

Laboratório 8 - Redes Neurais Convolucionais

Marcelo Buga Martins da Silva

CT-213 - Professor Marcos Ricardo Omena de Albuquerque Máximo

26/05/2021



1 Introdução

O propósito desse laboratório foi a implementação de uma rede neural convolucional para a identificação de imagens de números de 0 a 9 com a *framework* de Deep Learning Keras, que utiliza do *TensorFlow* para a implementação dos algoritmos. Esse laboratório foi uma tentativa de implementar, treinar e testar a rede neural LeNet-5 usando o *dataset* MNIST, um conjunto grande de imagens anotadas de dígitos decimais escritos à mão.

Esse relatório apresentará uma breve descrição da implementação, do treinamento e dos testes realizados para essa rede neural convolucional.

2 Implementação

A implementação dessa rede neural foi bem simples, visto que o Keras a torna muito intuitiva. Basicamente, bastou-se adicionar as camadas com os dados da Tabela 1 através da função *add*, que tem como argumento o tipo de camada e seus respectivos parâmetros.

Tabela 1: Arquitetura da LeNet-5

# Camada	Tipo	Número de Filtros	Tamanho da Saída	Tamanho do Kernel	Stride	Activation Function
Entrada	Imagem	1	32x32	-	-	-
1	Conv2D	6	28x28	5x5	1	tanh
2	AveragePooling2D	6	14x14	2x2	2	-
3	Conv2D	16	10x10	5x5	1	tanh
4	AveragePooling2D	16	5x5	2x2	2	-
5	Conv2D	120	1x1	5x5	1	tanh
6	Dense (FC)	-	84	-	-	tanh
7	Dense (FC)	-	10	-	-	softmax

3 Treinamento

Para o treinamento da rede neural, utilizou-se o *script train_lenet5*, que dividiu o *training set* em uma parte para treinamento e outra para validação. Foram utilizados *batches* de 128 imagens para o treinamento e um total de 60000 imagens divididas em 48000 para o treinamento e 12000 para a validação. Esse *script* também permitiu a utilização do *Tensorboard*, que deu informações gráficas acerca da evolução do custo e da precisão alcançadas pelo algoritmo por época. As Figuras 1 e 2 representam *prints* dos gráficos do *Tensorboard* ao fim do treinamento, de forma q a linha rosa representa o treinamento e a verde a validação.

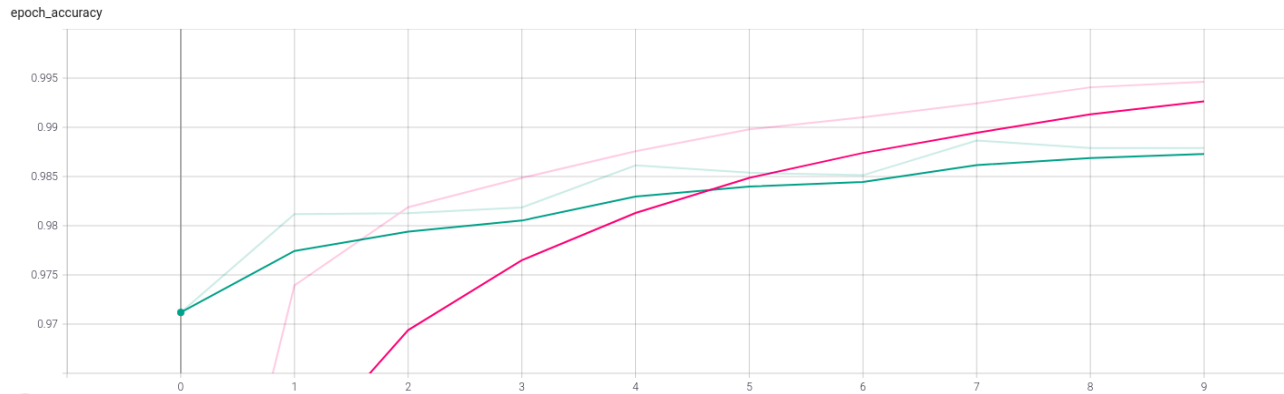


Figura 1: Precisão da classificação por época

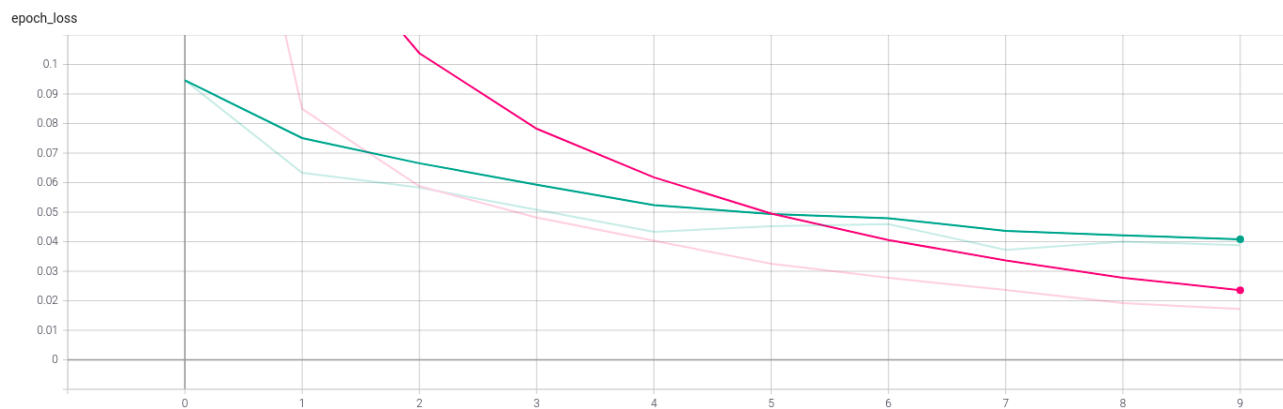


Figura 2: Custo médio por época

O treinamento foi bem sucedido, uma vez que a precisão convergiu para valores próximos de 100% e o custo médio para valores próximos a 0. Foram salvos os parâmetros de peso no arquivo *lenet5.h5* e os parâmetros da rede neural em *lenet5.json*. Prosseguiu-se então para o teste da rede neural.

4 Teste

Para o teste, foi utilizado um *dataset* diferente do utilizado para o treinamento, de forma a testar a rede para 10000 imagens. A rede neural teve um excelente desempenho, tendo sido atingido precisão de 98,95% e custo médio 0,034. Dessa forma, a grande maioria do *dataset* de teste foi classificado corretamente. A Figura 3 mostra uma dessas classificações corretas:

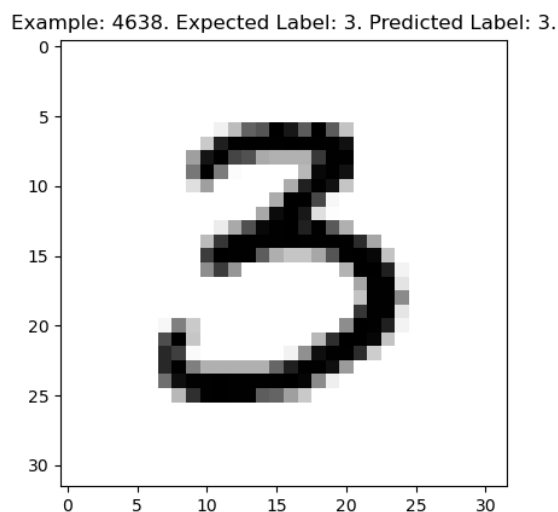


Figura 3: Classificação correta - Esperado: 3. Classificação da rede neural 3

Entretanto, ainda existiram algumas imagens que a rede neural classificou incorretamente. A Figura 4 mostra um exemplo de um desses erros:

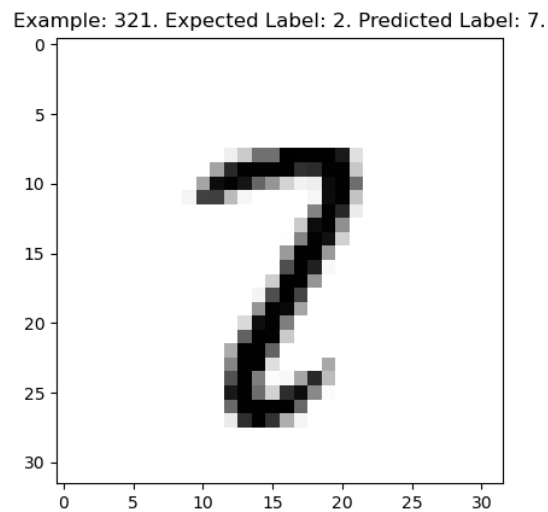


Figura 4: Classificação errada - Esperado: 2. Classificação da rede neural: 7

Para a grande maioria dos casos de teste que a rede neural falhou, a identificação do número é difícil até mesmo para um ser humano. Além disso, para alguns desses casos, é possível entender porque a rede neural “se enganou”, pois existem padrões parecidos entre dois algarismos, como o que ocorreu na Figura 4.

De modo geral, a implementação da rede neural LeNet-5 teve resultado muito bom, conseguindo precisão acima de 98,5% para as imagens analisadas e tendo errado figuras que, em sua maioria, são de difícil identificação.