DEEP LEARNING E REDES NEURAIS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO

Diego Alves Rodrigues

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2018

Diego Alves Rodrigues

DEEP LEARNING E REDES NEURAIS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO.

Monografia apresentada ao curso de Ciência da Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Leonardo Vidal Batista

Novembro de 2018

Catalogação na publicação
Seção de Catalogação e Classificação

R696d Rodrigues, Diego Alves.
DEEP LEARNING E REDES NEURAIS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO / Diego Alves Rodrigues. - João Pessoa, 2018.
37 f.

Orientação: Dr Leonardo Vidal Batista, Me Arnaldo Gualberto de Andrade e Silva, Me João Janduy Brasileiro Primo.
Monografia (Graduação) - UFPB/CI.

1. aprendizagem profunda, redes neurais convolucionai. I. Dr Leonardo Vidal Batista. II. Me Arnaldo Gualberto de Andrade e Silva. III. Me João Janduy Brasileiro Primo. IV. Título.

UFPB/CI

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos **09** dias do mês de **Novembro** de **2018**, às **11:00** horas, em sessão pública na sala **do Laboratório Visio** do Campus V da Universidade Federal da Paraíba, na presença da banca examinadora presidida pelo professor(a) orientador **Dr. Leonardo Vidal Batista** e pelos professores **Me. Arnaldo Gualberto de Andrade e Silva** e **Me. João Janduy Brasileiro Primo**, o aluno **Diego Alves Rodrigues**, apresentou o trabalho de conclusão de curso intitulado: **DEEP LEARNING E REDES NEURAIS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO** como requisito curricular indispensável para a integralização do Curso de **Ciência da Computação**.

Após a exposição oral, o candidato foi arguido pelos componentes da banca que reuniram-se reservadamente, e decidiram, APROVAR a monografia, com nota 8,5. Divulgando o resultado formalmente ao aluno e demais presentes, eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

Dr. Leonardo Vidal Batista

Me. Arnaldo Gualberto de Andrade e Silva

Me. João Janduy Brasileiro Primo

Diego Alves Rodrigues

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

AGRADECIMENTOS

Gostaria de registrar o agradecimento aos meus pais, Pedro e Irene, pelo suporte dado no período da graduação, a minha esposa Elaine pela paciência e apoio durante os períodos de dificuldade. Um agradecimento especial ao meu Orientador, Prof. Dr. Leonardo Vidal Batista pelas orientações e compreensão ao longo dessa jornada.

Agradecer aos colegas e professores que ao longo dos anos foram presentes no dia a dia auxiliando e colaborando para o andamento do curso. Aos amigos que apoiaram de forma direta e indireta a minha graduação.

RESUMO

Controlar o tráfego rodoviário, seja em vias públicas ou privadas, as fronteiras, quem estaciona em um determinado estacionamento, ou quem está infringindo as leis de trânsito são exemplos de tarefas que exigem a identificação veicular, identificação essa feita por meio de placas de licenciamento automotivas reguladas por leis específicas. Para o processo de identificação de placas é necessário o uso de alguma técnica de Visão Computacional, campo este que tem ganhado bastante atenção da comunidade científica e das empresas ao longo dos últimos anos como forma de aumentar eficiência e cortar custos. Com o advento da Aprendizagem Profunda, soluções para os mais diversos problemas vêm sendo pesquisadas. Com as Redes Neurais Artificiais, especialmente as Redes Neurais Convolucionais, um salto foi dado em termos de resultados se comparados com técnicas tradicionais. As Redes Neurais Convolucionais se mostram extremamente bem-sucedidas em tarefas que envolvem principalmente classificação. Este trabalho buscou demonstrar a viabilidade de uma Rede Neural Convolucional para leitura de caracteres em placas de licenciamento veiculares, obtendo resultados de inferência na ordem de 89,24% de caracteres inferidos corretamente.

Palavras-chave: aprendizagem profunda, redes neurais convolucionais, reconhecimento, placas de licenciamento veiculares

ABSTRACT

Controlling road traffic, public or private, borders, parking lots, trespass traffic laws are examples of tasks that require license plate identification, being the license plates regulated by specific laws. For the license plate identification process, it is necessary the use of some computer vision technic, field that has gained enough attention of the scientific community and of the companies over the last years as a way to increase efficiency and to cut costs. With the advent of Deep Learning, solutions to the most diverse problems have been researched. With Artificial Neural Networks, especially of Convolutional Neural Networks, a jump in terms of results compared to traditional methods was given. Convolutional Neural Networks are extremely successful in tasks that mainly involve classification. This work aims to demonstrate the viability of a Convolutional Neural Network for reading characters in vehicular license plates, obtaining results on the order of 89.24% of correctly inferred characters.

Key-words: deep learning, convolutional neural networks, recognition, license plates

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

PDI	–	Processamento Digital de Imagens
ALPR	–	Automatic License Plate Recognition
ANN	–	Artificial Neural Networks
DNN	–	Deep Neural Network
DL	–	Deep Learning
CNN	–	Convolutional Neural Network

SUMÁRIO

1 Introdução

Com o advento das imagens digitais surgiu-se a necessidade imediata de técnicas para processar essas imagens, originando um novo campo de estudo, denominado de Processamento Digital de Imagens (PDI). O PDI é todo processo de análise e manipulação de imagens digitais, seja para identificar, extrair informações ou transformar uma imagem. O PDI engloba uma gama de softwares, hardwares e principalmente fundamentos teóricos.

Como exemplo das diversas aplicações e procedimentos do PDI, temos (QUEIROZ, 2003):

- Retificação e Restauração de Imagens, onde algoritmos são aplicados para minimizar as distorções e degradações dos dados que uma imagem possa vir a apresentar, com o objetivo de obter uma imagem mais fiel da cena;
- Realçamento de Imagens, onde o objetivo passa a ser o melhoramento de uma imagem, ou parte dela, para uma melhor visualização;
- Classificação de Imagens, que têm por finalidade a substituição da análise visual dos dados por técnicas quantitativas de análise automática, visando a identificação das regiões presentes na cena.

Com a evolução dos métodos de PDI e um melhor entendimento dos processos de manipulação de imagens um outro campo de pesquisa surgiu, a Visão Computacional. Segundo Ballard e Brown (BALLARD&BROWN, 1982), o campo da visão computacional pode ser entendido como o estudo do processo de extração de informação em imagens.

A Visão Computacional difere do PDI a medida que o PDI se preocupa prioritariamente com a relação imagem-imagem, onde os processos iniciam com uma imagem e resultam também em uma imagem, e a Visão Computacional se preocupa prioritariamente com a relação imagem-modelo, onde leva-se imagens a descrições matemáticas (modelos) que representam essas imagens, possibilitando a inferência ou extração de determinada característica que se deseja estudar.

O avanço das técnicas de visão computacional permitiu o desenvolvimento de sistemas para o problema do reconhecimento automático de placas de licenciamento, do inglês *Automatic License Plate Recognition* ou *ALPR*, permitindo com isso poupar capital humano valioso e permitindo também um monitoramento mais uniforme dada a quantidade crescente de veículos nas ruas.

Ter o controle dos veículos que trafegam em uma rodovia, atravessam as fronteiras, param em um estacionamento, seja ele público ou privado, é de uma importância estratégica para o controle de trafego em geral. Identificar veículos irregulares possibilita um controle e penalização, nos âmbitos administrativos e penais, de infratores e automatizar esse processo otimiza o custo e o tempo, tornando os serviços que tiram proveito dessa identificação mais eficazes.

Nos últimos anos, países como Inglaterra, França, Estados Unidos e Canadá implementaram sistemas de reconhecimento automático de placas de licenciamento e colocaram em produção sistemas reais de monitoramento e gerenciamento de tráfego (JIN *et al.*, 2012).

Redes Neurais Artificiais, do inglês "*Artificial Neural Networks"*, existem a bastante tempo (ROSENBLATT, 1958). Criadas pelo Ph.D. Frank Rosenblatt, as *ANNs* e suas posteriores derivações compõem hoje as diversas técnicas de Aprendizagem Profunda disponíveis e se baseiam no funcionamento dos neurônios no cérebro humano. As *ANNs* atraíram maior atenção da comunidade acadêmica e da indústria, nas últimas décadas, com o advento das Redes Neurais Profundas, do inglês "*Deep Neural Networks*", e tem transformado os mais diversos campos, especialmente os campos de Visão Computacional, Processamento de Sinais, Audio e Fala e Processamento de Linguagem Natural.

Aprendizado profundo, do inglês "*Deep Learning*", é uma das formas mais conhecidas de se referir às *DNNs*. Uma técnica de *DNNs* bastante poderosa e difundida são as Redes Neurais Convolucionais, do inglês "*Convolutional Neural Networks*", ou *CNNs*. Com os avanços das recentes arquiteturas de processadores e placas gráficas, ou *GPUs*, o maior poder computacional, a memória disponível e a acessibilidade a esses recursos, possibilitou o uso e desenvolvimento mais intensivo de soluções utilizando *CNNs*, que tem se mostrado promissoras na resolução de problemas de aprendizado, em especial os de classificação.

Aplicar o modelo de *CNNs* para resolver o problema do reconhecimento automático de placas de licenciamento surge como opção possivelmente viável as tradicionais técnicas de visão computacional e aprendizado de máquina existentes, e já utilizadas comercialmente, para resolver esse problema em específico.

1.1 Tema

Este trabalho tem por tema, como o título sugere, o *Deep Learning* com ênfase no uso das *CNNs* para resolver o problema do reconhecimento de caracteres em placas veiculares. Tendo como hipótese a possibilidade de se resolver o problema apenas com o uso de uma *CNN*, removendo-se a etapa de segmentação dos caracteres, comum para sistemas que buscam resolver o problema.

1.1.1 Objetivo geral

Demonstrar a viabilidade da resolução do problema de reconhecimento de placas de licenciamento veiculares utilizando apenas uma *CNN*.

1.1.2 Objetivos específicos

- Criar uma base de dados com placas de licenciamento veiculares brasileiras;
- Desenvolver uma CNN para processar classificação de dígitos de placas veiculares;
- Demonstrar a viabilidade do tema.

1.2 Estrutura do trabalho

O presente trabalho foi dividido em quatro seções. A primeira seção contendo uma visão introdutória do tema bem como uma contextualização da problemática, incluindo os objetivos e apresentação do trabalho.

A segunda seção trata dos conceitos e do referencial teórico do assunto, fornecendo conceitos essenciais para o entendimento das seções seguintes. A terceira seção abordando os resultados obtidos e discutindo esses resultados e a quarta seção tratando sobre trabalhos futuros e fazendo uma conclusão acerca do tema.

2 CONCEITOS GERAIS

Serão introduzidos nesta porção do trabalho alguns conceitos de relevância para o tema. Os termos utilizados, bem como seus acrônimos, serão em inglês, em virtude da vasta utilização nas publicações e artigos acadêmicos.

2.1 Perceptron e feedforward

Baseado nos trabalhos de Warren S. McCulloch e Walter Pitts (MCCULLOCH&PITTS, 1943), em 1958, Frank Rosenblatt (ROSENBLATT, 1958) criou um conceito básico baseado no funcionamento dos neurônios humanos, o *Perceptron*.

Basicamente, um *Perceptron* realiza uma operação sobre uma ou mais entradas x_i, cada entrada está associada a um peso w_i, é realizado o somatório do produto escalar de $x_i . w_i$ e a este somatório é adicionado um bias b. O resultado obtido é então adicionado a uma função de ativação σ conhecida como função degrau, ou função de Heaviside, que produz como resultado 0, se o resultado for menor ou igual a 0 ou 1 se o resultado for maior que zero.

$$f(x) = \begin{cases} 0, & b + \sum_i w_i . x_i \leq 0 \\ 1, & b + \sum_i w_i . x_i > 0 \end{cases}$$

Dada a saída binária, um *perceptron* é considerado um classificador binário e também a menor rede neural do tipo *feedforward*. As redes neurais do tipo *feedforward* são redes neurais que se propagam apenas em uma direção, ou seja, os inputs entram nos *perceptrons* e o sinal se propaga para a saída em uma única direção, não retornando o sinal para a entrada.

ANNs normalmente possuem multiplas camadas, sendo a camada de entrada, a camada de saída e camadas ocultas que se localizam entre a entrada e a saída. Uma rede *feedforward* também pode ser multicamadas, o que significa que temos uma ou mais camadas, onde a saída de uma camada é a entrada da outra imediatamente posterior hierarquicamente. Um exemplo de uma rede *feedforward* pode ser vista na figura abaixo:

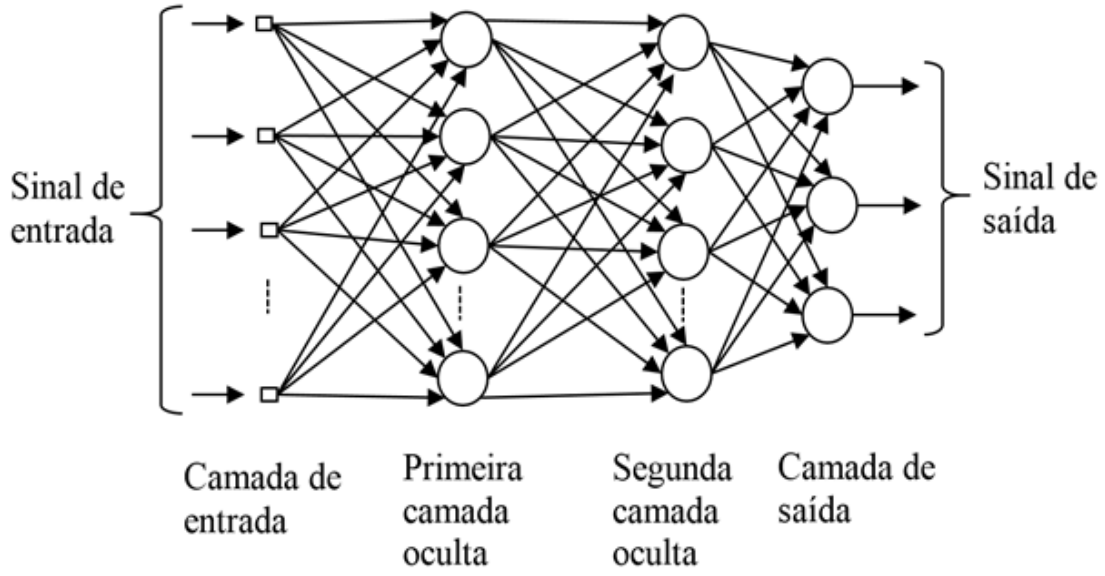


Figura 1 - Exemplo de uma rede feedforward (Fonte: NASCIMENTO&OLIVEIRA, 2016)

2.2 Função de erro, gradiente descendente e *backpropagation*

As *ANNs* aprendem através do processo de ajuste dos pesos das entradas em cada camada, para o ajuste desses pesos necessitamos definir uma forma de calcular o quão próximo um resultado está do correto, e para isso se faz necessária uma função de erro, que normalmente mede o quão imprecisa a rede foi dado um resultado esperado.

Funções de erro incluem *Mean Square Error*, bastante utilizada em problemas de regressão e a *Cross-Entropy*, que para problemas de classificação é comumente utilizada (MCCAFFREY, 2013).

Para o cálculo da *Cross-Entropy*, ou entropia cruzada, utilizamos a seguinte equação, abaixo discriminada, onde w é a matriz de pesos, y_i e t_i o resultado inferido e o correto, respectivamente, e n é o número total de itens de treinamento.

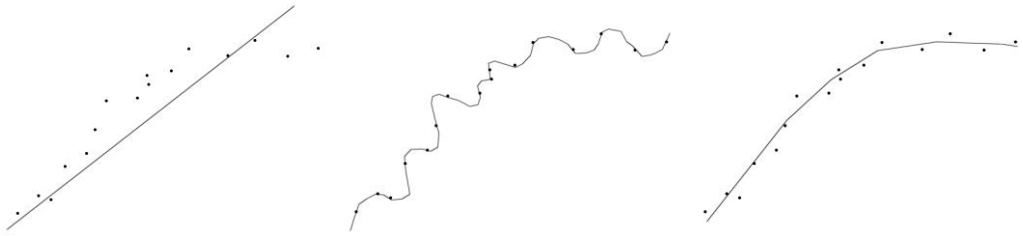
$$E(w) = \frac{1}{n} * \sum_{i=1}^{n} [t_i * \log(y_i) + (1 - t_i) * \log(1 - y_i)]$$

Como dito, precisamos ajustar os pesos para atingirmos o resultado desejado e o algoritmo mais utilizado para realizar tal tarefa, quando temos a informação da entrada e da saída desejada, é o *backpropagation* com gradiente descendente. Uma

forma generalizada do algoritmo é que ele funciona indo no caminho inverso ao dos dados, indo da saída e retornando camada a camada da rede neural, ajustando os valores dos pesos w, com isso é possível a rede aprender padrões sobre os dados, e ao entrar na rede uma entrada nunca vista, a rede será capaz de reconhecer os padrões aprendidos anteriormente, mesmo que a entrada possua ruído (BARCA *et al*, 2005).

No gradiente descendente temos que, de forma iterativa, os valores dos pesos são alterados pelo *backpropagation* seguindo um passo fixo, chamado de taxa de aprendizagem. O objetivo do gradiente descendente é minimizar a função de erro para o menor erro local, necessitando de um cuidado na escolha da taxa de aprendizado, pois ela impacta diretamente no tempo necessário para se chegar a esse mínimo.

Se a rede não generalizar o suficiente e aprender os dados de entrada utilizados no processo de treino da rede (processo pelo qual os dados são inseridos e se indica a saída válida), temos o que chamamos de *overfitting*. No caso de a rede generalizar demais e não detectar os padrões de forma correta, chamamos de *underfitting*.



(a) (b) (c)

Figura 2 - Exemplos de (a) underfitting, (b) overfitting e (c) boa generalização (Fonte: PAPAGELIS&KIM)

O algoritmo atua na rede em conjunto com o *feedforward*, onde na ida é calculado o erro e na volta os pesos são ajustados. Uma observação do algoritmo do gradiente descendente é o fato de ele encontrar mínimos locais da função de erro, o que pode levar a resultados considerados pela rede como ótimos, mas que não são o melhor que poderia ser atingido (mínimo global).

2.3 Função de ativação

Funções de ativação são uma importante parte das *ANNs*, pois são elas que determinam a saída de cada perceptron (ou neurônio). O uso de funções lineares como função de ativação limita o poder das *ANNs* a soluções convexas. Para soluções não-convexas, funções não lineares são necessárias. As principais funções utilizadas podem ser vistas a seguir (FACURE, 2017):

- Função sigmoide:

Conhecida também como função logística, a função sigmoide é dada pela equação:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Ela assume valores apenas entre 0 (não ativação) e 1 (ativação). Comportamento bastante parecido com os próprios neurônios biológicos, que são ativados ou não dadas as entradas que recebe.

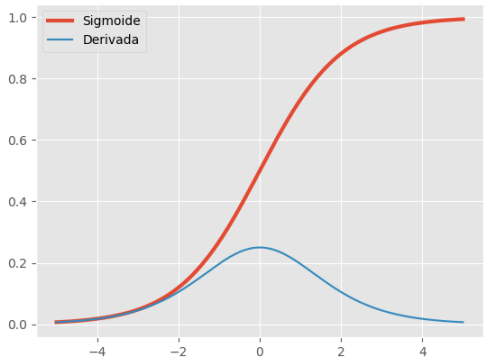


Figura 3 – Gráfico da função sigmoide (Fonte: FACURE, 2017)

- Linear retificada (ReLU):

Retorna 0 ou um valor real maior que zero. Semelhante a função identidade, difere das funções anteriores, onde a propagação do gradiente desvanece nas regiões de cauda, e se mostra bastante fácil de se otimizar. É uma alternativa bastante utilizada na prática. Sua fórmula é:

$$ReLU(x) = \max(0, x)$$

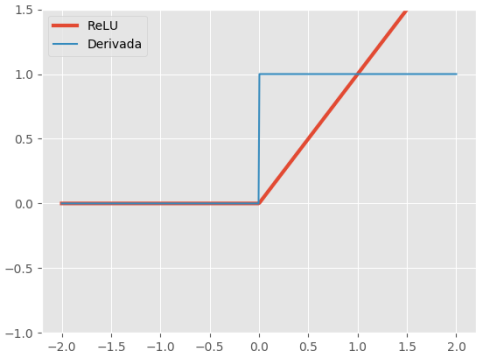


Figura 4 – Gráfico da função ReLU (Fonte: FACURE, 2017)

2.4 Momentum

O termo momentum é uma adição feita ao cálculo do gradiente descendente, onde o termo é uma constante que define o quão a mudança de pesos anterior afeta a mudança de pesos atual. Possui valor entre 0 e 1, onde 0 significa que a mudança depende apenas da função de erro e 1 depende apenas da mudança anterior do peso.

Na prática, o termo momentum é usado para impedir que o gradiente descendente fique preso em mínimos locais, o que minimiza uma das desvantagens do algoritmo.

2.5 Dropout

O dropout é uma técnica utilizada para aumentar a quantidade de padrões que a rede detecta de uma mesma entrada. O processo se dá realizando a remoção aleatória de neurônios das camadas ocultas durante o treinamento de um mini lote de entrada, onde os neurônios são reinseridos ao final do processamento do mini lote. A técnica ajuda a evitar o *overfitting*. Uma representação da técnica pode ser vista abaixo:

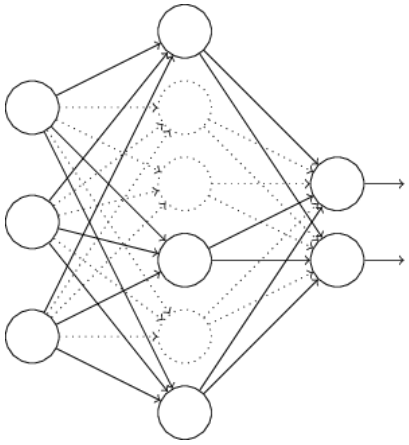


Figura 5 - Exemplo de dropout (Fonte: DLB)

2.6 *Data augmentation*

Data augmentation é um método de regularização bastante importante para impedir o *overfitting*. Nem sempre possuímos um conjunto de treinamento grande o suficiente para que a rede aprenda os padrões que desejamos, tornando difícil a resolução de um problema com bancos de dados de treino pequenos. No processo de *data augmentation* utilizamos diversas técnicas para aumentar os dados de forma artificial como: variação de brilho, saturação, zoom, reflexão horizontal e vertical, recorte, entre outros métodos.

Com esta técnica conseguimos melhorar os resultados da rede sem adicionar nenhuma característica nova a ser aprendida, reduzindo de forma drástica o *overfitting*, tornando a rede mais robusta.

2.7 Convolução

Matematicamente, a convolução é um operador linear que recebe duas funções como entrada e retorna uma terceira função originária do somatório da multiplicação da função *kernel* na região superposta da função alvo pelo deslocamento do *kernel* sobre ela. A formula da convolução no domínio discreto é a seguinte:

$$(f * g)(k) = h(k) = \sum_{i=0}^{k} f(i) \cdot g(k - i)$$

f e *g* são sequenciar numéricas de tamanhos iguais ou variados, e a função retorna o k-ésimo elemento do somatório da multiplicação.

2.8 Redes Neurais Convolucionais

Do inglês *Convolutional Neural Networks* (*CNNs*), as *CNNs* são um tipo específico de *ANN*, proposta pelo pesquisador francês Yann LeCun (LECUN, *et al.*, 1998). As *CNNs* se mostraram, desde a sua criação, serem muito eficazes para resolver problemas de classificação, se mostrando uma alternativa viável aos métodos tradicionais para esse tipo de problema.

Uma das desvantagens das *CNNs* é o fato de existir a necessidade de uma grande quantidade de dados rotulados para a extração dos padrões, ou *features*. Basicamente, para extração dessas *features*, podem existir três componentes básicos em uma *CNN*, camada de convolução, *pooling* e rede totalmente conectada. Qualquer arquitetura básica é composta por esses blocos. Um exemplo da rede criada por Yann LeCun, a *LeNet* pode ser visto abaixo:

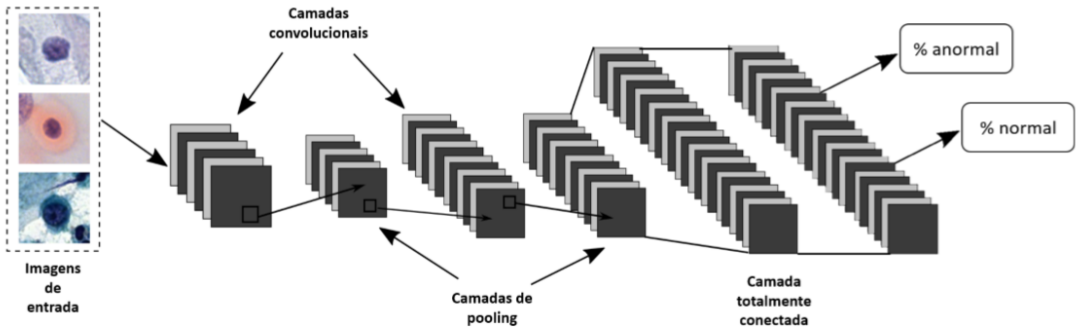


Figura 6 - Exemplo da arquitetura LeNet para o problema de classificação de células normais e anormais (Fonte: ARAÚJO et al., 2017)

2.8.1 Camada convolucional

A camada convolucional em uma *CNN* é responsável por extrair as chamadas *features* da entrada. O processo de extração dessas *features* se dá por meio de filtros convolucionais de tamanhos reduzidos, onde os filtros percorrem os dados de entrada

em largura, altura e profundidade (chamada de dimensão) realizando a operação de convolução sobre os dados.

A cada processamento de entrada no período de treinamento da rede, os filtros vão sendo ajustados de tal modo a disparar quando a entrada contiver uma determinada característica comum aos lotes de entrada, como por exemplos, arestas, cores, dentre outras características.

Com o andar da entrada na rede, os filtros vão aprendendo estruturas cada vez mais complexas, ou seja, quanto mais filtros convolucionais, mais *features* extraímos da entrada, porém isso tem um custo de memória e processamento, o que precisa ser balanceado na hora de definir a arquitetura.

2.8.2 Camada de pooling

A camada de *pooling* consiste basicamente em uma camada destinada a reduzir o tamanho dos dados de entrada. Normalmente, após uma camada convoluconal utilizamos uma camada de *pooling*, com isso as próximas camadas de convolução receberão uma outra forma de representação dos dados, possibilitando a rede aprender diversas representações dos dados, evitando com isso o *overfitting* (KARPATHY).

Uma técnica bastante utilizada para executar o *pooling* é chamada de *max-pooling*. Nessa técnica, reduzimos subpartes dos dados originais pelo maior valor encontrado nessas sub regiões, reduzindo com isso o tamanho da imagem por um fator de filtro $m\ x\ n$.

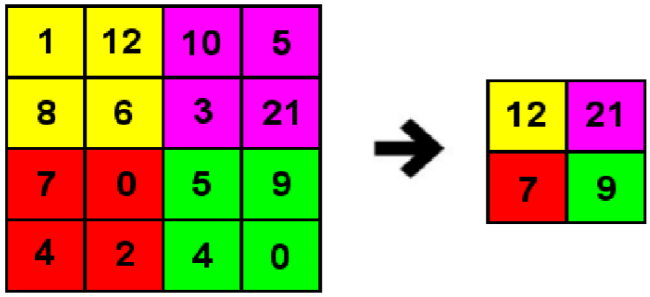


Figura 7 - Exemplo do funcionamento do max-pooling com filtro 2x2 em uma imagem 4x4 (Fonte: FERREIRA, 2017)

A técnica de *pooling* também beneficia a rede neural, ao passo que reduz a quantidade de dados para a camada seguinte e melhora a regularização da rede, reduzindo o custo de memória e processamento.

2.8.3 Camada totalmente conectada

As camadas totalmente conectadas normalmente se situam ao final da rede. Nessas camadas os *features* extraídos nas camadas de convolução anteriores são utilizados para se ter a saída de classificação da rede.

Um exemplo de arquitetura com as três camadas básicas pode ser encontrado abaixo:

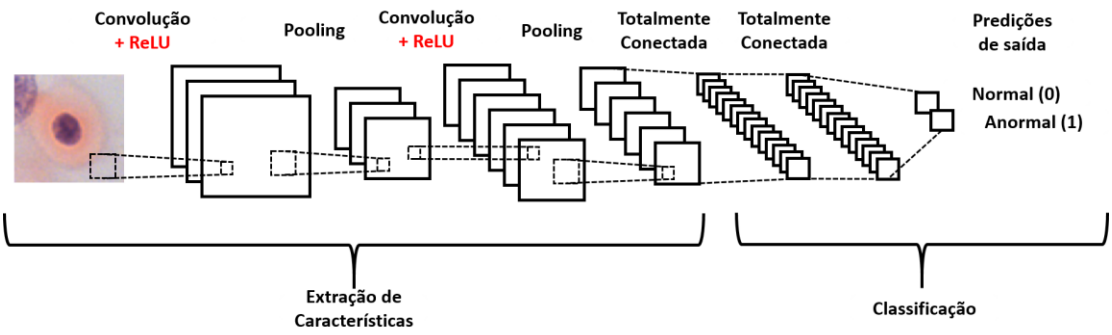


Figura 8 - Exemplo de arquitetura, baseada na LeNet, utilizando as camadas básicas de convolução, pooling e totalmente conectadas (Fonte: ARAÚJO et al., 2017)

3 METODOLOGIA

O presente trabalho seguiu uma abordagem metodológica qualitativa. Os resultados serão apresentados sob a forma de conclusões deles extraídas. Para a obtenção dos referidos resultados, alguns passos foram seguidos e serão discriminados a seguir.

3.1 Obtenção do banco de placas veiculares

Como primeiro passo, decidiu-se que o banco de imagens de placas seria obtido de forma manual, ou seja, um banco de imagens próprio seria criado e para isso foi escolhida a UFPB, Campus I, para a tomada das fotos. Por ser um local com um enorme número de veículos e estacionamentos, a UFPB possibilitaria uma rápida aquisição das fotos dos veículos.

Para a etapa de obtenção do banco de dados foi aberto o processo de número 23074.010615/2018-13 junto a Prefeitura Universitária da Universidade Federal da Paraíba. No processo, foi solicitada a autorização para fotografar os veículos nos estacionamentos do Campus I. Com a devida autorização em mãos o processo de tomada das imagens foi iniciado.

Para a aquisição das imagens foi utilizada uma câmera da marca Canon, modelo Rebel T2i, configurada para modo manual com os seguintes parâmetros definidos:

- **ISO auto**: Máx.:400
- **Comp.exp./AEB**: 0
- **WB SHIFT/BKT**: 0,0/±0
- **Disparo flash**: Desactivar
- **Qualidade**: L

As configurações acima buscaram zerar ao máximo as configurações de melhoramento de iluminação para que as imagens fossem capturadas com a maior fidelidade possível.

Figura 9 - Exemplo de fotos de veículos retiradas durante a fase de montagem do banco de dados (Fonte: O Autor)

Foram retiradas 800 imagens para compor o banco em 2 turnos. Por restrições de privacidade exigidas pela Prefeitura Universitária para liberação das fotos, apenas o veículo do autor foi mostrado como exemplo.

3.2 Obtenção dos recortes das placas veículares

Para o recorde das placas foi utilizada uma ferramenta online chamada Supervisely. As imagens dos veículos tiveram suas placas segmentadas de forma manual através da ferramenta online, disponibilizada de forma gratuita para pesquisa, sendo exportadas no formato JSON.

Após o processo de segmentação manual, um script desenvolvido pelo autor utilizando a linguagem Python foi executado em cada uma das 800 imagens, recortando as placas das imagens através das marcações feitas no Supervisely.

3.3 Processo de padronização das imagens

Após a obtenção dos recortes das placas, as imagens foram convertidas para escala de cinza e submetidas a um processo de equalização de histograma. Para esta tarefa, um script desenvolvido pelo autor, também na linguagem Python, utilizou uma biblioteca de manipulação de imagens de código aberto, a OpenCV.

Figura 10 - Imagens de placas após a conversão para escala de cinza e equalização de histograma (Fonte: O Autor)

3.4 *Data augmentation*

Conforme citado no referencial teórico, *CNNs* necessitam de grande quantidade de dados de entrada para o treinamento, e o tamanho do banco de dados com 800 imagens foi dividido inicialmente em 100 imagens para teste e 700 imagens para treinamento, sendo considerado um número muito baixo de imagens para treinamento.

Técnicas de *data augmentation* se mostraram úteis, onde aumentou-se o banco de imagens em 4 vezes o seu tamanho original, de 800 para 3200 imagens. Ainda um número baixo, mas para o que o trabalho se propõe, satisfatório.

3.5 Arquitetura da rede

A seguinte arquitetura de *CNN* foi implementada apara efetuar a classificação das placas em:

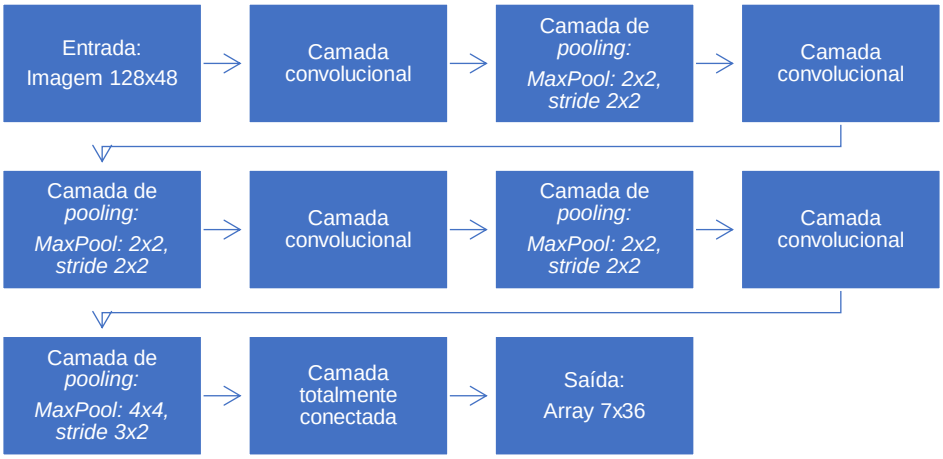


Figura 11 - Arquitetura de CNN implementada (Fonte: O Autor)

Na arquitetura acima, vemos 4 camadas de convolução, cada uma com 1 camada de *pooling* adjacente onde a primeira recebe a imagem em seu tamanho original, 128x48, e a última reduz a imagem a um tamanho de 6x1.

3.6 Treinamento da rede

Para dar celeridade ao processo de treinamento da rede foi-se utilizada uma placa gráfica (*GPU*) da marca EVGA com chipset NVIDIA 1080 TI com 11GB de memória GDDR5X, processador Intel i7 4790k e 16 GB de memória RAM DDR3 da marca Corsair, modelo Vengeance Pro. O sistema operacional utilizado foi o Microsoft Windows 10 Pro.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

O presente trabalho obteve resultados satisfatórios para o que se propunha. Como objetivo inicial, de demonstrar a viabilidade da resolução do problema de reconhecimento de placas de licenciamento veiculares utilizando apenas uma *CNN*, o trabalho obteve como resultado uma taxa média de acerto de caracteres de 89,24% e de placas inteiras de 49,22%.

Uma normalização do banco foi necessária, onde algumas placas que levavam a casos isolados como, por exemplo, placas de carros de aluguel que por serem vermelhas levavam a poucos casos com caracteres de cor branca, foram removidas do banco de teste e treinamento.

Figura 12 - Exemplo de placas de transito vermelhas removidas do banco de imagens de teste e treinamento (Fonte: O Autor)

Placas com alto grau de rotação também foram retiradas do banco de treinamento e teste, visto que a baixa ocorrência interferiu no treinamento da rede. Outros problemas como sombra parcial em demasiado ou alto grau de luminosidade natural foram fatores que interferiram bastante no treinamento e aumento da precisão nas inferências, visto que mesmo com o pré-processamento com equalização de histograma, permaneceu-se o problema.

Placas com letras pouco vistas também foram motivo de erros de inferência nos caracteres. A rede teve um grande sucesso no acerto de caracteres numéricos visto que a possibilidade é de 10 caracteres e a disponibilidade de caracteres numéricos nas placas para treinamento da rede foi vasta.

Figura 13 - Exemplo de placas removidas dos bancos de treinamento e teste com alta rotação e iluminação não uniforme (Fonte: O Autor)

Ao final do processo de normalização restou-se uma base de dados total de 2820 imagens. Para a obtenção dos números de acerto apresentados previamente, uma técnica conhecida como validação cruzada do tipo *k-fold* foi utilizada. Foi-se utilizado apenas os grupos de treinamento e de teste, o grupo de validação foi ignorado.

As 2820 imagens foram divididas em 10 grupos mutuamente excludentes de 282 imagens ($k = 10$), foi-se realizado o primeiro treinamento (chamada de primeira iteração pelo autor) utilizando o grupo (*fold*) 1 como teste e o restante como treinamento. Posteriormente cada grupo foi, de forma iterativa, assumindo a posição de grupo de teste até o grupo 10. Os resultados obtidos podem ser encontrados na tabela a seguir:

Tabela 1: Resultados obtidos após $k = 10$ testes

	% de acerto de caracteres	**% de acerto de placas**	**Custo de treinamento**
1ª Iteração	93,4	57,09	0.3544
2ª Iteração	84,54	33,69	0.5532
3ª Iteração	90,33	52,83	0.5363
4ª Iteração	85,28	40,07	0.5414
5ª Iteração	96,18	76,24	0.611
6ª Iteração	85,71	34,04	0.3175
7ª Iteração	89,07	48,22	0.6368
8ª Iteração	93,27	65,25	0.3501

9ª Iteração	83,61	32,98	0.557
10ª Iteração	90,96	51,77	0.3008
Média	89,24	49,22	0.475857

Como exemplificação dos resultados, seguem imagens de algumas das inferências obtidas com êxito:

Figura 14 – Exemplos de inferências obtidas corretamente (Fonte: O Autor)

Durante o processo da validação cruzada, em uma das iterações, uma matriz de confusão foi criada visando demonstrar os erros da rede e dar um norte sobre quais e onde se concentram os erros de inferência da *CNN*. Segue a tabela com a matriz de confusão onde a legenda C indica o valor correto e a I o valor inferido:

Tabela 2: Matriz de confusão originada de iteração realizada

C/I	A	E	F	G	M	N	O	Q	R	X
A		6.66%			20%	20%				
C								16.66%		
D							12.5%			
I										4.34%
K						9.52%				
L					25%					
M										
Q				0.62%			1.25%			
T	40%	10%							10%	
Z			7.69%							
0										
1										
2										
5										
8										

C/I	Y	Z	0	1	2	4	5	7	9
A		20%						6.66%	
C									
D									
I									
K									
L									
M	1.78%								
Q									
T		10%	10%	20%					
Z					15.38%				
0				0.76%					
1						0.9%		0.9%	0.9%
2							1.02%		
5						0.99%		0.99%	
8						1.01%			

Na matriz de confusão, podemos observar claramente uma divisão acentuada entre erros de letra-letra e erros de número-número, o que mostra que a rede foi, na iteração pesquisada, bem com relação a entender o padrão de 3 letras seguidas de 4 números presentes nas placas de licenciamento automotivo brasileiras.

Durante o processo de construção da matriz um caso de erro interessante que se mostrou bem presente na lista de erros foram os erros de caracteres e números inferidos repetidamente de forma incorreta, dois exemplos ilustram esse caso específico:

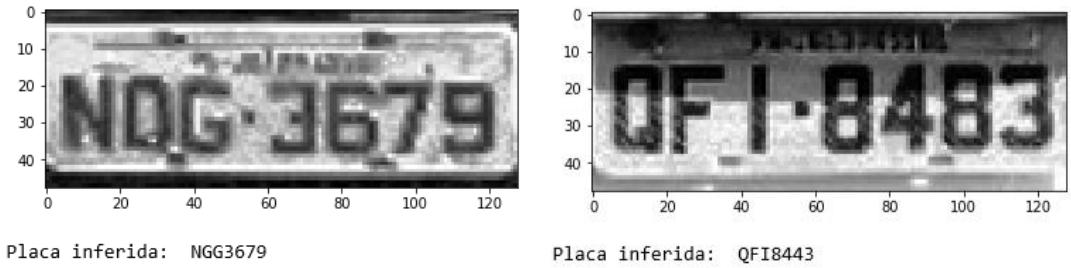


Figura 15 – Erros de inferência repetida (Fonte: O Autor)

O restante dos erros se mostrou bastante ligado a quantidade de caracteres presentes no teste, ou seja, a baixa ocorrência de um determinado caractere. Seguem exemplos de outros erros apresentados pela rede:

Figura 16 – Exemplos de erros de inferência (Fonte: O Autor)

5 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho atingiu o objetivo geral e os específicus, a medida que criou uma base de dados com placas de licenciamento veiculares brasileiras, desenvolveu uma CNN para processar as placas e obteve resultados que demonstram a viabilidade do tema.

Não era objetivo do trabalho desenvolver um sistema que pudesse ser utilizado comercialmente, portanto algumas tarefas poderão ser feitas em trabalhos futuros como:

- Testar o grau de acurácia da inferência feita a partir de um banco de testes gerado por computador;
- Realizar críticas sobre os dados como por exemplo o fato de existir um padrão AAA9999 nos dados que não foi levado em consideração para as inferências;
- Construir um banco de dados via computador para reconhecimento das novas placas do Mercosul;
- Alterar a arquitetura da rede de forma radical e testar o nível de influência sobre as inferências;
- Testar outras arquiteturas já existentes de *CNN*.

REFERÊNCIAS

ARAÚJO, Flávio H. D.; CARNEIRO, Allan C.; SILVA, Romuere R. V.; MEDEIROS, Fátima N. S.; USHIZIMA, Daniela M. **Redes Neurais Convolucionais com Tensorflow: Teoria e Prática**. III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos, v. 1, n. 1, p. 382-406, jun, 2017. Disponível em: http://www.eripi.com.br/2017/images/anais/minicursos/7.pdf. Acesso em: 15 out. 2018.

BARCA, Maria Carolina Stockler; SILVEIRA, Tiago Redondo de Siqueira; MAGINI, Marcio. TREINAMENTO DE REDES NEURAIS ARTIFICIAIS: O ALGORITMO BACKPROPAGATION. In: IX Encontro Latino Americano de Iniciação Científica e V Encontro Latino Americano de Pós-graduação, Universidade do Vale do Paraíba. Vale do Paraíba: Univap; 2005. p. 46-49. Disponível em: http://www.inicepg.univap.br/cd/INIC_2005/inic/IC1%20anais/IC1-17.pdf. Acesso em: 12 out. 2018.

DPL, Deep Learning Book. **Capítulo 23 – Como Funciona o Dropout?**. Disponível em: http://deeplearningbook.com.br/capitulo-23-como-funciona-o-dropout/. Acesso em: 14 out. 2018.

FACURE, Matheus. **Funções de Ativação - Entendendo a importância da ativação correta nas redes neurais**, 2017. Disponível em: https://matheusfacure.github.io/2017/07/12/activ-func/. Acesso em: 14 out. 2018.

FERREIRA, Alessandro dos Santos. **Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja**. Dissertação de mestrado, Universidade Federal de Mato Grosso do Sul, 2017. Disponível em: http://www.gpec.ucdb.br/pistori/orientacoes/dissertacoes/alessandro2017.pdf. Acesso em: 15 out. 2018.

JIN, Lisheng; XIAN, Huacai; BIE, Jing; SUN, Yuqin; HOU, Haijing; NIU, Qingning. **License plate recognition algorithm for passenger cars in chinese residential areas**. Sensors (Basel), p. 8355–8370, 2012. Disponível em: https://www.ncbi.nlm.nih.gov/pubmed/22969404. Acesso em: 25 out. 2018.

KARPATHY, A. **CS231n Convolutional Neural Networks for Visual Recognition**. Disponível em: http://cs231n.github.io/convolutional-networks/. Acesso em: 15 out. 2018.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. **Gradient-based learning applied to document recognition**. Proc. of the IEEE, nov.,. 1998. Disponível em: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf. Acesso em: 14 out. 2018.

MCCAFFREY, James D. Why You Should Use Cross-Entropy Error Instead Of Classification Error Or Mean Squared Error For Neural Network Classifier Training. 2013. Disponível em: https://jamesmccaffrey.wordpress.com/2013/11/05/why-you-should-use-cross-entropy-error-

instead-of-classification-error-or-mean-squared-error-for-neural-network-classifier-training/. Acesso em: 12 out. 2018.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, Vol. 5, p. 115-133, 1943. Disponível em: https://pdfs.semanticscholar.org/5272/8a99829792c3272043842455f3a110e841b1.pdf. Acesso em: 11 out. 2018.

NASCIMENTO, E. O.; OLIVEIRA, L. N. Sensitivity Analysis of Cutting Force on Milling Process using Factorial Experimental Planning and Neural Networks. **IEEE Latin America Transactions**, vol. 14, n. 12, p. 4811-4820, 2016. Disponível em: https://www.researchgate.net/publication/312027170_Sensitivity_Analysis_of_Cutting_Force_on_Milling_Process_using_Factorial_Experimental_Planning_and_Artificial_Neural_Networks. Acesso em: 12 out. 2018.

PAPAGELIS, Anthony J.; KIM, Dong Soo. **Backpropagation**. Disponível em: https://www.cse.unsw.edu.au/~cs9417ml/MLP2/BackPropagation.html. Acesso em: 13 out. 2018.

QUEIROZ, Corina Jará de. **Análise de Transformações Geométricas para o Georreferenciamento de Imagens do Satélite CBERS-I**. Dissertação de Mestrado. UFRGS - CEPSRM, 2003. Disponível em: http://hdl.handle.net/10183/6349. Acesso em: 23 out. 2018.

ROSENBLATT, F. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. **Psychological Review**, Vol. 65, Nº 6, p. 386-408, 1958. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.335.3398. Acesso em: 11 out. 2018.