

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Engenharia da Computação

Moisés Oliveira Silva

**USO DE TÉCNICAS COMPUTACIONAIS PARA O RECONHECIMENTO DE
PLACAS AUTOMOTIVAS PARA O AUXÍLIO NA FISCALIZAÇÃO DE
VELOCIDADE**

Timóteo

2014

Moisés Oliveira Silva

**USO DE TÉCNICAS COMPUTACIONAIS PARA O RECONHECIMENTO DE
PLACAS AUTOMOTIVAS PARA O AUXÍLIO NA FISCALIZAÇÃO DE
VELOCIDADE**

Monografia apresentada ao curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais Campus Timóteo, como requisito parcial de obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Douglas Nunes de Oliveira

Timóteo

2014

Moisés Oliveira Silva

**USO DE TÉCNICAS COMPUTACIONAIS PARA O RECONHECIMENTO DE
PLACAS AUTOMOTIVAS E AUXILIO NA FISCALIZAÇÃO DE VELOCIDADE**

Monografia apresentada ao curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais Campus Timóteo, como requisito parcial de obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Douglas Nunes de Oliveira

Msc. Deisymar Botega tavares

Professor Orientador

Msc. Douglas Nunes de Oliveira

Professor Avaliador 1

Msc. Aléssio Miranda Junior

Professor Avaliador 2

Timóteo, 14 de março de 2014.

A Deus, minha família e amigos,
pelo incentivo e apoio nessa etapa da minha vida.

AGRADECIMENTOS

Agradeço a Deus por tudo que ele já fez por mim, pois eu sei também que continuará a fazer muitas outras grandes coisas em minha vida.

Sou grato também a toda a minha família, minha mãe e meu pai por sempre terem me apoiado e estarem todo esse tempo ao meu lado. Aos meus irmãos Jefferson e Hudson, que sempre foram meus melhores companheiros e sempre estiveram comigo nos meus momentos mais difíceis. Aos meus amigos ou praticamente irmãos André Luccas, Júlio César e Pedro Evandro que me ajudaram durante todo meu ensino fundamental, ensino médio e também no cursinho pré-vestibular, pois foram uns dos que me deram mais forças para que chegasse até aqui.

Agradeço também a minha namorada, por ser minha companheira do dia-a-dia, minha melhor amiga e meu grande amor. Pois sem ela seria muito mais difícil e mais chato cursar Engenharia.

Por último agradeço aos meus professores Marcelo Balbino, Deisymar Botega e Douglas Nunes. Marcelo por me ajudar a formar com horários das disciplinas mais flexíveis e também por sempre ser prestativo e atencioso aos problemas do curso. A Deisymar por colaborar muito com minha formação acadêmica, e me fazer a gostar da área de engenharia de software devido ótimas aulas. Finalmente ao professor Douglas, por me orientar neste trabalho de conclusão de curso com grande dedicação e atenção e também pela melhora na minha formação acadêmica.

“Tudo posso naquele que me fortalece.” Filipenses 4:13.

RESUMO

Este trabalho trata-se em desenvolver Redes Neurais Artificiais para reconhecimento de padrões em placas veiculares, esse desenvolvimento poderá auxiliar a fiscalização eletrônica de velocidade. Para a implementação dessas redes foi necessário desenvolver um software que pudesse realizar o cadastro e a rotulação dos dados, ou seja, um software capaz de definir e armazenar a localização da placa, de identificar e rotular cada caracteres, de identificar e rotular a cor da placa e também definir a localização dos caracteres da placa. A criação de uma massa de dados fez-se necessária para tornar possível os treinamentos das redes neurais. Neste trabalho foi necessário utilizar conceitos de Processamento de Imagens juntamente com os de Redes Neurais Artificiais para alcançar os objetivos. A maioria dos objetivos propostos neste trabalho foram alcançados com êxito, sendo possível identificar um caractere, diferenciar o cor da placa e também saber localizar a placa confirmando assim a obtenção dos resultados esperados.

Palavras-chave: Reconhecimento de Placas, Processamento de imagens, Redes Neurais Artificiais, Reconhecimento de Padrões.

ABSTRACT

This work is in developing neural networks for pattern recognition in vehicular plates , this development may assist the electronic speed trap . For the implementation of these networks was necessary to develop a software that could perform the registration and labeling of data , ie a software able to set and store the location of the plate , to identify and label each character , to identify and label the color plate and also define the location of the characters of the plate. The creation of a mass of data was needed to make the training of neural networks possible. This work was necessary to use concepts of image processing together with Artificial Neural Networks to achieve the goals . Most proposed objectives of this study were successfully achieved , and can identify a character to differentiate the color of the plate and also learn to locate the plate thus confirming the achievement of expected results .

Keywords: Plate Recognition, Image Processing, Neural Networks, Pattern Recognition.

.

LISTA DE FIGURAS

Figura 1- Redes Neurais Artificiais.....	19
Figura 2- Funções de Ativação	20
Figura 3 - Perceptron Camada Única.....	22
Figura 4 - Multicamada.....	22
Figura 5 - Rede de Hopfield	23
Figura 6 - Rede de Kohonen.....	24
Figura 7 - Pixel RGB	26
Figura 8 - Esquema SVA.....	27
Figura 9 - Imagem Monocromática no eixo X,Y	30
Figura - 10 Matriz dos pixels	31
Figura 11 - Vizinhos de um pixel.....	32
Figura 12 - Exemplos de Histogramas.....	34
Figura - 13 Placa Cinza Nova	35
Figura 14 - Placa Cinza Desgastada.....	35
Figura 15 - Placa Com Vários Caracteres	36
Figura - 16 Placa Vermelha.....	36
Figura 17 - Software de Cadastro de Placas.....	37
Figura 18 - Histograma de uma placa	40
Figura 19 - Gráfico da RNA de Reconhecimento de Caractere.....	43

LISTA DE ABREVIATURAS

RNA - Rede Neural Artificial

AG - Algoritmos Genéticos

V_m - Velocidade média

Δt - Variação do tempo

Δs - Variação do espaço

SVA - Sistema de Visão Artificial

MLP - Perceptron de Múltiplas Camadas

SOM - Self-organized map

SUMÁRIO

1. INTRODUÇÃO	12
1.1. Motivação	12
1.2. Objetivo	13
2. ESTADO DA ARTE	14
2.1. Histórico	14
2.2. Redes Neurais nos dias atuais	15
2.3. Uso híbrido de redes neurais com outras tecnologias	16
2.3.1. Redes neurais com Algoritmos Genéticos	16
2.3.2. Redes neurais com Lógica Fuzzy.....	17
3. FUNDAMENTOS TEÓRICOS.....	17
3.1. Conceitos de Inteligência Artificial.....	18
3.1.1. Redes Neurais Artificiais.....	18
3.1.2. Tipos de aprendizagem.....	20
3.1.3. Rotulação de dados.....	21
3.1.4. Topologia e Modelos de redes neurais	21
3.1.5. Algoritmo de Treinamento Perceptron.....	24
3.2. Processamento digital de Imagens	25
3.2.1. Estrutura de um sistema de visão Artificial.....	26
3.2.1.1. Aquisição de imagem e Pré-Processamento	27
3.2.1.2. Segmentação e Extração de Caracteres	28
3.2.1.3. Reconhecimento, Interpretação e Base de Conhecimento	28
3.2.2. Propriedade de uma Imagem digital.....	29
3.2.3. Digitalização.....	30
3.3. Propriedades de uma imagem digital	31
3.3.1. Vizinhança e Conectividade.....	31
3.3.2. Adjacência e Distância.....	32
3.4. Histograma.....	33
4. RESULTADOS	34
4.1. Coleta de Imagens.....	34
4.2. Obtenção massa de dados	37
4.3. Extração de Características da imagem.....	38
4.4. Treinamento das RNA's	40
4.4.1. RNA de reconhecimento da localização da placa.	41

4.4.2.	<i>RNA de reconhecimento da cor ou tipo da placa.....</i>	42
4.4.3.	<i>RNA para a identificação caracteres.....</i>	43
5.	CONCLUSÃO E TRABALHOS FUTUROS.....	44
	REFERÊNCIAS	45

1. INTRODUÇÃO

O não cumprimento das leis de trânsito pelos condutores no Brasil é uma atitude frequente nos dias atuais, o que gera uma preocupação nos órgãos de trânsito brasileiros, que sempre buscam formas para que os condutores respeitem e cumpram as Leis do Código de Trânsito.

Para isso a fiscalização sempre é necessária para auxiliar o cumprimento das leis de trânsito e também contribui com a redução de acidentes no trânsito, principalmente em trechos perigosos, como a BR-381, ou em regiões escolares.

Art. 6º São objetivos básicos do Sistema Nacional de Trânsito:

I - estabelecer diretrizes da Política Nacional de Trânsito, com vistas à **segurança**, à fluidez, ao conforto, à defesa ambiental e à educação para o trânsito, e **fiscalizar** seu cumprimento. (CÓDIGO DE TRÂNSITO BRASILEIRO, 1997, art. 6).

Essa redução pode ser alcançada através de ações corretoras como penalidades e medidas administrativas, dentre as penalidades a mais comum é a multa.

A aplicação de multas contribui com a reeducação dos condutores, pois ao receber uma punição pelo não cumprimento de uma lei, de certa forma fará com que o condutor pense e reflita no motivo pelo qual este foi punido.

Este trabalho tem como finalidade desenvolver Redes Neurais Artificiais para o reconhecimento de placas veiculares. Essas redes podem ser utilizadas para auxiliar na fiscalização e controle da velocidade de autos nas vias brasileiras sendo elas rodovias, ruas e avenidas.

1.1. Motivação

A motivação principal para realização desse projeto foi a implementação, o estudo e também o desenvolvimento de redes neurais artificiais para o reconhecimento de padrões. O motivo pelo qual optou-se pela construção dessas redes foi devido a não disponibilidade dos códigos fontes dos programas já existentes (códigos proprietários), o baixo número de trabalhos científicos focados

nessa área e também pelo desconhecimento de projetos que utilizem os mesmos métodos e técnicas a serem implementadas neste trabalho, ou seja, uso de redes neurais artificiais para o reconhecimento de padrões. A equipe envolvida neste trabalho também busca verificar se os métodos e técnicas adotados serão capazes de cumprir com o objetivo principal, e caso não cumpra qual seria o motivo e que será necessário para alcançá-lo.

O desenvolvimento dessas redes também servirá para que em um futuro seja possível realizar um comparativo entre as técnicas existentes no meio científico, no mercado ou em outras técnicas desenvolvidas pela própria equipe do projeto, para que assim seja possível verificar o método mais eficiente para cada situação.

1.2. Objetivo

O objetivo principal desse projeto é o reconhecimento de placas de automóveis em imagens obtidas por radares. Nesse projeto optou-se pelo desenvolvimentos completo de novas estruturas de reconhecimento de placas, pois a maioria dos softwares existentes nessa área são proprietários e também não é possível saber os recursos e técnicas utilizadas para obter o resultado desejado. Para alcançar essa meta dividiu-se o objetivo principal em vários outros subobjetivos. Tais como:

- Geração da massa dados. A massa dados é uma coleção de imagens a ser utilizada no treinamento das Redes Neurais Artificiais (RNA's). Para a criação dessa base é necessário realizar alguns trabalhos de campo com o intuito de capturar imagens de veículos. A etapa de extração de característica também utiliza essa base.
- Extração de características de cada imagem. As imagens coletadas nas fase de geração de massa de dados são utilizadas no software de cadastro de características, software esse responsável por extrair informações chaves da imagem e armazená-las no banco. Essa informações serão utilizadas nos treinamento das RNA's.
- Treinamento e teste de uma RNA para localizar a posição das placas em uma imagem. Esta subfase é responsável pelo treinamento,

através de uma rede neural de múltiplas camadas, para localizar a placa na imagem.

- Treinamento de uma RNA para localizar a posição dos caracteres em uma placa. Essa etapa é similar a anterior com um diferença apenas, ao invés de treinar a localização da placa essa fase treina a localização dos caracteres contidos na placa.
- Treinamento de uma RNA para identificar se uma placa é vermelha ou não. Esse treinamento é responsável por verificar o tipo da placa da imagem, se é uma placa de cor cinza ou de cor vermelha.
- Treinamento de uma RNA para identificação dos caracteres. Neste caso o intuito do treinamento é aprender a identificar e também reconhecer um caractere.

2. ESTADO DA ARTE

Para um maior esclarecimento sobre o assunto será feito uma pequena introdução histórica sobre o assunto. Com isso será possível, de um certo modo, ter um embasamento teórico atualizado e coerente.

2.1. Histórico

Segundo Russell e Norvig (2004), em 1943 foi desenvolvido o primeiro modelo artificial de um neurônio, com suas capacidades computacionais. Em homenagem aos seus criadores Warren Mcculloch, um psiquiatra e neuroanatomista, e Walter Pitts, um matemático, o modelo foi batizado com o nome de MCP

Donald Hebb apresentou em 1949 o primeiro trabalho sobre aprendizagem de redes biológicas e artificiais. Neste trabalho ele conseguiu demonstrar que a plasticidade da aprendizagem de um rede neural é obtida através dos pesos de entrada dos nodos biológicos, baseado no reforço das ligações sinápticas entre os nodos excitados.

No ano de 1958 Frank Rosenbalt demonstrou, com o seu novo modelo, que se fossem acrescentadas de sinapses as RNAs com nodos MPC poderiam ser treinadas para classificar certos tipos de padrões. Com isso ele buscava que as RNAs fossem capazes de fazer descobertas interessantes e também pertinentes ao assunto proposto. O Perceptron, nome que foi dado ao modelo de Rosenbalt, é um classificador binário que mapeia sua entrada x para um valor de saída $f(x)$ através de uma matriz. Este valor da saída é um número binário simples.

$$f(x) = \begin{cases} 1 & \text{se } w \cdot i + b > 0 \\ 0 & \text{se } w \cdot i + b \leq 0 \end{cases}$$

A variável w representa um vetor de peso, a operação $w \cdot i$ é um produto escalar responsável pela obtenção das soma dos pesos. Já b é um constante independente de qualquer valor da entrada.

Em 1986 Rumelhart e McClelland criaram o algoritmo Back-Propagation para o treinamento de um MLP (Perceptron Múltiplas Camadas). Paralelamente a estes trabalhos, Hopfield (1982) descreve uma rede auto associativa e de aprendizado não supervisionado, esta rede foi batizada com seu próprio nome.

2.2. Redes Neurais nos dias atuais

Segundo Iris (2003), com o passar do tempo as Rede Neurais Artificiais (RNA) começaram a ser empregadas em diversas áreas, isso é desde áreas médicas, mecânicas, metalúrgicas, processamento de imagens e outras.

Um dos projetos mais sucedidos e também um dos mais importantes e influentes na atualidade é o projeto de Ronald A. Fisher, que desenvolveu uma base de dados chamada Iris Dat Set. Neste projeto, ele quantificou a variação morfológica das flores Iris utilizando como base três espécies desta.

O conjunto de dados é constituído por 50 amostras de cada uma das três espécies de Iris proposta (Iris Setosa, Iris Virginica e Iris Versicolor). Quatro características foram medidas a partir de cada uma das amostras, o comprimento e a largura das sépalas e pétalas, isso em centímetros. A partir dessas medidas e com base na combinação destas quatro características, Fisher desenvolveu um modelo discriminante linear para distinguir as espécies uma das outras.

Com base no modelo discriminante linear de Fisher, este conjunto de dados tornou-se um caso de teste típico para muitas técnicas de classificação em algoritmo de aprendizagem, um exemplo é máquinas de vetores de suporte.

Existem vários outros projetos na área utilizando os conceitos de redes neurais, tais como:

- Fertility (Médica)
- WineQuality (Vinho Tinto, Química)
- PimaIndians Diabetes (Médica)
- CardioTocoGraphy (Médica)
- AcuteInflammations (Médica)
- BreastCancer (Médica)
- PokerHand (Jogos)
- Glass Identification (Ciência criminal)

2.3. Uso híbrido de redes neurais com outras tecnologias

Uma tendência de acordo com Braga (2007) que alguns anos vem sendo seguida é o uso híbrido da RNA com outras tecnologias da Inteligência artificial. As mais utilizadas são as combinações entre as RNAs com Algoritmos Genéticos (AGs) e RNAs com Lógica Fuzzy.

2.3.1. Redes neurais com Algoritmos Genéticos

Algoritmos de otimização e busca segundo Braga (2007) são baseados no mecanismo de seleção natural e genética e também na Teoria da Evolução de Chales Darwin, são denominados Algoritmos Genéticos.

Esses algoritmos utilizam conceitos como os de seleção natural, para uma tentar garantir uma melhora na população (conjunto de possíveis soluções) e consequentemente o mantimento dos indivíduos (uma possível solução) mais aptos. Alguns métodos são utilizados para que se obtenha uma população mais apta, dentre elas as mais comuns são o Recombinação, Crossover e a Mutação. Isso faz as gerações melhorar suas características mais importantes a cada geração.

Com a utilização de Algoritmos Genéticos associados a Redes Neurais, o conjunto de entrada das RNAs já será uma população melhorada pelo AG, pressupondo que este aplicará várias técnicas com o intuito de melhorar sua população (conjunto de entradas) para que a cada geração se obtenha os melhores indivíduos, sendo neste caso as melhores entradas.

2.3.2. Redes neurais com Lógica Fuzzy

Implementações da Lógica Fuzzy possibilita a obtenção de estados indeterminados que podem ser tratados por diversos dispositivos de controle. Com isso, é possível avaliar e chegar a conceitos não quantificáveis. É como avaliar grandezas que não possui apenas dois estados definido, a temperatura representa um bom exemplo para essa situação, pois ela pode ser quente, morna, médio, fria e etc. O sentimento das pessoas seria um outro bom exemplo de grandezas não quantificáveis, ela pode ser *feliz*, *apático*, *triste* e etc.

A Lógica Fuzzy é uma área de pesquisa sobre tratamento da incerteza, ou de modelos matemáticos que são dedicados ao tratamento da incerteza, para que se obtenha os seus devidos estados não quantificáveis.

Semelhante a relação que um Algoritmo Genético tem com uma rede neural, a Logica Fuzzy também possui uma cooperação parecida com o melhoramento das entradas das funções ativadoras, pois aumenta o conjunto de possíveis entradas. Isso faz com que se possa trabalhar com conceitos ou assuntos não quantificáveis de uma forma computacional.

3. FUNDAMENTOS TEÓRICOS

Neste tópico serão tratados todos os conceitos relacionados ao trabalho proposto, tanto os conceitos da área de informática como também das outras áreas que tem influência direta neste trabalho de conclusão de curso.

3.1. Conceitos de Inteligência Artificial

A partir desse tópico serão tratados os conceitos mais específicos de inteligência artificial e suas aplicações no projeto.

3.1.1. Redes Neurais Artificiais

As redes neurais artificiais consistem em um método de solucionar problemas de inteligência artificial, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São mais que isso, são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, enquanto que o cérebro de um mamífero pode ter muitos bilhões de neurônios. (TATIBANA; KAETSU. [200-?])

Uma Rede Neural Artificial apresenta dois segmentos principais: a arquitetura e o algoritmo de aprendizagem. Diferentemente da arquitetura de Von Neumann que executa programas, a rede treina em exemplos. A partir desses treinamentos ela obtém um certo conhecimento do problema e os armazena, para garantir esse conhecimento da situação problema é necessário que sempre esteja disponível os exemplos de treinamento, pois são neles que as informações foram guardadas. Os pesos são onde se memoriza a generalização dos dados feita pelo algoritmo de aprendizagem, pois eles são os parâmetros adaptáveis da rede.

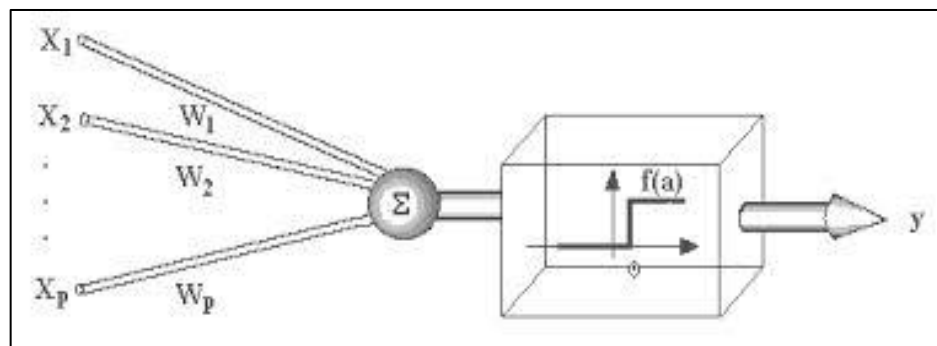
Para se definir bem uma RNA basta então o construtor desta definir o tipo de rede para resolver o problema e o algoritmo de treinamento. Lógico que também deverá ajustar os pesos da rede para solucionar o problema proposto.

Em RNA são utilizados os conhecimentos informáticos já existentes para criar uma similaridade, isso é, na maneira de processamentos dos dados e reconhecimentos de padrões ou tendências já existentes, para criar uma relação que se assemelha com funcionamento do cérebro humano. Essas técnicas proporcionam uma possibilidade de aprendizagem do programa, e com o tempo ele se torna mais eficiente devido aos vários treinamento baseados na base de dado (exemplos).

Quanto maior for sua base de dados, maior é a probabilidade com que o programa fique mais eficiente com os treinamentos.

Para maior detalhamento conceitual do modelo será demonstrado o funcionamento de uma RNA. Sejam $X_1, X_2, \dots, X_n$ as representações de entradas dos neurônios. Cada uma dessas entradas de neurônios possuirá um peso a serem atribuído, esses pesos serão $W_1, W_2, \dots, W_n$. Estas entradas juntamente com os seus respectivos pesos farão parte de um somatório que servirá de entrada para a função de ativação. A Figura 1 demonstra a explicação anterior.

Figura 1- Redes Neurais Artificiais

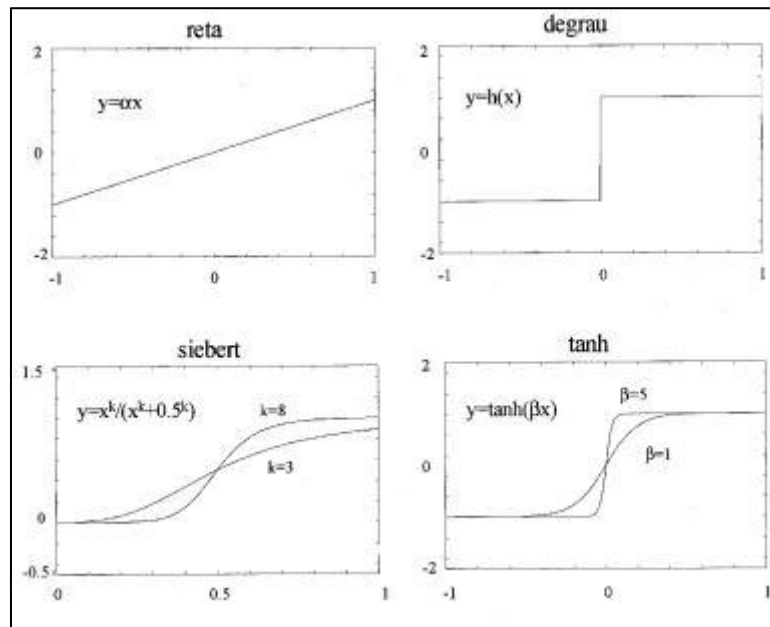


Fonte: BONAFE, 2012.

As funções de ativação estão relacionadas com o valor da saída do neurônio e podem ser de vários tipos, os mais comuns são: Função Linear, Função Escada, Função Sigmoidal. Para obter uma saída linear e contínua o recomendado seria a função linear, já se o desejável é uma saída binária a melhor opção seria a função escada e o último caso seria para uma saída contínua e não linear, neste caso o recomendável seria a função sigmoidal.

A função de ativação $f()$ em princípio pode ser qualquer função matemática, embora na prática exista um conjunto de funções que são as mais utilizadas. A figura 2 abaixo apresenta uma lista de algumas delas.

Figura 2- Funções de Ativação



Fonte: NEVES, 2008.

3.1.2. Tipos de aprendizagem

Esse tipos de redes apresentam vários tipos de aprendizagem, os mais comuns são aprendizagem supervisionado e a não supervisionado.

Em um aprendizado supervisionado, a Rede Neural é treinada com auxílio de um supervisor. Logo, a rede deverá possuir pares de entrada e saída, ou seja, um conjunto de entradas e um conjunto com as saídas desejadas para cada entrada. Toda vez que uma rede de entrada for apresentada, será verificado se a saída dessas respectivas entradas confere com a saída desejada. Caso isso não ocorra, a rede deverá ajustar seus pesos de forma que armazene o conhecimento desejado. Esse procedimento deverá ser repetido com todo conjunto de treinamento (entradas e saídas), para que a taxa de acerto alcance uma faixa consideravelmente satisfatória.

O aprendizado não supervisionado ocorre quando não há um agente externo indicando a resposta desejada para os padrões de entrada.

3.1.3. Rotulação de dados

Segundo Von Zuben (2008) dados rotulados são aqueles que assumem valores em um mesmo espaço vetorial multidimensional, e são acompanhados da classe a qual cada um pertence (rótulo). Pode existir múltiplas classes, com variâncias e número de dados distintos ou não para cada classe. Já dados não rotulados são aqueles que assumem valores em um mesmo espaço vetorial multidimensional, e que não se conhece a classe a qual cada um pertence, embora cada um pertença a uma classe específica. O número de classes pode ser conhecido ou não. A variância e o número de dados de cada classe pode ou não divergir.

3.1.4. Topologia e Modelos de redes neurais

Uma primeira classificação entre RNAs de acordo com Tatibana; Kaetsu (200?) é feita nos padrões de ligações que elas possuem. Assim se define praticamente dois tipos básicos de um RNA, o que apresentam pelo menos um ciclo em sua topologia e os acíclicos.

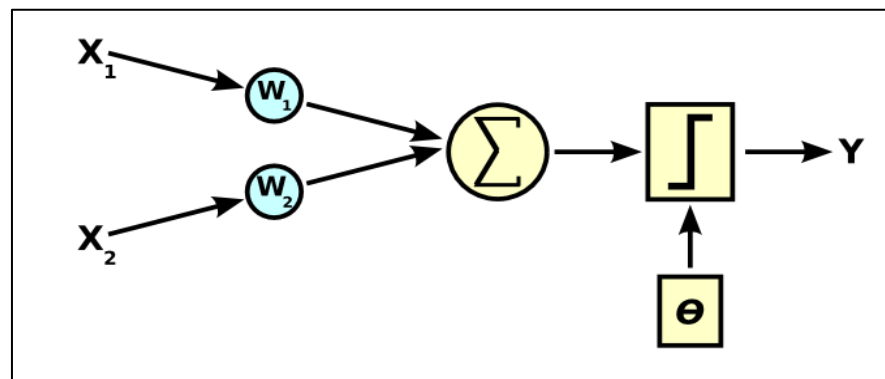
As topologias acíclicas, como o próprio nome já diz, não possuem ciclos ao longo de sua topologia. Então todos os sinais vão desde a camada de entrada para a saída destino sem existir ciclos, ou ligações entre os neurônios da mesma camada de rede neural. Esse tipo de topologia apresenta ainda uma subclassificação. São eles Perceptron camada única ou Perceptron de multicamadas. A Figura 4 representa um exemplo de uma camada única.

A classe de múltiplas camadas, normalmente são interconectadas em uma forma de alimentação avante. Isso quer dizer que cada neurônio em uma camada tem conexões diretas a neurônios da próxima camada. A Figura 5 ilustra bem este caso.

As topologias cíclicas apresentam pelo um ciclo fechado de ativação neural em sua estrutura. Podendo um nodo estar associado a outros nodos da sua mesma camada ou também a elementos de uma outra camada.

Os Perceptrons de uma camada são capazes de resolver somente sobre problemas linearmente separáveis, ou seja, que podem ser separados por uma reta em um hiperplano. Nota-se que apesar de suas limitações o Perceptron simples apresenta bons resultados para determinadas situações. Todas as redes de unidades paralelas ao ponto inicial podem aproximar toda a função contínua para um intervalo real limitado entre $[-1]$ e $[1]$. O Perceptron simples pode ser visto na Figura 3.

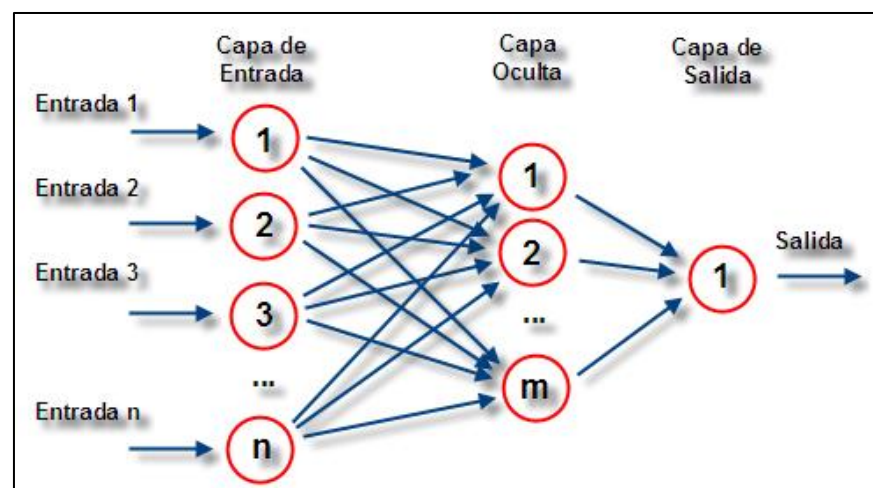
Figura 3 - Perceptron Camada Única



Fonte: BONAFE, 2012.

O Perceptron de Múltiplas Camadas (MLP) é uma rede com uma camada de entrada, que possui tantos nós de entrada quantos forem os sinais de entrada, uma ou mais camadas ocultas de neurônios e uma camada de saída com um número de neurônio igual ao número de sinais de saída. Diferentemente do Perceptron de camada simples, o de múltiplas camadas resolve problemas não linearmente separáveis.

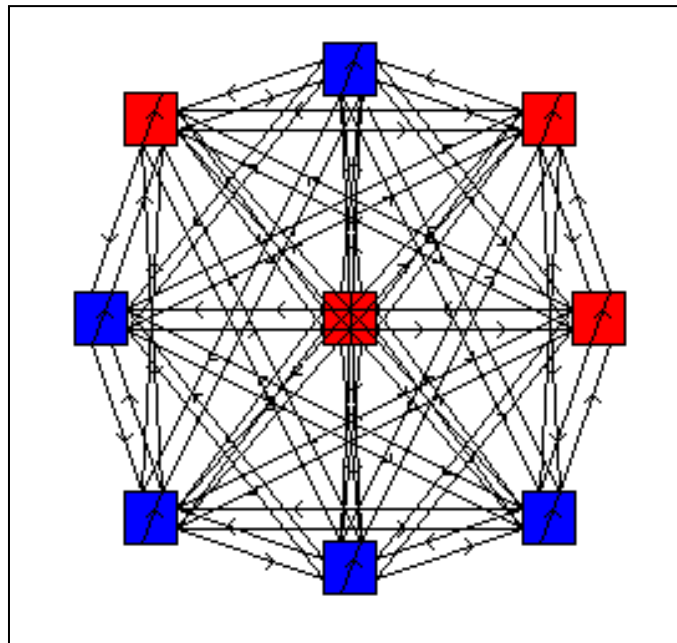
Figura 4 - Multicamada



Fonte: BONAFE, 2012.

Um modelo de redes neurais muito utilizado é o apresentado por John Hopfield em 1982, onde a rede pode ser descrita como uma rede autoassociativa, tendo uma única camada de neurônios totalmente conectada. Isso significa que apenas uma camada de nós existe e que todos estão interconectados.

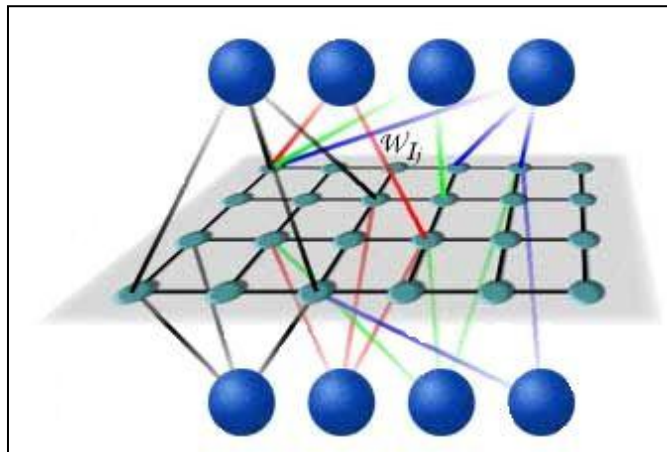
Figura 5 - Rede de Hopfield



Fonte: BONAFE, 2012.

Outro modelo de redes neurais famoso é o de Kohonen no qual é caracterizado por uma rede neural linear não supervisionada de camada única e totalmente conectada. A saída é organizada em uma ou duas dimensões, mas normalmente é considerada uma rede de duas dimensões. Isso não impõe em absoluto nenhuma forma topológica fixa, podendo então ser hexagonal, retangular, triangular, etc.

Figura 6 - Rede de Kohonen



Fonte: BONAFE, 2012.

3.1.5. Algoritmo de Treinamento Perceptron

Neste tópico será descrito dois algoritmos de treinamento, o primeiro a ser detalhado é Perceptron Simples e o segundo é o Backpropagation que treina a rede MLP.

Segundo Oliveira (2012) para o treinamento do algoritmo de Perceptron Simples pode ser descrito pelos seguintes passos no Quadro 1:

Quadro 1 - Algoritmo de Perceptron Simples

Inicialize os valores de w_i com pequenos valores randômicos.
 Enquanto não encontra uma condição de parada.
 Para cada par $\langle x, t \rangle$ (massa de treinamento) faça.
 Inicialize cada Δw_i com valor igual a 0 (zero).
 Execute para o vetor x na RNA e encontre a saída $\underline{q}$.
 Para cada w_i calcule Δw_i .

$$\Delta w_i = n (t - \underline{q}) x_i$$

 Para cada w_i calcule o novo valor.

$$w_i = w_i + \Delta w_i$$

Fonte: OLIVEIRA, 2012. 1

Onde as variáveis do Perceptron Simples são:

- Δw_i : Variação do peso mediante ao erro.
- w_i : Peso da rede.

- $\langle x, t \rangle$: Par de entrada e saída de uma amostra rotulada.
- n : Coeficiente de aprendizado.
- $\underline{q}$: Saída da rede.

O segundo quadro apresenta o Perceptron de múltiplas camadas (MLP).

A única diferença de variáveis existente entre o Simples e o de Múltiplas camadas é a variável h , que representa o valor de saída da camada intermediária.

Quadro 2 - Algoritmo de Backpropagation

```

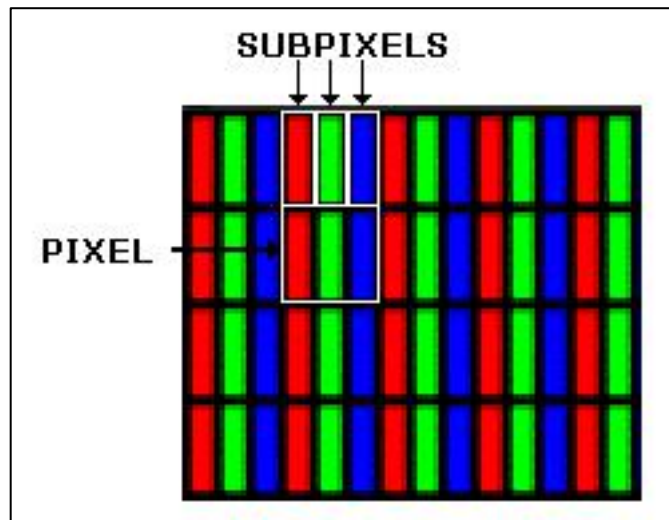
Inicialize os valores de  $w_i$  com pequenos valores randômicos.
Enquanto não encontra uma condição de parada.
  Inicialize cada  $\Delta w_i$  com valor igual a 0 (zero).
  Para cada par  $\langle x, t \rangle$  (massa de treinamento) faça.
    Execute para o vetor  $x$  na RNA e encontre a saída  $\underline{q}$ .
    Para cada  $w_i$  da camada de saída calcule  $\Delta w_i$ .
       $\Delta w_i = \underline{q} * (1 - \underline{q}) * (t - \underline{q})$ 
    Para cada  $w_i$  da camada de intermediária calcule  $\Delta w_i$ .
       $\Delta w_i = h * (1 - h) * \sum \Delta w_i * w_i$ 
    Para cada  $w_i$  da camada de intermediária recalcule.
       $\Delta w_i = w_i + n \Delta w_i * h$ 
    Para cada  $w_i$  da camada de saída recalcule.
       $w_i = w_i + n \Delta w_i * \underline{q}$ 

```

Fonte: OLIVEIRA, 2012.

3.2. Processamento digital de Imagens

Segundo o site Hardware (2005) um pixel representa cada ponto de uma imagem digital, seu nome vem de Picture Element. Cada um desses pixels, em um monitor colorido, são compostos por três pontos sendo eles um vermelho, um verde e um azul (padrão RGB). Cada um desses três pontos possui o tamanho de 8 bits, sendo então capazes de exibir cada 256 tonalidades diferentes, e se forem combinados suas tonalidades é possível obter 16 milhões de cores distintas. O pixel é a menor parte de uma imagem digital, a figura 7 a baixo mostra um exemplo de pixel.

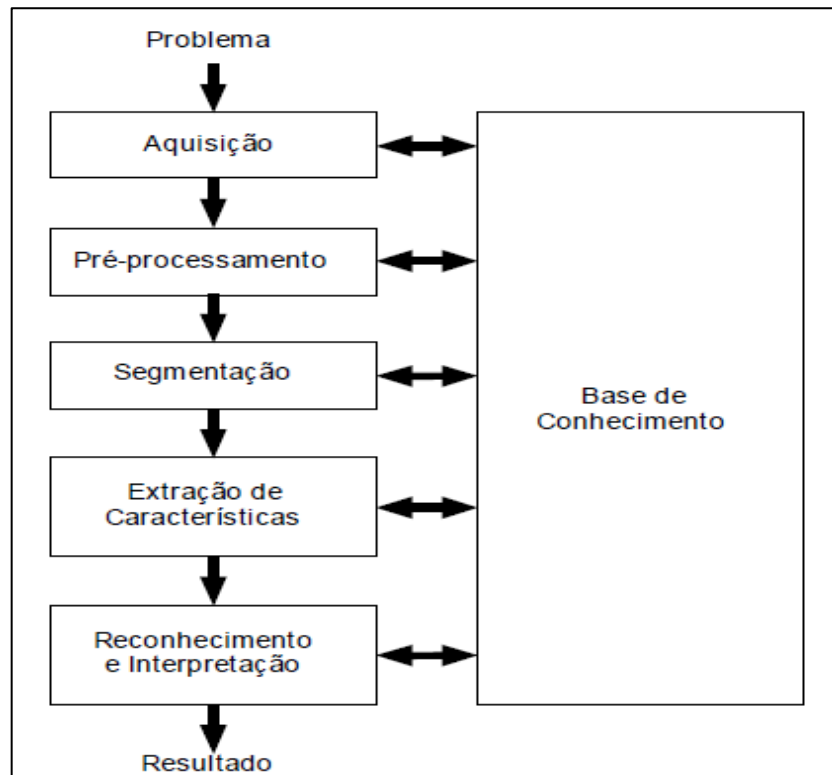
Figura 7 - Pixel RGB

Fonte: SANTIAGO, 2010

3.2.1. Estrutura de um sistema de visão Artificial

Segundo Marques Filho e Vieira Neto (1999) um Sistema de Visão Artificial (SVA) a partir de um sistema computadorizado, deve ser capaz de processar e interpretar imagens correspondentes a cenas reais. Uma demonstração de como um esquema SVA funciona esta representado na figura 8.

Figura 8 - Esquema SVA



Fonte: MARQUES FILHO, 1999.

O detalhamento das etapas do esquema SVA são apresentados nas próximas seções deste trabalho.

3.2.1.1. Aquisição de imagem e Pré-Processamento

A primeira coisa a se fazer é a obtenção de imagens. Para que isso ocorra são necessários um sensor e um digitalizador. Cada um deles fará um papel distinto, o sensor receberá a informação óptica e a converterá em um sinal elétrico, isto é, produzirá uma saída de sinal elétrico proporcional ao nível de energia detectado. Já o digitalizador fará uma conversão de imagem analógica para uma imagem digital, ou seja, que pode ser representada através de bits 0s e 1s.

A etapa de aquisição tem como função converter uma imagem em uma representação numérica adequada para o processamento digital subsequente.

Na etapa de Pré-Processamento ocorre uma melhoria na qualidade da imagem para as demais etapas. Isto porque na etapa anterior pode ocorrer imperfeições, tais como: pixel ruidoso, contraste ou brilho inadequado, caracteres

interrompidos, caracteres indevidamente conectados e outros. O resultado dessa etapa é uma imagem digitalizada de melhor qualidade, pois são realizadas operações de baixo nível que trabalham diretamente com os valores de intensidades dos pixels.

3.2.1.2. Segmentação e Extração de Caracteres

A principal tarefa da etapa de Segmentação segundo Marques Filho e Vieira Neto (1999) é a divisão da imagem em unidades significativas, ou seja, nos objetos de interesse que a compõem. Aparentemente é uma etapa de simples, mas sua implementação não é tão simples assim.

No caso específico do problema da Placa, é possível que ele seja dividido em duas etapas: em um primeiro momento os algoritmos de segmentação tentaram localizar a Placa do restante das informações. Posteriormente trabalhará sobre esta sub imagem, segmentando cada caractere individualmente. Segundo esta linha de raciocínio, este bloco produzirá à saída oito sub imagens, cada qual correspondendo a um caractere da Placa (L1, L2, L3, T, N1, N2, N3, N4).

A outra etapa é a de Extração de Caracteres, que tem como objetivo extrair características das imagens resultantes na segmentação (sub imagens), isso é, através das descrições permitir a caracterização dos dígitos com uma boa precisão. Deve apresentar também um bom poder de discriminação entre dígitos parecidos, como é o caso dos dígitos '5' e '6'. Essas descrições devem ser representadas por uma estrutura de dados devidamente adequada para algoritmo, possibilitando assim o reconhecimento.

Importante ressaltar que até este momento a entrada ainda é uma imagem, porém a saída é composta por um conjunto de dados relativos a esta imagem.

3.2.1.3. Reconhecimento, Interpretação e Base de Conhecimento

Na etapa de Reconhecimento, nada mais é do que a rotulação de um objeto baseando-se nas suas características, que foram traduzidas por suas descrições. Já

na etapa de Interpretação é dado o significado de cada caractere reconhecido, ou seja, quando é reconhecido o caractere “8” ele irá interpretar que este caractere é o número 8.

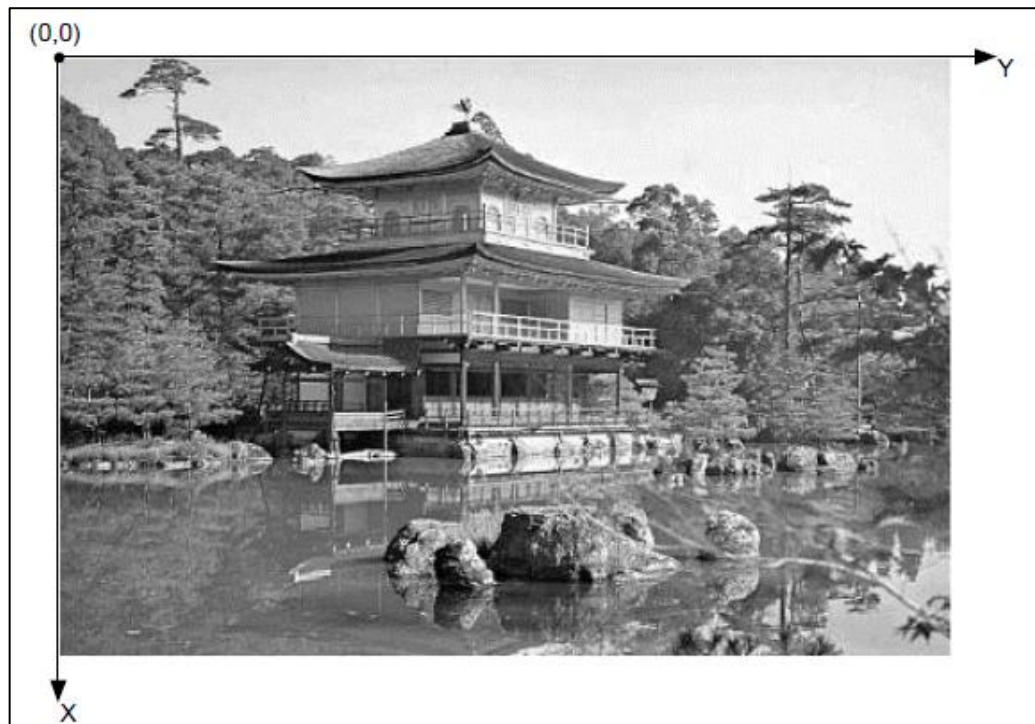
Todas as tarefas descritas acima pressupõem a existência de um conhecimento sobre o problema a ser resolvido, armazenado em uma base de conhecimento, cujo tamanho e complexidade podem variar. Idealmente, esta base de conhecimento deveria não somente guiar o funcionamento de cada etapa, mas também permitir a realimentação entre elas.

Por exemplo, se a etapa de representação e descrição recebesse 7 caracteres ao invés de 8, ela deveria ser capaz de realimentar a etapa de segmentação (provável responsável pela falha) para que esta procurasse segmentar novamente a sub imagem 'suspeita' (aquela de maior largura), buscando dividi-la em duas.

3.2.2. Propriedade de uma Imagem digital

De acordo com Marques Filho e Vieira Neto (1999), uma imagem monocromática pode ser descrita por uma função matemática $f(x,y)$ da intensidade luminosa. Sendo o seu valor, proporcional ao brilho (nível de cinza) da imagem em qualquer ponto (x,y) do plano espacial.

Figura 9 - Imagem Monocromática no eixo X,Y



Fonte: MARQUES FILHO, 1999.

Essa função $f(x,y)$ é o produto da interação entre a iluminância $i(x,y)$, isso é, que representa a quantidade de luz de incidência sobre o objeto e também as propriedades de refletância ou de transmitância próprias do objeto. Elas podem ser representadas pela função, cujo valor é a fração de luz incidente que o objeto transmitirá ou refletirá ao ponto (x,y) .

3.2.3. Digitalização

Do ponto de vista eletrônico, a digitalização consiste em uma conversão analógico digital na qual o número de amostras do sinal contínuo por unidade de tempo indica a taxa de amostragem e o número de bits do conversor A/D utilizado determina o número de tons de cinza resultantes na imagem digitalizada. (MARQUES FILHO; VIEIRA NETO. 1999)

A imagem analógica é convertida em uma matriz de M por N pontos, que na verdade são os pixels da imagem. Os valores de M e N estão diretamente ligados a resolução da imagem, quanto maior os seus valores maior será a resolução da imagem. A figura 10 demonstra essa matriz.

Figura - 10 Matriz dos pixels

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Fonte: MARQUES FILHO, 1999.

O processo de amostragem pode ser visto como uma divisão do plano xy em uma grade, com as coordenadas do centro de cada grade sendo uma dupla de elementos do produto cartesiano $Z \times Z$. Esse é o conjunto de todos os pares ordenados dos elementos (a, b) com a e b sendo números pertencentes a Z .

Portanto $f(x,y)$ é uma imagem digital se (x,y) forem números pertencentes a $Z \times Z$, e f seja uma função que atribua valores de níveis de cinza para cada par distinto de coordenadas. Se os níveis de cinza resultantes forem também números inteiros (como geralmente é o caso), Z substitui R e uma imagem digital então se torna uma função bidimensional cujas coordenadas e valores de amplitude são números inteiros.

3.3. Propriedades de uma imagem digital

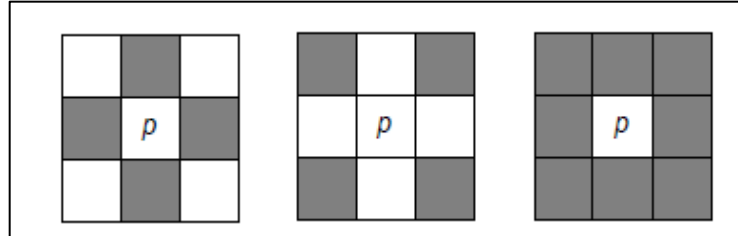
Uma imagem digital é uma imagem $f(x,y)$ discretizada tanto espacialmente quanto em amplitude. Uma imagem digital pode ser vista como uma matriz cujas linhas e colunas identificam um ponto na imagem, cujo valor corresponde ao nível de cinza da imagem naquele ponto.

3.3.1. Vizinhança e Conectividade

Um pixel pode ter vários tipos de pixels vizinhos, podem ser eles os quatros horizontais e verticais, podem também possuir outros quatros vizinhos diagonais. Por último pode se considerar a junção dos dois primeiros grupos de vizinhos

anteriores em um somente um grupo, tornando-se assim oito vizinhos. Os tipos de vizinhos de um pixel estão representados na figura 11.

Figura 11 - Vizinhos de um pixel



Fonte: o autor.

Para se estabelecer se dois pixels estão conectados segundo Wikipedia (2012), é necessário determinar se eles são adjacentes segundo algum critério e se seus níveis de cinza satisfazem a um determinado critério de similaridade. Por exemplo, em uma imagem binária, onde os pixels podem assumir os valores 0 e 1, dois pixels podem ser 4-vizinhos, mas somente serão considerados 4-conectados se possuírem o mesmo valor.

Considere um conjunto V de valores de tons de cinza utilizados para se definir a conectividade. Por exemplo, em uma imagem binária, $V = \{1\}$ para a conexão de pixels com valor 1. Para uma imagem de múltiplos tons de cinza, de conexão de pixels com valores de intensidade entre 32 e 64, $V = \{32, 33, \dots, 63, 64\}$. Como é conhecido o conceito de vizinhança e dado o conjunto V , pode ser definir alguns critérios de conectividade:

- "4-conectividade": dois pixels p e q com valores de tom de cinza contidos em V , são "4- conectados" se $q \in N4(p)$.
- "8-conectividade": dois pixels p e q com valores de tom de cinza contidos em V , são "8- conectados" se $q \in N8(p)$.
- "m-conectividade (conectividade mista)": dois pixels p e q com valores de tom de cinza contidos em V , são "m-conectados" se:
 - $q \in N4(p)$ ou
 - (ii) $q \in Nd(p)$ e $N4(p) \cap N4(q) \neq \emptyset$

3.3.2. Adjacência e Distância

A adjacência seria, como o próprio nome já sugere, a conectividade com os pixels adjacentes. Há vários critérios para definir adjacência entre os pixels, porém o mais geral seria: se dois conjuntos de subimagens I1 e I2 são adjacentes, se algum pixel de I1 é adjacente de algum pixel em I2.

Existem algumas maneiras de se calcular distância entre dois pixels. De acordo com a Wikipédia (2013), as mais comuns são: Distância Euclidiana, Distância D4 ou City-Block e a última seria a Distância D8 (Tabuleiro de xadrez).

- Euclidiana seria através da fórmula $D(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$
- D4 utiliza $D(p, q) = |x - s| + |y - t|$
- D8 utiliza $D(p, q) = \max(|x - s|, |y - t|)$ (onde max retorna o maior valor)

3.4. Histograma

Um histograma de uma imagem de acordo com CAMBRIDGE IN COLOUR 2013 simplesmente é o conjunto de números que indica a porcentagem de pixels desta mesma imagem que apresentam um determinado nível de tons em cinza.

Os valores dessas porcentagens geralmente são representados através de gráficos de barras, fornecendo assim para cada nível de cinza um valor numérico, seja este valor em porcentagem ou não, de pixels correspondentes na imagem. Através desse gráfico é possível obter o nível de contraste e também o brilho médio, e com isso define-se se a imagem é predominantemente clara ou escura. Os elementos do conjunto são calculados através de:

$$Pr(rk) = \frac{nk}{n}$$

Onde:

$$0 \leq rk \leq L-1$$

k = 0, 1, ..., L-1, onde L é o número de níveis de cinza da imagem digitalizada;

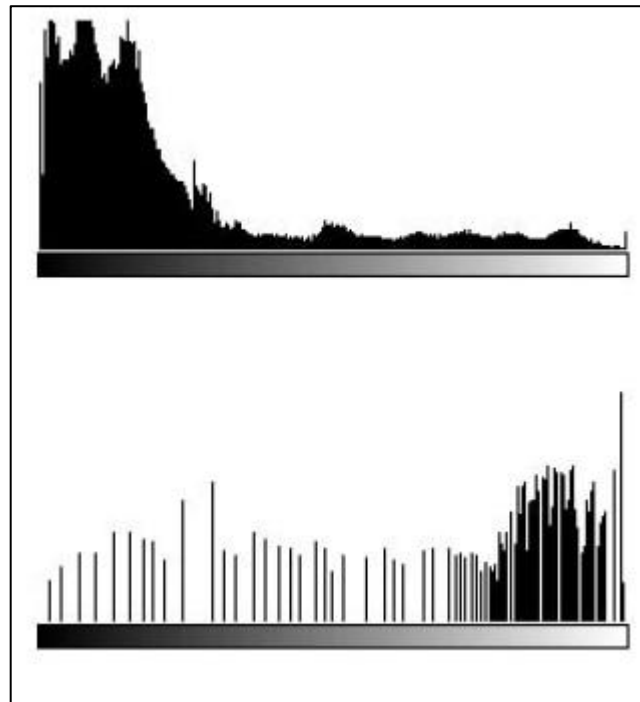
n = número total de pixels na imagem;

pr(rk) = probabilidade do k-ésimo nível de cinza;

nk = número de pixels cujo nível de cinza corresponde a k.

Na Figura 12 há dois exemplos de gráficos de histogramas.

Figura 12 - Exemplos de Histogramas



Fonte: Cambridge in Colour, 2013.

4. RESULTADOS

Este tópico é referente aos resultados que foram obtidos durante a realização deste projeto de conclusão de curso.

4.1. Coleta de Imagens

Inicialmente foi criada uma base de dados contendo diversas fotografias de veículos, sendo eles: carros utilitários, ônibus e caminhões. Além de possuir fotos de diferentes veículos, também existem uma divisão entre veículos de placas vermelhas e veículos de placas cinzas (normal).

Para a criação dessa base de dados foram realizados vários trabalhos de campo, com o intuito de obter fotografias da traseiras e frente dos veículos, sendo essa fotos de diferentes ângulos e distância. Houve também uma preocupação em obter fotos de placas que estavam velhas, desgastadas, em mal estado e novas,

isso para que o banco tivesse uma boa diversidade para ao realizar o treinamento do sistema. Abaixo algumas das fotografias que foram tiradas para a base de dados.

Figura - 13 Placa Cinza Nova



Fonte: o autor.

Figura 14 - Placa Cinza Desgastada



Fonte: o autor.

Figura 15 - Placa Com Vários Caracteres



Fonte: o autor.

Figura - 16 Placa Vermelha



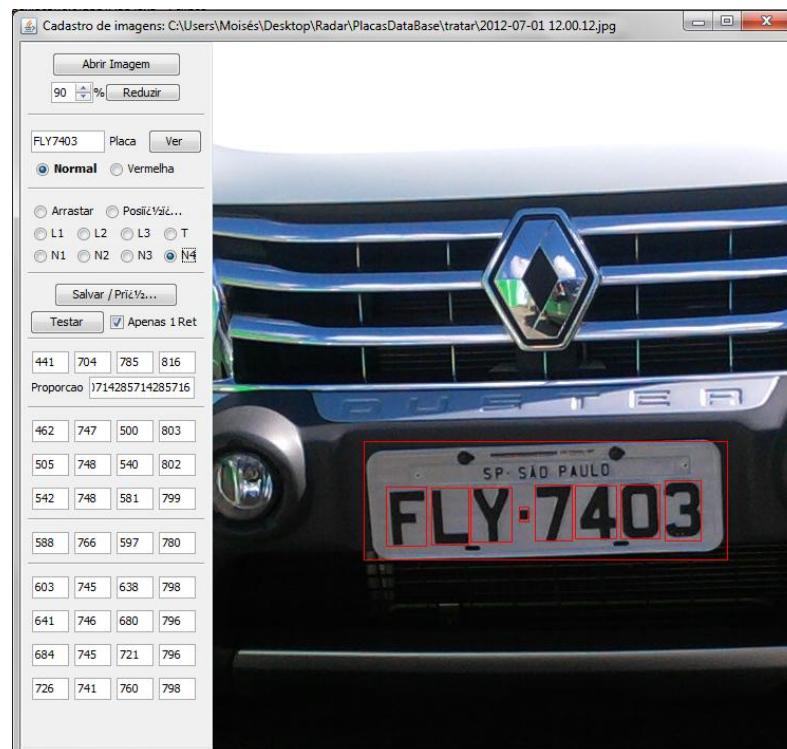
Fonte: o autor.

4.2. Obtenção massa de dados

A obtenção da massa de dados é fundamental como parte de um sistema de reconhecimento de padrões. Cada amostra pertencente a massa de dados necessita de algumas informações chaves, informações essas que são de grande importância para o treinamento dos padrões. Essas informações são: a imagem, posição da placa na imagem, posição dos caracteres na placa (letras, traço e números), tipo da placa (cinza ou vermelha), o significado alfa numérico de cada caractere da placa.

Foi necessário para este trabalho desenvolver um software de cadastro de placas, para que fosse possível cadastrar todas as amostras de placas a serem utilizadas no treinamento das RNA's. Muitas necessidades foram levantadas durante a definição deste software, isso para que ele pudesse facilitar o cadastro de placas para o usuário. As necessidades foram: o usuário ter facilidade ao manusear a imagem, uma forma fácil para definir a área correspondente à placa e os seus caracteres, definição do tipo de placa (vermelho ou cinza) através do usuário, permitir ao usuário definir a representação alfa numérica dos caracteres. Na Figura 17 é possível verificar todas as características do software.

Figura 17 - Software de Cadastro de Placas



Fonte: o autor.

Primeiramente o usuário clicaria em *Abrir Imagem* para carregar uma imagem de algum arquivo ou pasta, no caso desse projeto selecionamos imagens da pasta *Tratar*: Pasta esta que contém todas as imagens que ainda não foram tratadas pelo software de cadastro, e selecionamos uma imagem. Depois desse primeiro passo, o usuário entra com os caracteres da placa, exemplo: placa: FLY - 7403, cor: cinza. Essa etapa é responsável pela representação alfa numérica.

Após os passos anteriores, o usuário selecionaria entre as opções *Arrastar e Posição*. A primeira é para arrastar a imagem de maneira a posiciona-la da melhor forma possível para visualizar a região da placa. Já a segunda seria para identificar e definir a área da placa na imagem. Depois de selecionar a área da placa deve-se definir a área de cada caractere, selecionando um caractere por vez de acordo com as seguintes opções *L1, L2, L3, T, N1, N2, N3, N4*. A opção *L1* representa a área da primeira letra, *L2* e *L3* representam a área da segunda e terceira letra respectivamente. Para representar o traço e os números foi utilizado a mesma ideia, *T* para traço e *Ns* para os números. A Figura 18 mostra todas as áreas já selecionadas.

Para finalizar o usuário clicar em *Salvar*, ao fazer isso o software salvará a imagem juntamente com suas características no banco de dados. O software também retira essa imagem que foi selecionada da pasta inicial *Tratar* e a salva em uma outra pasta, no nosso caso o nome desta outra pasta é *base*, juntamente com um arquivo binário. Após várias repetições deste processo, será obtido uma base de dados com todas as características necessárias para efetuar o treinamento.

4.3.Extração de Características da imagem

O processo de extração de características consiste em conseguir capturar as informações necessárias da imagem para que seja possível utilizá-las em outras etapas do projeto. Para que fosse possível extrair as características de uma determinada imagem foram necessárias a aplicação de algumas técnicas relacionada ao conceito de imagem digital.

Primeiramente ocorre o redimensionamento da região referente à placa. Isto é necessário devido a falta de padrão no tamanho da placa. O tamanho de cada placa

foi padronizado e passou a ter 125 pixels de largura por 23 pixels de altura. OS caracteres também possuem tamanhos e largura distintos, novamente foi necessário redimensioná-los, para que o tamanho passasse para 36 pixels de largura por 21 pixels de altura. Em seguida ocorre a conversão, para tons de cinza, de cada imagem a ser treinada. Para realizar essa conversão, cada pixel foi transformado utilizando a formula a baixo e também os conceitos citados no tópico 3.3 deste trabalho.

$$x = \frac{R + G + B}{3}$$

Logo após a conversão para tons de cinza, a imagem passa pelo processo de remoção de ruídos. O processo é relativamente simples, calcula-se a média e o desvio padrão de todos os pixels da região da placa. Os pixels que possuem valores inferiores à media menos o desvio padrão são considerados ruídos, da mesma forma os valores superiores da media mais desvio padrão também são considerados ruídos. No primeiro caso, todos os pixels com valores inferior a media menos desvio padrão recebem exatamente o valor desse cálculo. Já no segundo, os pixels com valores maiores do que a media mais desvio padrão recebem o valor da media mais o desvio. Dessa forma diminuimos os ruídos da imagem.

Depois de diminuimos os ruídos é feito a normalização dos pixels. Isto é feito pegando o maior valor de todos os pixels, o menor valor e também o pixel a ser normalizado e os aplicam na fórmula de normalização a baixo:

$$N = \frac{X - m}{M - m}$$

- N é normalização.
- X é o valor a ser convertido.
- m é o menor valor coluna.
- M é o maior valor da coluna.

O resultado obtido é a representação dos pixels em tons de cinza em valores entre 0 e 1, sendo o valor 0 (zero) representando a cor preta e 1 representa a cor branca. Os valores próximos a esses limites representam os tons de cinza mais escuro e mais claro respectivamente. Para efeito de melhor visualização foi

realizado a inversão de polaridade dos pixels, ou seja, o valor zero agora representa a cor branca e o valor 1 a cor preta.

Após a normalização dos pixels são calculados o histograma vertical (com 125 valores) e o histograma horizontal (com 23 valores) das imagens em tons de cinza. Isso é feito para ter uma representação gráfica da placa. A Figura 18 representa os histogramas de uma placa.

Figura 18 - Histograma de uma placa



Fonte: O autor.

O próximo passo é a normalização dos histogramas. Neste procedimento é feito o cálculo de normalização para todos os valores contidos em cada coluna e linha do histograma, essa normalização coloca todos os valores no intervalo de 0 a 1. Para realizar esse cálculo pega-se o valor a ser convertido, o maior da coluna, e o menor valor da coluna e os aplicam na fórmula de normalização citada neste tópico.

Os valores obtidos pelos histogramas vertical e horizontal são utilizados para treinar a RNA responsável pela a localização da placa em uma imagem e a RNA responsável pela identificação da cor ou tipo da placa.

4.4. Treinamento das RNA's

Para se projetar uma rede neural são necessários definir alguns critérios essenciais, tais como: o tipo de rede, a topologia e tipo de aprendizado da rede (supervisionado e não supervisionado). Neste trabalho as redes de Hopfield e Kohonen (SOM) foram descartadas devido a existência de outras redes que trabalham melhor com dados rotulados.

Optou-se em utilizar uma rede neural MLP ao invés de um Perceptron simples, isto pelo fato de sua capacidade em solucionar problemas não linearmente

separáveis. Os problemas não linearmente separáveis se aproximam mais das condições reais enfrentadas, o que não é possível de se obter em um Perceptron simples.

Normalmente em problemas com grande volume de dados a rotulação das informações é inviável, seja por tempo ou por custo. Em nosso caso, apesar dos dados originais (fotografias) não serem rotulados, o software de cadastro que foi produzido tornou este processo de rotulação possível, rápido e não custoso.

Para cada vez que se inicia a execução dos treinos o código fonte faz uma busca no banco de dados, para carregar todas as imagens já cadastradas. Cada imagens carregada do banco é quebrada em varias outras subimagens, esse procedimento serve para trabalhar a localização da placa e a identificação da cor.

No caso da localização da placa, para cada nova mini-imagem atribui-se um peso de valor entre 0 e 1. O motivo pelo qual é feito essa atribuição é para saber se aquela mini-imagem faz parte o não da placa, ou seja, se uma determinada subimagem possui em sua composição uma parte da placa do veiculo correspondente a 10%, o peso atribuído a essa mini-imagem será 0,1. Caso não encontre nenhuma parte da placa o valor atribuído será 0, e será atribuído 1 para caso encontre 100% da placa na subimagem, dessa maneira pode-se validar os pesos de cada mini-imagem e assim saber a localização exata ou quase exata da placa. Neste trabalho optou-se por considera subimagens com valores de peso superiores a 0,8 como mini-imagens que contem a localização da placa, e para valores inferiores a 0,8 como subimagens que não contem a localização da placa.

Para a rna responsável por reconhecer a cor utiliza-se o mesmo procedimento citado acima com algumas mudança. Para cada subimagem agora o valor do peso atribuído será 0 ou 1, sendo 0 pra vermelha e 1 para cinza. No caso dessa rna em especial, é mais rápido descobri se a placa é vermelha ou não, pois com a analise da primeira subimagem já é possível responder o tipo da cor da placa.

4.4.1. RNA de reconhecimento da localização da placa.

Essa RNA apresenta 148 de neurônio na Camada de entrada. O tamanho da camada entrada para uma RNA deve corresponder com a quantidade de

componentes de uma amostra. Os componentes de uma amostra são todos dos valores encontrados nos histogramas vertical e horizontal de cada placa.

Optou-se em definir a mesma quantidade de neurônios utilizados na camada de entrada para a camada intermediária, ou seja, as duas possuem a mesmo numero de neurônios. Neste trabalho não foi feito uma investigação profunda e detalhada, para que fosse possível determinar qual é a quantidade de neurônios ideal a serem utilizados na camada intermediária. Isso será um trabalho futuro para esse projeto.

A camada de saída é representada por um único neurônio, que retorna valores entre 0 e 1. Sendo 0 (zero) para dizer que não é a placa e 1 para dizer que a placa foi encontrada. Considera-se zero todos os valores abaixo de 0,5, e considera-se 1 todos os valores maiores ou iguais a 0,5.

Uma representação da topologia desta rede pode ser assim visualizada na Figura 4 Multicamada, onde os valores de entradas são 148 e os valores da camada intermediária são 148.

Basicamente para executar um treinamento, a massa de dados é dividida em duas partes: massa de dados utilizada no treino e massa de dados utilizada para teste. Nesse projeto foi definido que tamanho da massa de treino seria $1/3$ (um terço) da massa de dados e que o tamanho da massa de teste seria $2/3$ (dois terços). Estas duas massas servem para dizer o quão especialista e o quão generalista respectivamente a rede se tornou. O objetivo dessa divisão entre massas de treinamento e de teste é alcançar o maior grau de generalização possível para rede. O treinamento se dá em épocas (quantidades), a cada época toda a massa de dados de treino é apresentada para a RNA, e onde seus pesos sinápticos são devidamente ajustados.

4.4.2. RNA de reconhecimento da cor ou tipo da placa.

A RNA para reconhecer o tipo da placa (vermelha ou cinza) é muito parecida com a RNA do item anterior, sendo a mesma quantidade de neurônios para a camada entrada e camada intermediaria. Também possui um único neurônio para a

camada de saída, cujo o valor de saída está entre 0 e 1. Sendo zero para representar a placa de cor vermelha e 1 para a placa de cor cinza.

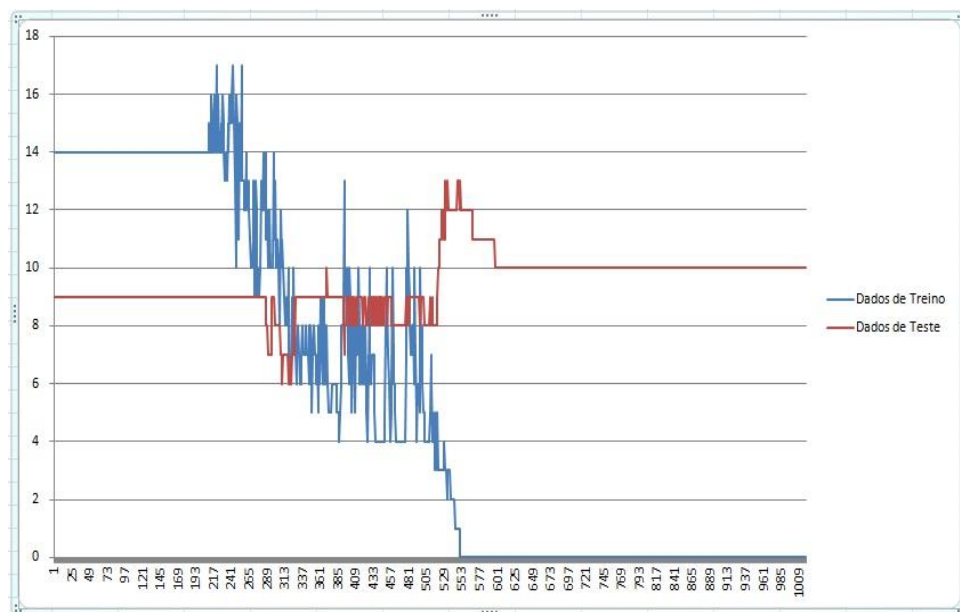
A mesma ideia de dividir a massa de dados em outra duas (treino e teste) foi utilizada nessa RNA.

4.4.3. RNA para a identificação caracteres.

Essa RNA apresenta uma pequena diferença com relação as duas outras. Nessa a quantidade de neurônios da camada de entrada e também da camada intermediária é dado pela multiplicação dos valores da largura e altura da imagem, e não foram utilizados histogramas para extrair as características. Isso foi feito para que fosse possível trabalhar com todos os pixels da imagem, e assim descobrir quais pertenciam ou não ao caractere em treinamento.

Gráfico demonstrado na Figura 20 representa o número de erros no treinamento e números de erros no teste. Nele é possível visualizar as curvas de erros de teste e erros de treino ao longo do tempo, sendo que a época ideal para se finalizar o treinamento dessa RNA seria quando o erro de teste atingisse o menor valor possível, ou seja em 6. Essa rede possui o percentual de aproximadamente de 90% de acertos.

Figura 19 - Gráfico da RNA de Reconhecimento de Caractere



Fonte: O autor.

5. CONCLUSÃO E TRABALHOS FUTUROS

O projeto seguiu com o planejado e conseqüentemente foram obtidos vários resultados contundentes e importantes. Dentre todos os subjetivos propostos por esse trabalho apenas um não foi possível cumprir, ***Treinamento de uma RNA para localizar a posição dos caracteres em uma placa***, isso devido a complexidade de implementar essa etapa e também a falta de tempo para o desenvolvimento do mesmo. Os demais objetivos, tais como: ***Geração da massa dados***, ***Extração de características de cada imagem***, ***Treinamento e teste de uma RNA para localizar a posição das placas em uma imagem***, ***Treinamento de uma RNA para identificação dos caracteres*** e ***Treinamento de uma RNA para identificar se uma placa é vermelha ou não*** foram concluídos com grande êxito por este projeto.

Todas as RNAs desenvolvidas neste projeto obtiveram altos índices de acertos, de 80% a 100%, o que mostra a qualidade e também a eficiência alcançada nessas redes neurais. O projeto apresentou várias dificuldades para sua realização, sendo as mais impactantes a complexidade do tema proposto e principalmente o tempo para a conclusão do trabalho. Apesar de todas as adversidades a equipe envolvida soube administrar bem os seus recursos disponíveis juntamente com o tempo restante para se trabalhar, o que possibilitou completar com êxito o maior número de objetivos possíveis.

Com relação aos trabalhos futuros esse projeto pode proporcionar alguns, dentre eles alguns se destacam, sendo eles: a RNA de treinamento da localização do caractere na placa, aplicação de novas redes neurais e técnicas para realizar um comparativo entre elas, realizar um estudo aprofundado para que seja possível determinar a quantidade ideal de neurônios a serem utilizados como entrada na camada intermediária, desenvolver um sistema de interface para tornar possível a comunicação entre os sistemas já existente e que também sirva para fiscalizar os veículos nas vias, ou seja, um sistema propriamente dito de fiscalização. Com esses possíveis trabalhos futuros o produto final poderá auxiliar na fiscalização do trânsito brasileiro, e também de uma forma indireta na reeducação dos condutores.

REFERÊNCIAS

- ANSARI, N. **Computational intelligence for optimization**. Boston: Kluwer Academic, 1997.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520: apresentação de citações em documentos**. Rio de Janeiro, 2002.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023: informação e documentação: referências: elaboração**. Rio de Janeiro, 2002.
- BONAFE, Douglas. **Introdução às Redes Neurais Artificiais (Perceptron, Adaline e Madaline)**. 2012. Disponível em: <<http://ccbits.blogspot.com.br/2012/10/introducao-as-redes-neurais-artificiais.html>> Acesso em: 13 sep. 2013.
- BRAGA, A. P. **Redes neurais artificiais: teoria e aplicações**. 2. ed. Rio de Janeiro: LTC, 2007.
- CAMBRIDGE COLOUR. **Histogramas**. 2013. Disponível em: <<http://www.cambridgeincolour.com/pt-br/tutorials/histograms1.htm>> Acesso em: 03 sep. 2013.
- COPPIN, B. **Inteligência artificial**. Rio de Janeiro: LTC, 2010.
- FERNANDES, A. M. R. **Inteligência artificial: noções gerais**. Florianópolis: Visual Books, 2003.
- Filipenses. In: A BÍBLIA: tradução ecumênica. São Paulo: Paulinas, 2002.
- GERSHENSON, C. **Artificial Neural Networks for Beginners**.
- HARDWARE. **Pixel**. 2013. Disponível em: <<http://www.hardware.com.br/termos/pixel>> Acesso em: 03 sep. 2013.
- HAYKIN, Simon. **Redes Neurais: princípios e prática**. Porto Alegre: Bookman, 2001.
- JANG, J. R.; SUN, C.; MIZUTANI, E. **Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence**. New York: Prentice-Hall Inc, 1997.
- NEVES, P. O. Rogério: **Introdução aos princípios conexionistas**. São Paulo, 2008.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**, Rio de Janeiro: Brasport, 1999. ISBN 8574520098.

OLIVEIRA, Douglas Nunes. **Notas de aula**. Timóteo 2012.

PRESIDÊNCIA DA REPÚBLICA CASA CIVIL (Brasil). **Código de Trânsito Brasileiro: LEI Nº 9.503, DE 23 DE SETEMBRO DE 1997**. Disponível em:
<http://www.planalto.gov.br/ccivil_03/leis/l9503.htm> Acesso em: 22 jul. 2013, 10:20.

RUSSELL, S.; NORVIG, Peter. **Inteligência Artificial**. 2. ed. Elsevier, 2004.

SANTIAGO, Camila. **Teoria da Cor**. Rio de Janeiro: 2010. Disponível em:
<<http://fau-do1.blogspot.com.br/2011/04/aula-cor-1.html>> Acesso em: 22 jan. 2013, 20:08

SILVA, J. D. S. **Uso de Redes neurais em visão computacional e processamento de imagens**. Knoxville: 2004.

Silva, J. M. N. **Rede Hopfield e Redes Estocásticas**. Niterói: 2003

UCI. **Machine Learning Repository**: Iris. Disponível em:<<http://archive.ics.uci.edu/ml/>> Acesso em: 09 ago. 2013.

VALENÇA, M. **Fundamentos das redes neurais**: exemplos em Java. Olinda: Livro Rápido, 2007.

VALLURU, B. R. **C++ Neural Networks and Fuzzy Logic**. New York: IDG Books Worldwide: 1995.

VON ZUBEN, F. J. **Modelos e Aplicações de Redes Neurais**

Artificiais (Parte III). Disponível em:
<ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ea072_2s08/notas_de_aula/topico1.2_08_Partell.pdf>