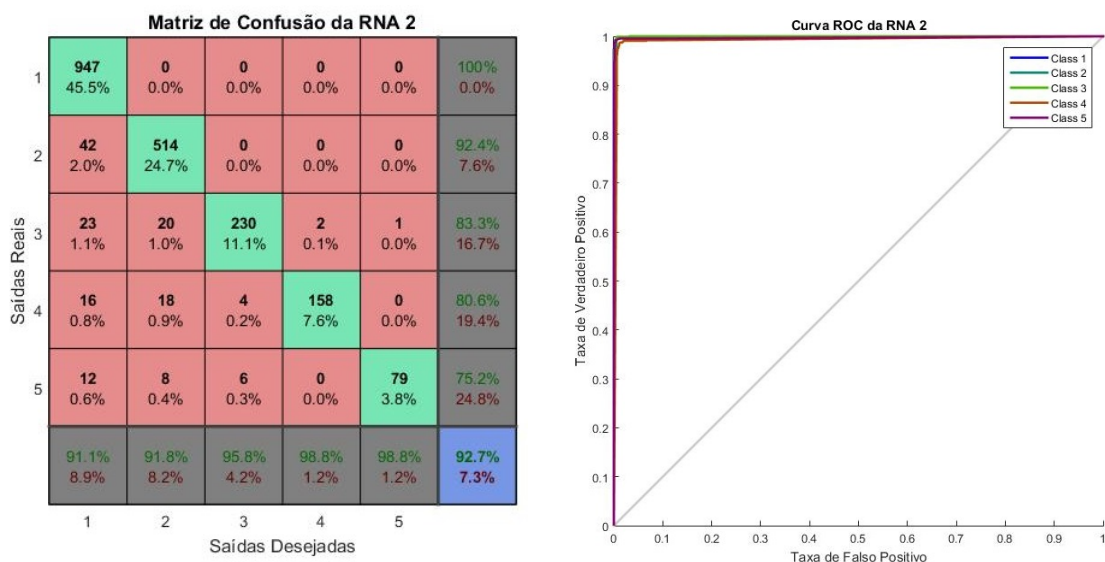
Para verificar a relação entre verdadeiros positivos e falsos positivos, que é quando a saída

real acerta ou erra, respectivamente, durante o treinamento, utiliza-se a curva ROC, demonstrada na Figura 26 (b) para a RNA 1. Cada classe da curva roc representa uma saída. No eixo das ordenadas se tem a taxa de verdadeiros positivos e no eixo das abcissas se tem a taxa de falsos positivos. Para um gráfico desse tipo é desejável que as curvas das classe cresçam apenas em relação ao eixo dos verdadeiros positivos.

4.3.2 Resultados do treinamento da RNA 2

O tempo de treinamento foi de 2.30 horas, utilizando 2400 exemplos do conjunto de dados. O mínimo erro médio quadrado (ε_{med}) foi 0.0066 e acurácia foi 92.70%, conforme demonstrado na matriz de confusão da Figura 27 (a). Na Figura 27 (b) é demonstrado a curva ROC para RNA 2.

Figura 27 – Matriz de Confusão e Curva ROC para a RNA 2



(a) Matriz de Confusão para a RNA de letras

(b) Curva ROC para a RNA de letras

Fonte: autores, 2017

4.3.3 Generalização das RNAs

Durante o processo de generalização, que consistiu em testar as RNAs, após treinadas, utilizando exemplos de entrada que não foram utilizados no conjunto de treinamento, quatro placas foram testadas. Na RNA 1, do total de 16 números segmentados, 13 foram corretamente identificados (cerca de 81,25%). Já para a RNA 2, de 12 letras segmentadas, 7 foram corretamente identificadas (cerca de 58,33%).

4.4 Sistema em conjunto

Após a finalização dos projetos de PDI e RNAs, as duas partes foram testadas em conjunto. Utilizando sete imagens de placas não pertencentes ao conjunto de treinamento, o algoritmo de PDI conseguiu identificar os objetos de interesse em apenas quatro casos (usadas para generalização), das quais a identificação da placa inteira, com 100% de acerto, ocorreu em apenas uma imagem. Na Figura 28 são exibidos os caracteres segmentados para as quatro placas reconhecidas. À esquerda de cada figura é exibido o reconhecimento pelas RNAs e tempo de processamento em cada caso para todo o sistema, em segundos. O tempo médio foi 1,15 segundo. Assim, o sistema em conjunto, durante o processo de generalização, reconheceu apenas 1 de 7 placas (14,29% de acertos).

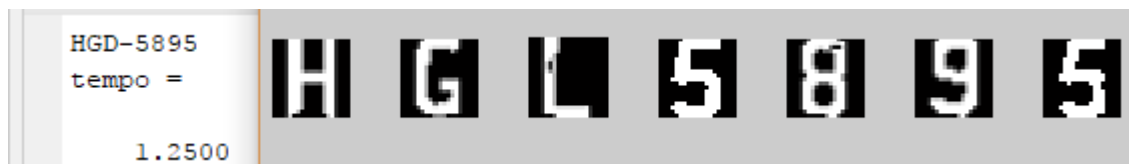
Figura 28 – Testes feitos durante a generalização



(a) Teste com a placa HCQ-1991 - 28.57% de acerto



(b) Teste com a placa PVA-5593 - 71.43% de acerto



(c) Teste com a placa HGL-5895 - 85.71% de acerto



(d) Teste com a placa HGL-5684 - 100% de acerto

Fonte: autores, 2017

5 CONSIDERAÇÕES FINAIS

O sistema de reconhecimento automático de placas desenvolvido pelos autores se mostrou rápido para a maior parte dos testes, executando a identificação em pouco mais de 1 segundo na maioria dos casos. Os algoritmos utilizados possuem falhas que poderiam ser melhorados em projetos futuros. Durante a realização do PDI, por exemplo, poderiam ser estudadas técnicas mais complexas de limiarização em que se considera os diferentes níveis de iluminação em cada parte da imagem. Dessa maneira, reduziria-se as falhas causadas pela escolha de um valor de limiar não ideal ou valores globais que não funcionam para toda a região da placa.

Durante o processo de treinamento, baseado no erro médio quadrado (ε_{med}), acurácia e curva roc, as RNAs demonstrarão bom desempenho. A RNA 1 teve um erro médio quadrado (ε_{med}) de apenas 0.0024 e acurácia de mais de 98% e a RN2 teve um erro médio quadrado (ε_{med}) de 0.0066 e acurácia de mais de 92%. Na generalização, as RNAs se demonstraram com bom desempenho, reconhecendo maior parte dos caracteres. Apesar de serem geradas novas imagens por meio de filtros e ruídos do Matlab, a dificuldade de aquisição do conjunto de treinamento e, conseqüentemente, o número reduzido de exemplos captados pela câmera, influenciou expressivamente em falhas de reconhecimentos de alguns caracteres na parte de generalização. A RNA 2 se demonstrou com menor qualidade em relação a RNA1 devido ao menor conjunto de treinamento e a maior quantidade de caracteres diferentes necessários para reconhecimento.

As maiores dificuldades no desenvolvimento do projeto foram a aquisição das imagens padronizadas pelos autores, a descoberta dos melhores parâmetros das RNAs por meio da teoria e testes empíricos e a segmentação dos caracteres quando as imagens possuíam diferentes níveis de luz na região da placa.

REFERÊNCIAS

- ANAGNOSTOPOULOS, C. N. E. et al. **A License Plate-Recognition Algorithm for Intelligent Transportation System Applications**. *IEEE Transactions on intelligent transportation systems*, 2006. Citado 2 vezes nas páginas 10 e 29.
- BABU, M. H. et al. **Design of Portable Security System Using Face Recognition with Back-Propagation Algorithm and MATLAB**. In: Suresh, LP and Dash, SS and Panigrahi, BK (Ed.). **Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, Vol 1**. HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY: SPRINGER-VERLAG BERLIN, 2015. (Advances in Intelligent Systems and Computing, 324), p. 247–255. ISBN 978-81-322-2126-5; 978-81-322-2125-8. ISSN 2194-5357. Citado na página 28.
- BRITO, E. A. de. **Biologia**. [S.l.]: Poliedro, 2015. Citado na página 16.
- CIRESAN, D. et al. **Multi-Column Deep Neural Network for Traffic Sign Classification**. 2012. Citado na página 29.
- D'ANGELO, H. **Cérebro humano tem 16 tipos diferentes de neurônios, revela estudo**. 2016. <<http://super.abril.com.br/saude/cerebro-humano-tem-16-tipos-diferentes-de-neuronios-revela-estudo/>>. Acesso em 1 de agosto de 2016. Citado na página 15.
- DESCONHECIDO, A. **Carros na Web**. 2017. <<http://www.carrosnaweb.com.br/dicasplacas.asp>>. Acesso em 12 de junho de 2017. Citado na página 32.
- GONZALES, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. 1. ed. São Paulo: Edgard Blucher, 2012. Citado 2 vezes nas páginas 13 e 14.
- GUINGO, B. C.; THOMÉ, A. C. G.; RODRIGUES, R. J. **Reconhecimento Automático de Placas de Veículos Automotores Através de Redes Neurais Artificiais**. 2002. Citado na página 28.
- HAYKIN, S. **Redes Neurais: princípios e prática**. 2. ed. [S.l.]: Bookman, 2001. Citado 7 vezes nas páginas 16, 17, 19, 21, 23, 27 e 28.
- JR., O. L.; MONTGOMERY, E. **Redes Neurais: Fundamentos e aplicações com programas em c**. 1. ed. Rio de Janeiro: Ciência Moderna, 2007. Citado 6 vezes nas páginas 18, 20, 21, 22, 23 e 25.
- MEDEIROS, L. F. de. **Redes Neurais em Delphi**. 1. ed. Florianópolis: Visual Books, 2003. Citado 3 vezes nas páginas 23, 25 e 27.
- NEUBAUER, C. **Evaluation of convolutional neural networks for visual recognition**. *IEEE Transactions on Neural Networks, IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC*, 345 E 47TH ST, NEW YORK, NY 10017-2394 USA, 9, n. 4, p. 685–696, JUL 1998. ISSN 1045-9227. Citado na página 29.
- PERALES, T. R. **Utilização de Redes Neurais Artificiais no Diagnóstico de Cardiopatias**. Dissertação (Mestrado) — Departamento de Engenharia Elétrica da Universidade Estadual de Londrina, 2011. Citado 3 vezes nas páginas 20, 21 e 24.

- RODRIGUES, F. A. **Localização e Reconhecimento de Placas de Sinalização Utilizando um Mecanismo de Atenção Visual e Redes Neurais Artificiais**. Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2002. Citado na página 29.
- RUSSELL, P. N. S. **Inteligência Artificial**. 2. ed. [S.l.]: Eletrônica, 2004. Citado na página 17.
- TABRIZI, S. S.; CAVUS, N. **A hybrid KNN-SVM model for Iranian license plate recognition**. 2016. Citado na página 29.
- YANG, W. et al. **DropSample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition**. *Pattern Recognition*, ELSEVIER SCI LTD, THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD OX5 1GB, OXON, ENGLAND, 58, p. 190–203, OCT 2016. ISSN 0031-3203. Citado na página 28.

APÊNDICE A – TABELA DE COMPARAÇÃO DE PARÂMETROS

Tabela 10 – Variações dos passos e momentos

Item	Passo (η)	Momento (α)	Erro Médio Quadrado(ε_{med})	Acurácia (%)
1	0.1000	1.0000	0.00129259	94.06
2	0.0050	0.0050	0.00135208	96.88
3	0.1000	0.9000	0.00138074	95.16
4	0.0010	0.0900	0.00140512	94.69
5	0.0500	0.0009	0.00151648	97.50
6	0.0009	0.0005	0.00163534	94.84
7	0.0900	0.0090	0.00169413	95.31
8	0.9000	0.0010	0.00171226	93.44
9	0.0900	0.0500	0.00175079	95.78
10	0.5000	0.0001	0.00177223	95.78
11	0.0100	0.0900	0.00180518	97.50
12	1.0000	0.0100	0.00196320	94.53
13	0.0500	0.5000	0.00196488	94.53
14	0.9000	0.0001	0.00197709	96.88
15	0.9000	0.0100	0.00200559	90.47
16	0.0100	0.0090	0.00206232	95.78
17	0.0009	0.9000	0.00208317	95.00
18	1.0000	0.0005	0.00211598	97.03
19	0.0005	0.0090	0.00223479	94.38
20	0.0001	0.1000	0.00223770	96.25
21	0.0009	1.0000	0.00225288	97.03
22	0.0010	0.1000	0.00227597	89.53
23	0.0005	0.0005	0.00228203	95.47
24	0.5000	0.0100	0.00233772	96.56
25	0.0001	0.5000	0.00245901	96.09
26	0.0900	1.0000	0.00246438	98.13
27	0.5000	0.0010	0.00247302	98.13
28	0.1000	0.0001	0.00247631	94.69
29	0.0090	0.0009	0.00249352	86.25
30	0.1000	0.0500	0.00256237	93.91
31	0.0009	0.0090	0.00265745	94.38
32	0.0100	0.0001	0.00272582	97.19
33	0.1000	0.0009	0.00278182	96.72

34	0.0100	0.0050	0.00278458	97.34
35	0.0050	1.0000	0.00283491	96.09
36	0.0010	0.0010	0.00286667	93.28
37	0.5000	0.1000	0.00288071	96.09
38	0.9000	1.0000	0.00293460	89.53
39	0.0005	0.0009	0.00294068	93.44
40	0.0050	0.0001	0.00295172	89.53
41	0.0900	0.0900	0.00295657	91.09
42	1.0000	0.0900	0.00300721	92.97
43	0.1000	0.0050	0.00301314	92.50
44	0.0100	0.0009	0.00309305	92.66
45	0.0009	0.0001	0.00310806	96.72
46	0.1000	0.1000	0.00311633	93.28
47	0.0090	0.0900	0.00312886	89.84
48	0.0900	0.0005	0.00315763	96.09
49	1.0000	1.0000	0.00332214	95.47
50	0.1000	0.0900	0.00333054	94.69
51	0.0500	0.0010	0.00334301	95.47
52	0.0100	0.0010	0.00336488	94.06
53	0.0500	0.0090	0.00353950	96.41
54	0.0005	0.0900	0.00358666	90.63
55	0.1000	0.5000	0.00358965	92.50
56	0.5000	1.0000	0.00364136	91.56
57	0.0010	1.0000	0.00396313	96.41
58	0.0001	0.0900	0.00403448	95.78
59	0.0900	0.1000	0.00409597	87.97
60	0.0005	0.5000	0.00409662	94.53
61	0.0010	0.0009	0.00409704	95.31
62	0.0900	0.0009	0.00415674	96.25
63	0.0009	0.0010	0.00418098	95.78
64	0.0050	0.0009	0.00419381	84.53
65	0.0100	0.0005	0.00434216	93.59
66	0.0010	0.0005	0.00444169	96.09
67	0.0005	0.0500	0.00444914	95.00
68	1.0000	0.0090	0.00445103	94.84
69	0.0005	1.0000	0.00448095	95.31
70	0.0090	0.0090	0.00451990	93.44
71	0.0001	0.0090	0.00456511	82.81

72	0.1000	0.0090	0.00456711	93.75
73	0.5000	0.0500	0.00457869	82.66
74	0.0500	0.0500	0.00465933	93.28
75	0.5000	0.0005	0.00467092	92.81
76	0.0009	0.5000	0.00480308	89.69
77	0.0050	0.0500	0.00488718	93.28
78	0.0090	0.9000	0.00496378	90.78
79	0.5000	0.0900	0.00522380	84.38
80	0.1000	0.0100	0.00537519	95.00
81	0.0050	0.5000	0.00544874	95.00
82	0.0009	0.0009	0.00545076	75.47
83	0.0090	1.0000	0.00546542	88.91
84	0.0500	0.0001	0.00560820	96.56
85	0.5000	0.0090	0.00565304	89.69
86	1.0000	0.0001	0.00569338	96.09
87	0.0100	0.0100	0.00585849	89.69
88	0.0050	0.0010	0.00597260	88.13
89	0.0500	0.0900	0.00603402	94.53
90	0.5000	0.9000	0.00608869	92.03
91	0.0090	0.0500	0.00616299	94.69
92	0.0005	0.0010	0.00620529	93.13
93	0.0050	0.0900	0.00634362	81.88
94	0.0500	0.9000	0.00680257	93.28
95	0.0001	0.0100	0.00696752	85.47
96	1.0000	0.0010	0.00753889	86.25
97	0.0090	0.0100	0.00762544	94.22
98	1.0000	0.0500	0.00812076	92.19
99	1.0000	0.0009	0.00865687	89.38
100	0.5000	0.0050	0.00971510	86.72
101	0.0090	0.0010	0.01069006	89.53
102	0.0010	0.0500	0.01170302	88.44
103	0.0100	1.0000	0.01232956	78.13
104	0.9000	0.5000	0.01252785	79.84
105	0.9000	0.0500	0.01515237	92.34
106	0.0010	0.0090	0.01841681	87.97
107	0.0005	0.1000	0.01852545	92.81
108	0.5000	0.5000	0.02014053	75.94
109	0.9000	0.9000	0.02071802	91.25

110	0.0900	0.5000	0.02082563	89.53
111	1.0000	0.9000	0.02113854	87.50
112	0.0009	0.1000	0.02184987	87.50
113	0.9000	0.0005	0.02235939	58.44
114	0.9000	0.0050	0.02248672	90.63
115	0.0900	0.0010	0.02379455	97.03
116	1.0000	0.1000	0.02396033	80.31
117	0.0009	0.0100	0.02648804	93.75
118	0.0050	0.1000	0.02658803	77.81
119	0.0001	0.0005	0.02724104	89.53
120	0.0010	0.0050	0.02727618	86.09
122	0.9000	0.1000	0.02842324	92.97
123	0.0005	0.0001	0.02944219	79.06
124	0.0001	1.0000	0.02972811	86.56
125	0.0050	0.0005	0.02993337	80.16
126	0.0500	1.0000	0.03016337	78.59
127	1.0000	0.5000	0.03055332	85.63
128	0.0010	0.9000	0.03068525	92.81
129	0.0005	0.0050	0.03161657	84.22
130	0.0050	0.0100	0.03164370	85.31
131	0.0100	0.9000	0.03498836	91.41
132	0.0090	0.0050	0.03870576	75.31
133	0.0001	0.9000	0.03897460	88.75
134	0.0001	0.0050	0.04820197	79.06
135	0.0001	0.0001	0.04883381	86.25
136	0.0050	0.9000	0.04993814	92.03
137	0.0100	0.0500	0.06050730	83.44
138	0.0005	0.9000	0.06091843	92.81
139	0.9000	0.0900	0.06162624	79.06
140	0.0001	0.0500	0.06569042	66.25
141	0.0500	0.1000	0.06860482	76.41
142	0.0900	0.9000	0.07058864	78.91
143	0.0001	0.0010	0.07441465	80.00
144	0.0010	0.0001	0.08241559	78.28
145	0.0100	0.1000	0.08296223	90.63
146	0.0090	0.0001	0.08522064	88.13
147	0.0100	0.5000	0.08732132	86.56
148	0.0500	0.0005	0.08823229	89.53

149	0.1000	0.0005	0.10363760	81.25
150	1.0000	0.0050	0.12866895	72.97
151	0.0900	0.0100	0.13121695	79.53
152	0.0009	0.0500	0.13157671	74.38
153	0.9000	0.0090	0.13195656	79.06
154	0.9000	0.0009	0.13235123	79.69
155	0.1000	0.0010	0.13350984	73.59
156	0.0005	0.0100	0.13507962	79.38
157	0.0010	0.0100	0.13561937	77.50
158	0.0050	0.0090	0.13755300	78.28
159	0.0500	0.0100	0.13801919	78.13
160	0.0900	0.0001	0.14337126	72.50
161	0.0001	0.0009	0.14534788	77.97
162	0.0900	0.0050	0.14725509	59.84
163	0.0090	0.5000	0.15034039	77.66
164	0.5000	0.0009	0.15224443	80.00
165	0.0500	0.0050	0.15230223	57.81
166	0.0090	0.0005	0.15401504	86.41
167	0.0090	0.1000	0.15629084	88.75
168	0.0009	0.0900	0.16108481	53.13
169	0.0010	0.5000	0.16257441	59.69
