RECONHECIMENTO ÓTICO DE CARACTERES USANDO REDES NEURAIS CONVOLUCIONAIS

Aluno: Luis Fernando Uzai
Orientador: Prof. Associado Aparecido Nilceu Marana

Cronograma

	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro
Estudo sobre Aprendizado de Máquina, Aprendizado Profundo e Redes Neurais Convolucionais								
Revisão Bibliográfica								
Estudo sobre reconhecimento óptico de caracteres e visão computacional								
Implementação do reconhecedor de caracteres manuscritos baseado em rede neural convolucional								
Treinamento e otimização do reconhecedor de caracteres manuscritos								
Análise de desempenho e testes								
Redação do TCC								
Apresentação do TCC								

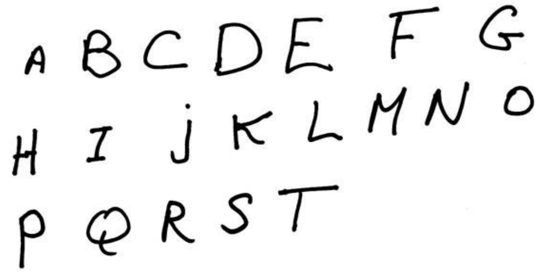
Objetivo Geral

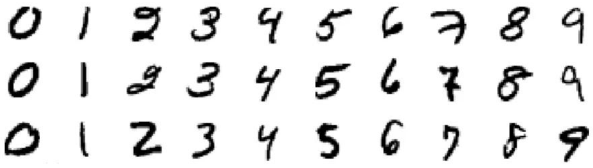
Este trabalho visou aplicar técnicas da Inteligência Artificial, em específico, Redes Neurais Artificiais Convolucionais, para o problema de classificação de caracteres manuscritos.

Objetivos Específicos

- Estudar Visão Computacional e Reconhecimento Óptico de Caracteres;
- Estudar Aprendizado Profundo e Redes Neurais Convolucionais;
- Desenvolver a arquitetura da Rede Neural Convolucional para a classificação de caracteres manuscritos;
- Treinar a Rede Neural Convolucional;
- Desenvolver um aplicativo para experimentos;
- Avaliar Resultados.

Caracteres Manuscritos

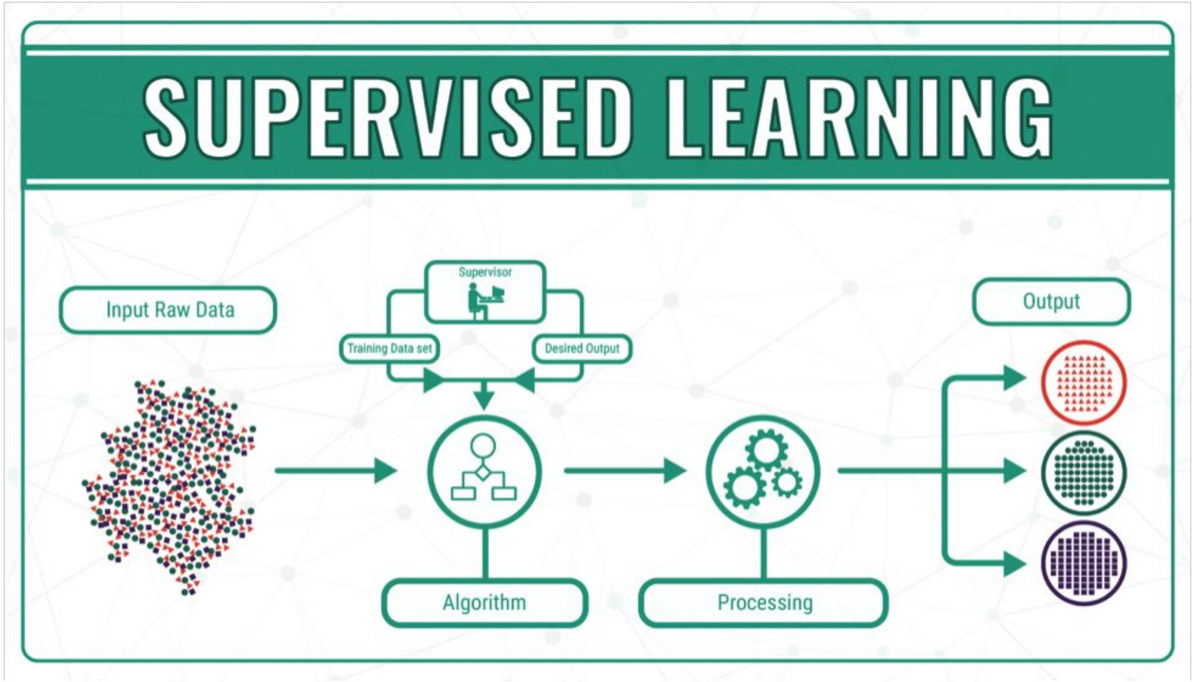




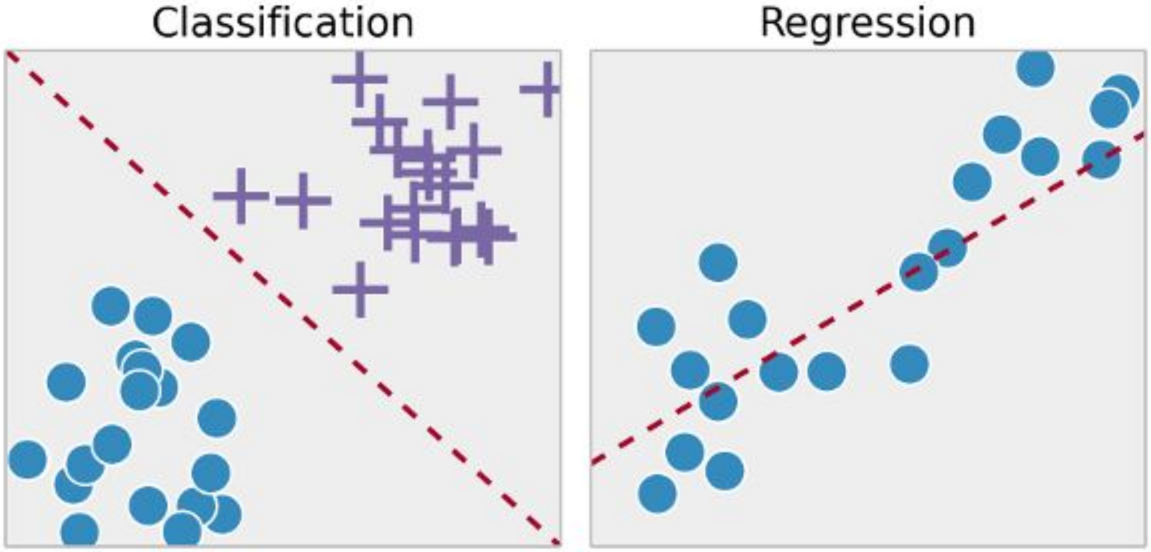
Aprendizado de Máquina

Definição: **Aprendizado de Máquina** é sobre extrair conhecimento de um grande conjunto de dados (**MUELLER e GUIDO, 2016**).

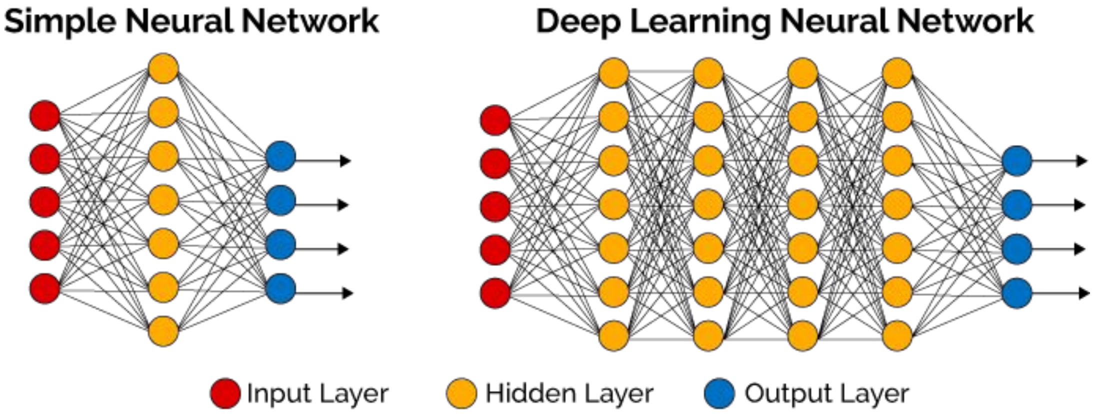
Aprendizado de Máquina do tipo Supervisionado



Aprendizado de Máquina do tipo Supervisionado



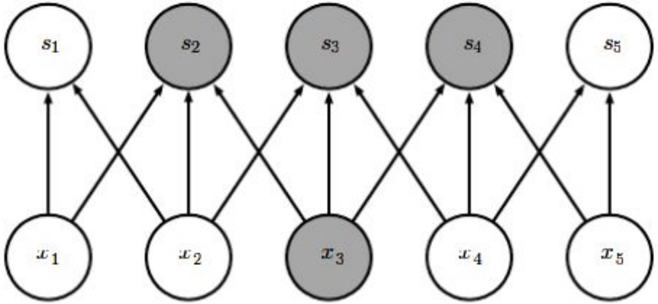
Aprendizado Profundo

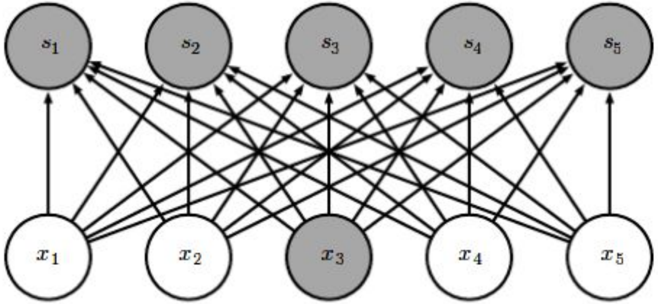


Redes Neurais Convolucionais

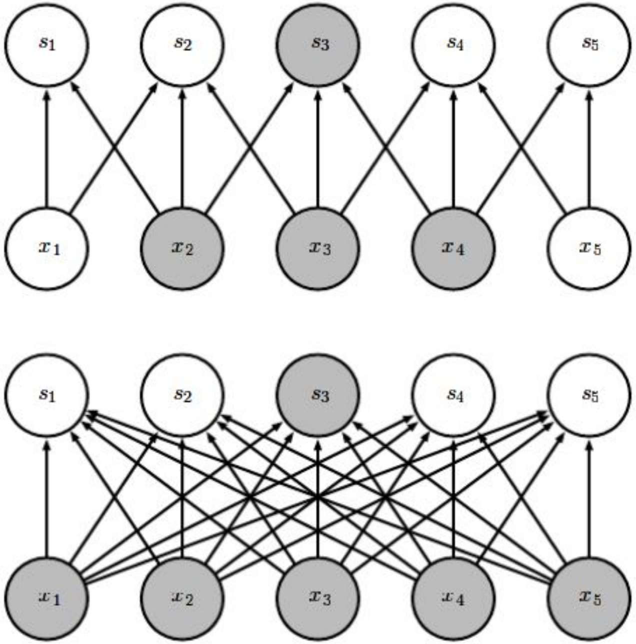
Redes Neurais Convolucionais ou CNN (Convolutional Neural Networks) é um tipo de rede neural especializada em processar dados que tem como característica topologia em grades, ou seja, dados representados em 1-D, 2-D, ..., n-D. O nome "Redes Neurais Convolucionais" vem por conta da **operação matemática chamada convolução que é aplicada nas camadas da rede.** Convolução é um tipo de operação linear. **CNN é uma rede que usa convolução no lugar da matriz comum de multiplicação, em pelo menos uma de suas camadas** (GOODFELLOW; BENGIO; COURVILLE, 2016).

Conexões Esparsas

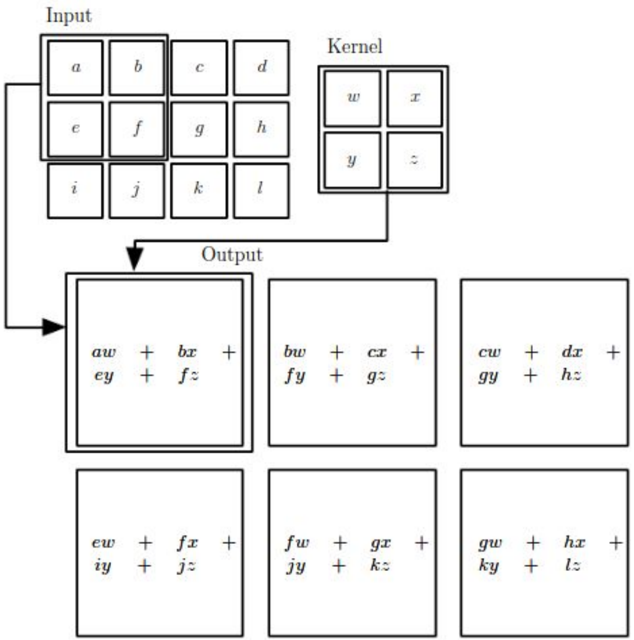




Conexões Esparsas

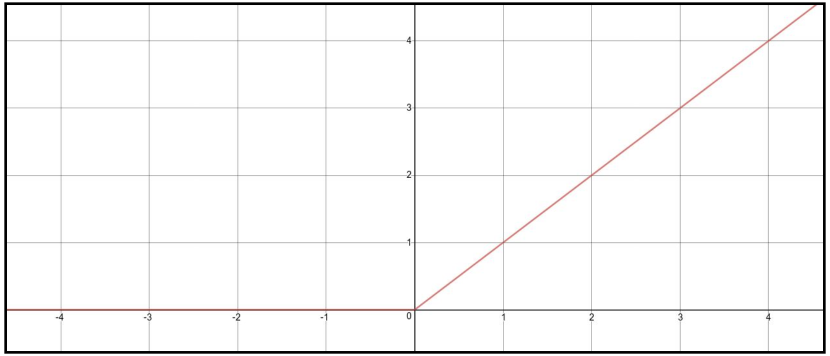


Camada de Convolução

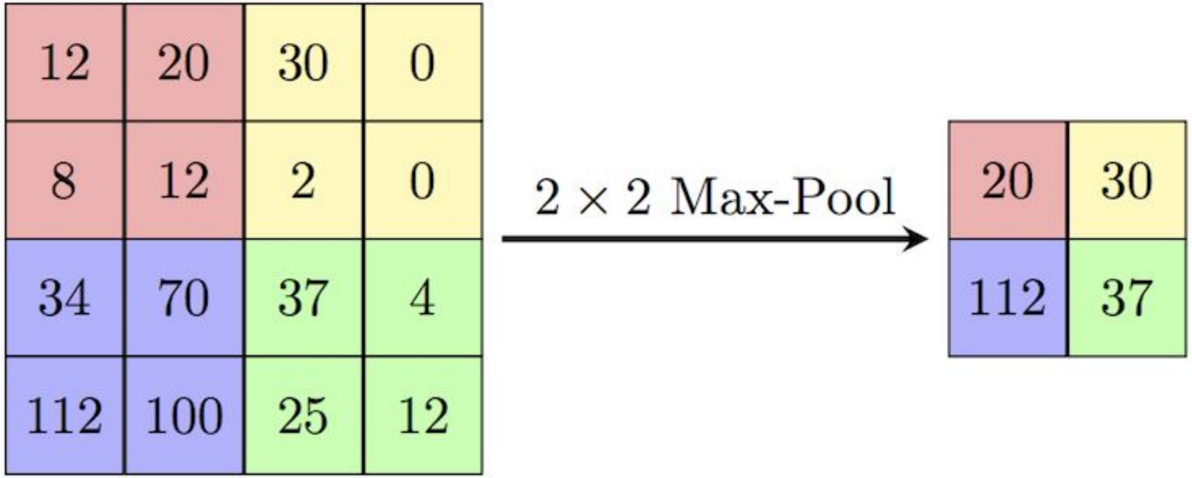


ReLU - Função de Ativação

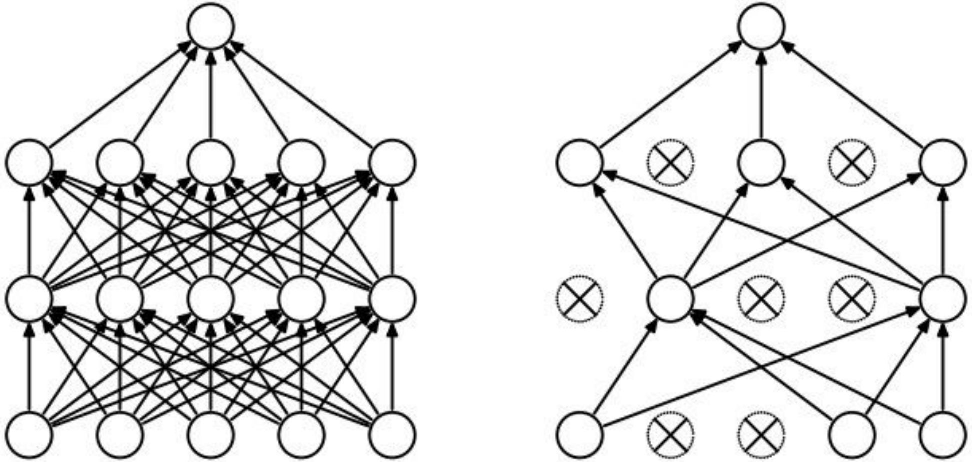
$$f = \max(0, x)$$



Max Pooling



Dropout Layer



(a) Standard Neural Net

(b) After applying dropout.

Nadam

Algorithm 3 Nesterov's accelerated gradient

$$\mathbf{g}_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1} - \eta\mu\mathbf{m}_{t-1})$$
$$\mathbf{m}_t \leftarrow \mu\mathbf{m}_{t-1} + \mathbf{g}_t$$
$$\theta_t \leftarrow \theta_{t-1} - \eta\mathbf{m}_t$$

Dataset: Extended Modified NIST

EMNIST: an extension of MNIST to handwritten letters

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik
The MARCS Institute for Brain, Behaviour and Development
Western Sydney University
Penrith, Australia 2751
Email: g.cohen@westernsydney.edu.au

arXiv:1702.05373v2 [cs.CV] 1 Mar 2017

***Abstract*—The MNIST dataset has become a standard benchmark for learning, classification and computer vision systems. Contributing to its widespread adoption are the understandable and intuitive nature of the task, its relatively small size and storage requirements and the accessibility and ease-of-use of the database itself. The MNIST database was derived from a larger dataset known as the NIST Special Database 19 which contains digits, uppercase and lowercase handwritten letters. This paper introduces a variant of the full NIST dataset, which we have called Extended MNIST (EMNIST), which follows the same conversion paradigm used to create the MNIST dataset. The result is a set of datasets that constitute a more challenging classification tasks involving letters and digits, and that shares the same image structure and parameters as the original MNIST task, allowing for direct compatibility with all existing classifiers and systems. Benchmark results are presented along with a validation of the conversion process through the comparison of the classification results on converted NIST digits and the MNIST digits.**

I. Introduction

The importance of good benchmarks and standardized problems cannot be understated, especially in competitive and fast-paced fields such as machine learning and computer vision. Such tasks provide a quick, quantitative and fair means of analyzing and comparing different learning approaches and techniques. This allows researchers to quickly gain insight into the performance and peculiarities of methods and algorithms, especially when the task is an intuitive and conceptually simple one.

As single dataset may only cover a specific task, the existence of a varied suite of benchmark tasks is important in allowing a more holistic approach to assessing and characterizing the performance of an algorithm or system. In the machine learning community, there are several standardized datasets that are widely used and have become highly competitive. These include the MNIST dataset [1], the CIFAR-10 and CIFAR-100 [2] datasets, the STL-10 dataset [3], and Street View House Numbers (SVHN) dataset [4].

Comprising a 10-class handwritten digit classification task and first introduced in 1998, the MNIST dataset remains the most widely known and used dataset in the computer vision and neural networks community. However, a good dataset needs to represent a sufficiently challenging problem to make it both useful and to ensure its longevity [5]. This is perhaps where MNIST has suffered in the face of the increasingly high accuracies achieved using deep learning and convolutional neural networks. Multiple research groups have published accuracies above 99.7% [6]–[10], a classification accuracy at which the dataset labeling can be called into question. Thus, it has become more of a means to test and validate a classification system than a meaningful or challenging benchmark.

The accessibility of the MNIST dataset has almost certainly contributed to its widespread use. The entire dataset is relatively small (by comparison to more recent benchmarking datasets), free to access and use, and is encoded and stored in an entirely straightforward manner. The encoding does not make use of complex storage structures, compression, or proprietary data formats. For this reason, it is remarkably easy to access and include the dataset from any platform or through any programming language.

The MNIST database is a subset of a much larger dataset known as the NIST Special Database 19 [11]. This dataset contains both handwritten numerals and letters and represents a much larger and more extensive classification task, along with the possibility of adding more complex tasks such as writer identification, transcription tasks and case detection.

The NIST dataset, by contrast to MNIST, has remained difficult to access and use. Driven by the higher cost and availability of storage when it was collected, the NIST dataset was originally stored in a remarkably efficient and compact manner. Although source code to access the data is provided, it remains challenging to use on modern computing platforms. For this reason, the NIST recently released a second edition of the NIST dataset [12]. The second edition of the dataset is easier to access, but the structure of the dataset, and the images contained within, differ from that of MNIST and are not directly compatible.

The NIST dataset has been used occasionally in neural network systems. Many classifiers make use of only the digit classes [13], [14], whilst others tackle the letter classes as well [15]–[18]. Each paper tackles the task of formulating the classification tasks in a slightly different manner, varying such fundamental aspects as the number of classes to include, the training and testing splits, and the preprocessing of the images.

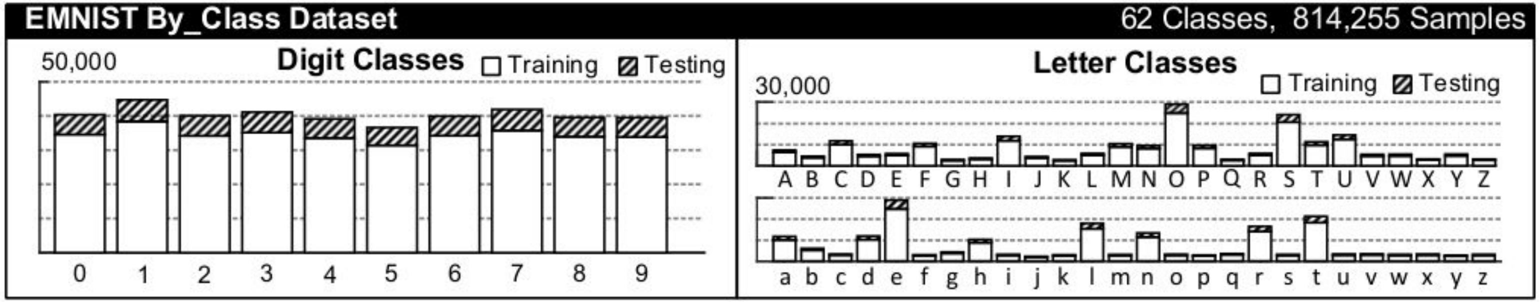
In order to bolster the use of this dataset, there is a clear need to create a suite of well-defined datasets that thoroughly specify the nature of the classification task and the structure of the dataset, thereby allowing for easy and direct comparisons

EMNIST conjuntos de dados

TABLE II
STRUCTURE AND ORGANIZATION OF THE EMNIST DATASETS.

Name	Classes	No. Training	No. Testing	Validation	Total
By_Class	62	697,932	116,323	No	814,255
By_Merge	47	697,932	116,323	No	814,255
Balanced	47	112,800	18,800	Yes	131,600
Digits	10	240,000	40,000	Yes	280,000
Letters	37	88,800	14,800	Yes	103,600
MNIST	10	60,000	10,000	Yes	70,000

Características do EMNIST



Keras

Keras é uma biblioteca de redes neurais artificiais escrita em Python e de **código aberto**. É capaz de rodar no seu backend bibliotecas como: **TensorFlow, Microsoft Cognitive Toolkit ou Theano.** O seu foco é **permitir um desenvolvimento fácil de redes neurais profundas**, com foco em ser amigável ao usuário, modular e extensiva. Um dos focos do Keras é ser capaz de ir da ideia ao resultado de forma rápida.

(Fonte: https://keras.io/).

Desenvolvimento

Problema: EMNIST contém as imagens rotacionadas 180º.

```python
# Função para a rotação da imagem
def rotate(image):
    image = image.reshape([28, 28])
    image = np.fliplr(image)
    image = np.rot90(image)
    return image.reshape([28 * 28])
```

Desenvolvimento

Problema: EMNIST contém as imagens rotacionadas 180º.

```
    # Rotacionar Imagens
    train_images = np.apply_along_axis(rotate, 1, train_images)/255
```

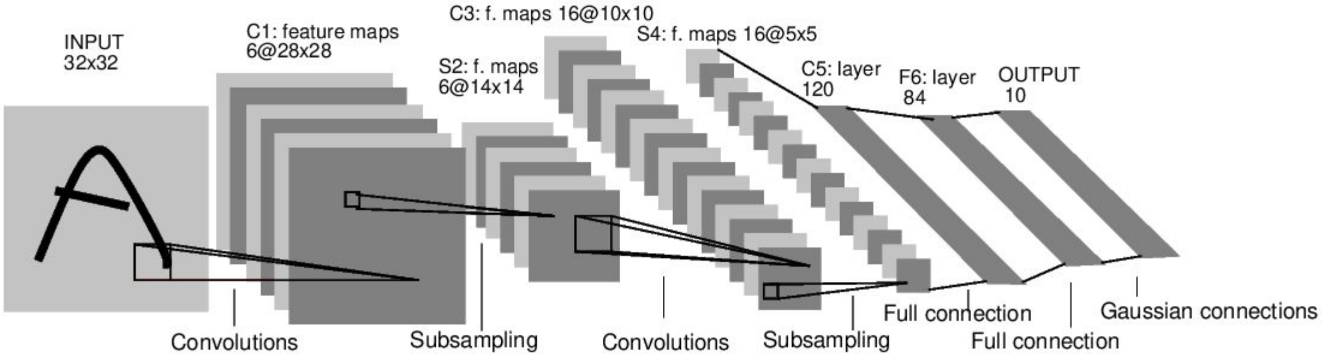
Desenvolvimento

Problema: grande volume no conjunto de dados

```
# Carregando os data sets
train = pd.read_csv(filepath_or_buffer=HOME+r'/dev/datasets/emnist-byclass-train.csv',
                    header=None, chunksize=40000)
test = pd.read_csv(filepath_or_buffer=HOME+r'/dev/datasets/emnist-byclass-test.csv',
                   header=None, chunksize=40000)
```

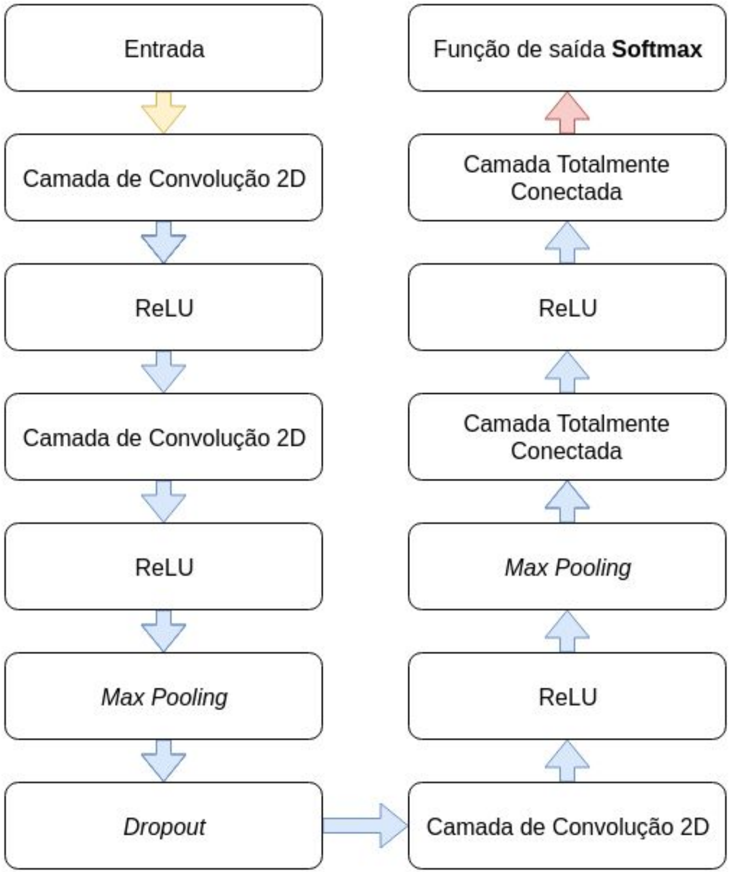
Desenvolvimento

Baseando-se na arquitetura proposta por LeNet, et al. 1998



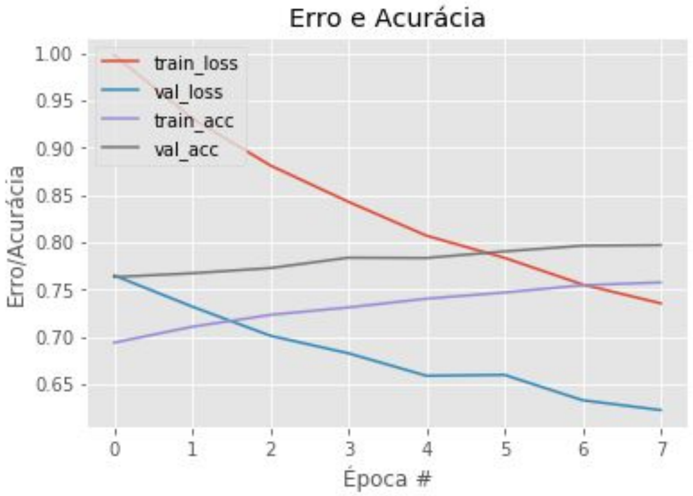
Desenvolvimento

Arquitetura proposta



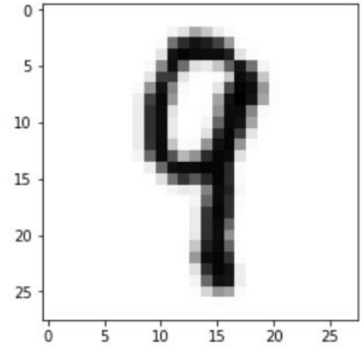
Desenvolvimento

Treinamento



Desenvolvimento

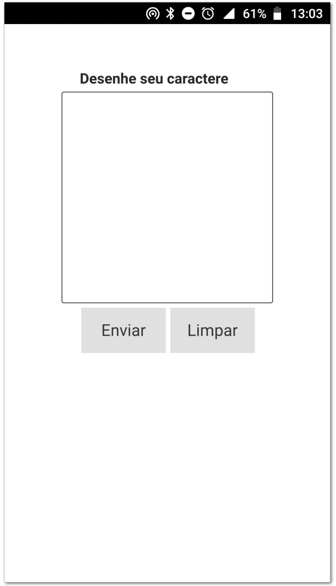
Teste

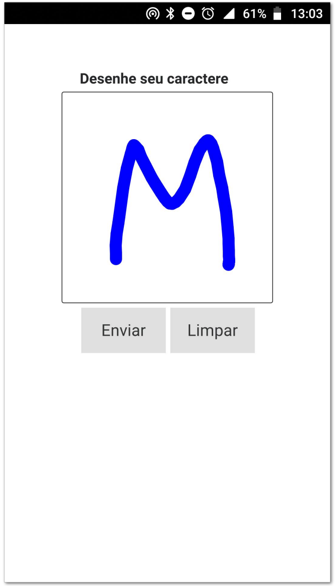


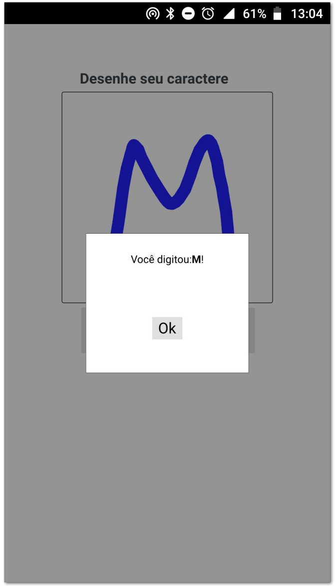

```
O modelo prevê: q
40000/40000 [==============================] - 13s 337us/step
Acurácia no conjunto de teste:: 77.08%
```

Desenvolvimento

Aplicativo







Resultados

Método	Fonte	Acurácia
Classificador Linear	Artigo EMINIST (COHEN et al., 2017)	69,71%
Redes Neurais Convolucionais	Este trabalho	76,67%

Conclusão

- **Desafio**: conjunto de dados com **62 classes**;
- As **redes neurais convolucionais** se mostraram um método **eficiente** para a classificação de caracteres manuscritos;
- Todo o estudo, desenvolvimento e modelagem contidos neste projeto são semelhantes ao que seria um estudo de aprendizado profundo para **aplicações práticas reais**;
- Conceitos **matemáticos e estatísticos** estão por trás dos métodos utilizados, e **contribuição para a formação do aluno**;
- Utilização de **tecnologia atuais**;
- Utilização de **metodologia atuais**;
- **Aplicação dos conhecimentos** obtidos ao longo da graduação.

Tecnologias Usadas

NumPy

Qt

Flask
web development,
one drop at a time

Keras

jupyter

Pandas

Referências

DEEP LEARNING BOOK . Deep Learning Book . 2018. Disponível em: <http://deeplearningbook.com.br/capitulo-1-deep-learning-a-tempestade-perfeita/>.

JOSHI, PRATEEK. Artificial Intelligence with Python, v.1, 2017.

BUDUMA, NIKHIL. Fundamentals of Deep Learning, v.1, 2017.

GOODFELLOW, BENGIO, COURVILLE. Deep Learning Book. 2018. Disponível em: <http://www.deeplearningbook.org/>.

Referências

MUELLER, GUIDO. Introduction to Machine Learning with Python, v.1, 2017.

CORNELISSE. An intuitive guide to Convolutional Neural Networks. 2018. Disponível em: <https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>.

Keras Team. Keras Documentation. 2018. Disponível em: <https://keras.io/>.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. EMNIST: an extension of MNIST to handwritten letters. 2017. Disponível em: <https://arxiv.org/pdf/1702.05373.pdf>.