

# Fundamentos de Redes Neurais Profundas:

Abordagem Baseada em Redes Convolucionais.

Rafael Gonçalves & Romis Attux.

Faculdade de Engenharia Elétrica e de Computação - Unicamp  
r186062@dac.unicamp.br, attux@dca.fee.unicamp.br



## Introdução

Redes neurais artificiais são sistemas de computação não lineares e adaptativos originalmente inspirados nas redes neurais biológicas presentes no sistema nervoso dos animais. Especialmente com o advento de redes neurais profundas e o conceito de aprendizado profundo, este se tornou um importante paradigma dentro do campo de aprendizado de máquina e é amplamente utilizado para resolver uma variedade de problemas atuais.

Neste contexto, esta pesquisa buscou estudar teoricamente redes neurais profundas baseado em um livro recente e representativo [1] e posteriormente aplicar um modelo específico de rede neural – a saber uma rede convolucional – ao problema conhecido de reconhecimento de dígitos escritos à mão utilizando a base de dados MNIST [2].

## Discussões e Resultados

### Perceptron de Múltiplas Camadas - MLP

As redes MLP podem ser vistas como redes de neurônios perceptron interligados em forma de camadas: uma camada de entrada que recebe cada entrada do vetor  $\mathbf{x}$ , uma ou mais camadas intermediárias e uma camada de saída. Exemplo de uma rede MLP pode ser visto na figura ??.

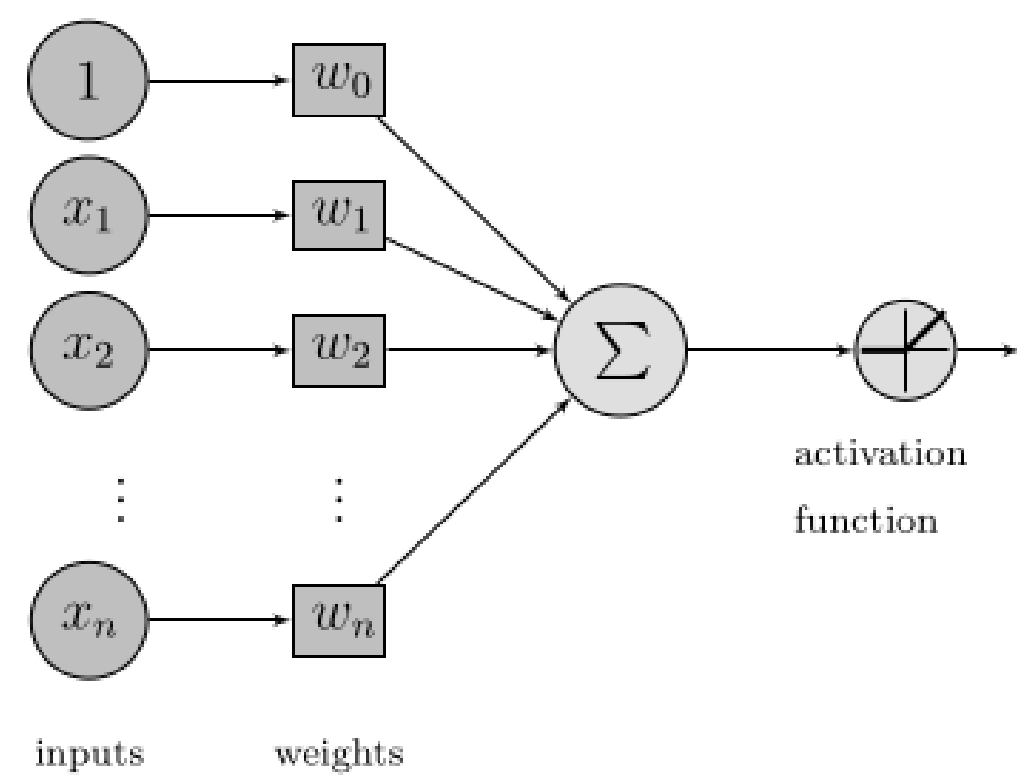


Figura 1: Neurônio de tipo perceptron.

Desta forma, matematicamente a saída de uma dessas redes – considerando  $\mathbf{w}^n$ ,  $w_0^n$  e  $f^n$  como respectivamente matriz de pesos, viés (*bias*) e função de ativação da camada  $n$  – é:

$$y = f^N(\mathbf{w}^N \cdot \dots f^2(\mathbf{w}^2 \cdot f^1(\mathbf{w}^1 \cdot \mathbf{x} + w_0^1) + w_0^2) + w_0^N) \quad (1)$$

Ou ainda, se definirmos uma matriz de entrada que admita  $M$  exemplos em uma mesma estrutura – podemos ainda definir um vetor  $\mathbf{W}$  que inclua  $w_0$  no vetor  $\mathbf{w}$  e definir uma matriz  $\Phi$  cuja linha  $\phi_i = [1 \quad \mathbf{x}_i]$  com  $i$  indicando cada exemplo:

$$\mathbf{y} = \mathbf{F}^n(\dots \mathbf{F}^2(\mathbf{F}^1(\Phi \cdot \mathbf{W}^1)\mathbf{W}^2)\mathbf{W}^N) \quad (2)$$

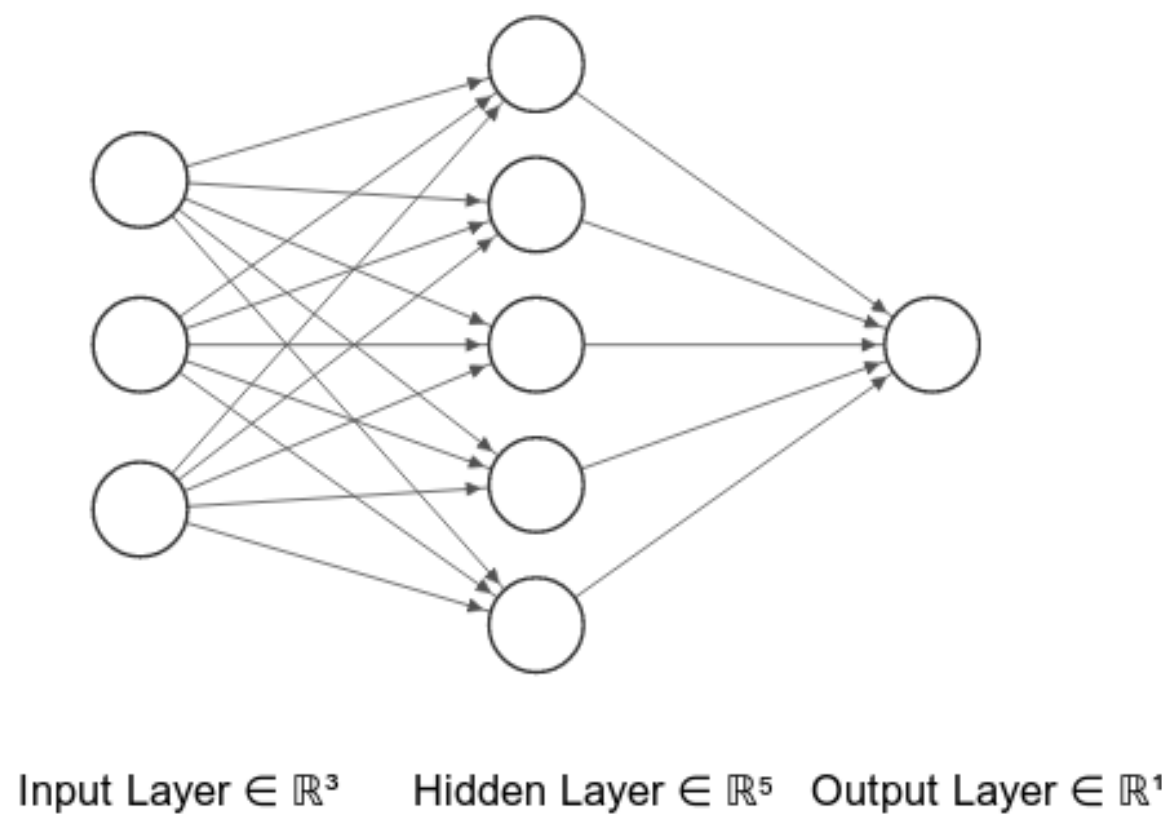


Figura 2: Exemplo de rede MLP com 3 atributos de entrada, uma camada intermediária com 5 neurônios e uma camada de saída com um único neurônio, gerado com [4].

### Rede Neural Convolucional - CNN

Redes CNN são arquiteturas de ANNs com ao menos uma camada convolucional. Camadas convolucionais usam kernels de pesos e a operação matemática de convolução para gerar a informação de saída do neurônio, diferentemente das MLPs que usam o produto interno da entrada com um vetor de pesos.

Exemplo da operação de convolução entre uma imagem  $I$  2D e um kernel  $K$ :

$$(I * K)(i, j) = (K * I)(i, j) = \sum_m \sum_n K(m, n) \cdot I(i - m, j - n) \quad (3)$$

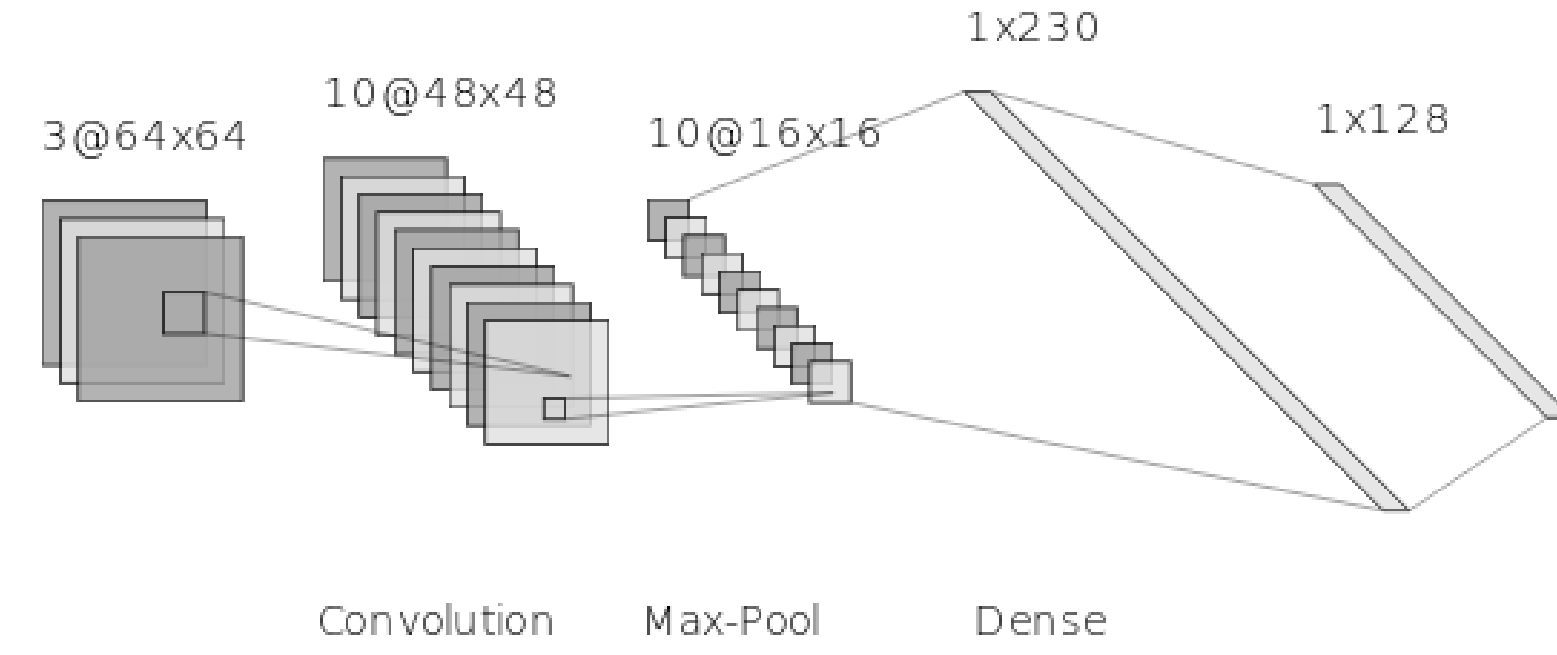


Figura 3: Exemplo de rede CNN com 1 camada convolucional - convolução e max-pooling - e 2 camadas FC, gerado com [4].

Redes MLP e CNN foram implementadas com diferentes configurações para dropout (no caso de ambas as arquiteturas) e quantidade de neurônios nas camadas internas (na arquitetura MLP).

### Progressão do erro ao longo das épocas de treinamento.

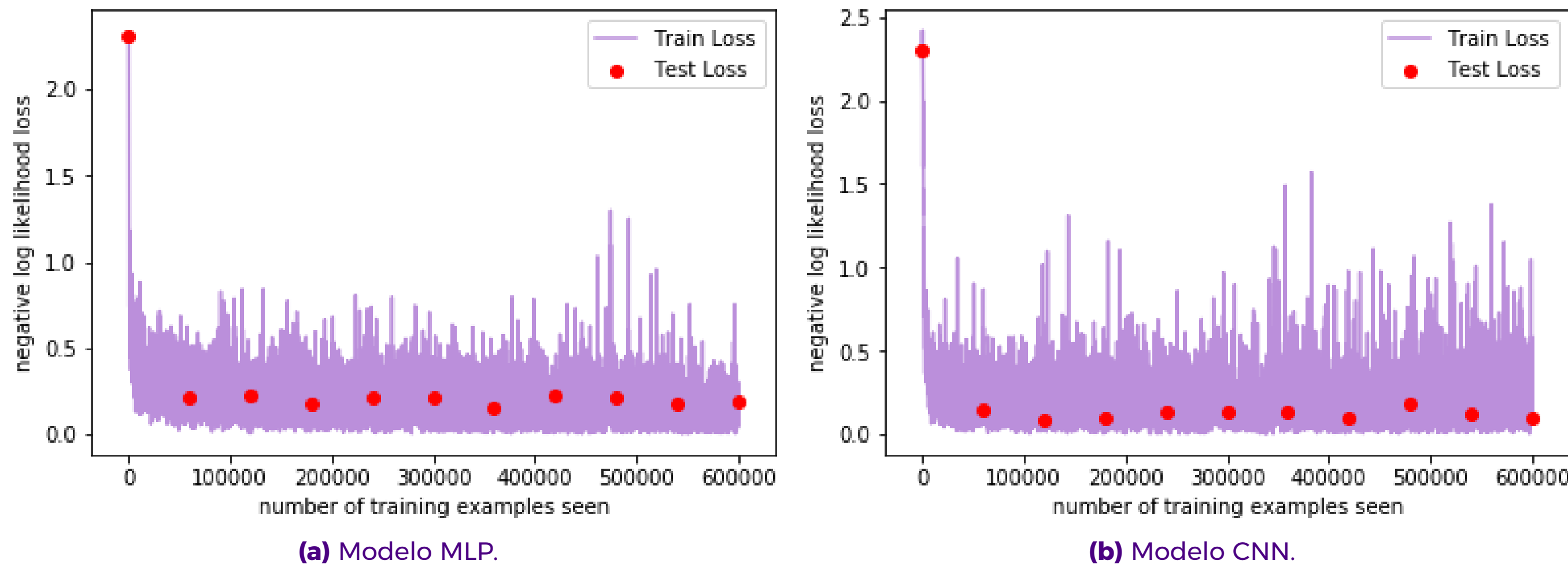


Figura 4: Curva de aprendizado dos melhores modelos de cada tipo de arquitetura.

O resultado final dos melhores modelos estão na tabela [?].

Tabela 1: Modelos finais avaliados no conjunto de teste.

Modelo	Melhor Época	Melhor Acurácia
MLP	10	0.9663
CNN	7	0.9771

## Conclusões

O estudo mostrou que tanto a arquitetura MLP como a CNN são viáveis para o problema de classificação de dígitos escritos à mão [2]. O modelo final de rede convolucional apresentou um resultado ligeiramente melhor – acurácia de 97.8% em comparação à 96.7% referente à MLP.

O uso de dropout teve pouca influência em ambas as arquiteturas, sendo que nas redes MLP o mesmo contribuiu negativamente para o aumento da acurácia e nas redes CNN um dropout de 30% foi o que proveu o maior índice. Nas redes MLP a variação do número de neurônios nas camadas intermediárias teve pouca influência o problema pode ser resolvido por modelos mais simples.

## Agradecimentos

O estudante gostaria de expressar seu agradecimento ao programa PIBIC/CNPq/Unicamp pelo auxílio financeiro e em especial à Romis Attux por todo o incentivo e apoio durante o desenvolvimento da pesquisa.

## Referências

[1] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. MIT Press, 2016.

[2] Y. LeCun. *The MNIST Database of Handwritten Digits*. <http://yann.lecun.com/exdb/mnist>. (acessado em 07/07/2019).

[3] R. Gonçalves. *mnist\_nn*. [https://github.com/RafaelGoncalves8/mnist\\_nn](https://github.com/RafaelGoncalves8/mnist_nn) (acessado em 20/07/2019).

[4] LeNail. *NN-SVG: Publication-Ready Neural Network Architecture Schematics*. <http://alexlenail.me/NN-SVG/>. Journal of Open Source Software, 2019. (acessado em 20/07/2019).