

Relatório 15 - Prática: Redes Neurais Convolucionais

Lucas Scheffer Hundsdorfer

Descrição da atividade

Seção 9 - Teoria sobre as redes Convolucionais.

Aula 1 - Imagens e pixels:

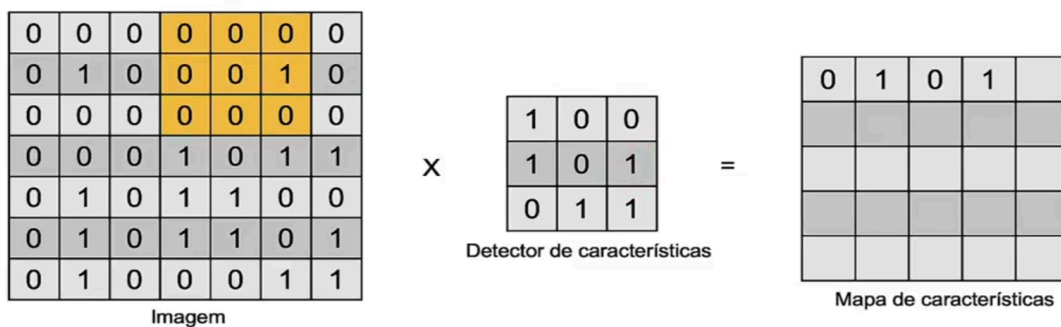
Já no começo ele vai explicando que é necessário entendermos sobre pixels para podermos avançar no conteúdo, pixel é a menor informação possível em uma imagem, dentro de cada pixel eles tem uma fonte de cor baseada no RGB (Red, Green, Blue) com uma combinação dessas 3 cores é possível chegar em todas as outras disponíveis.

Aula 2 - Introdução a redes neurais convolucionais:

Dentro das redes convolucionais não é utilizado todos os pixels, por exemplo ele utilizou os emojis, os emojis na parte lateral são todos iguais, algo que não vai mudar em nada a classificação deles, a própria rede neural escolhe os pixels necessários que representam a imagem. Dentro da rede neural há 4 passos, 1 - Operador de convolução, 2 - Pooling, 3 - Flattening, 4 - Rede neural densas.

Aula 3 - Etapa 1 - Operador de Convolução:

No passo de convolução para reduzirmos a dimensões dos pixels é fazer uma multiplicação da imagem que vai ser reduzida, pela uma outra matriz de detector de características, dessa forma:



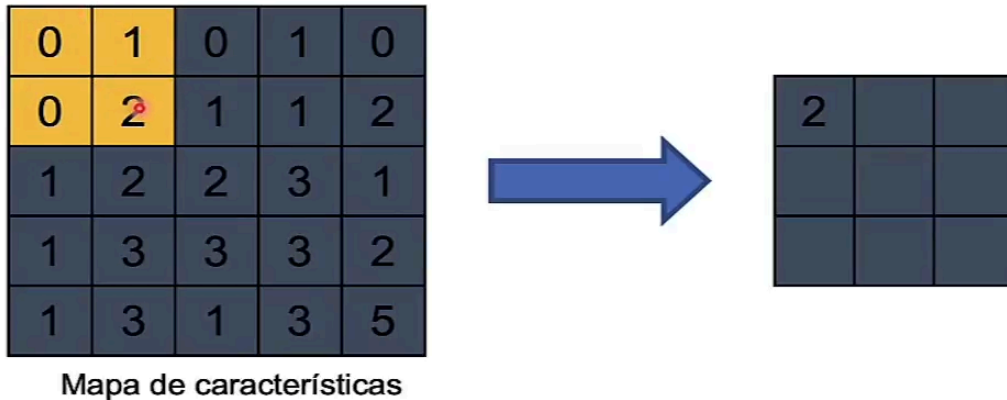
$$0 * 1 + 0 * 0 + 0 * 0 + 0 * 1 + 0 * 0 + 1 * 1 + 0 * 0 + 0 * 1 + 0 * 1 = 1$$

O vídeo explica também que o mapa de características muda de acordo com o detector ou (kernel) utilizado. Esse mapa de características representa os pixels mais importantes para a imagem. Após obter o mapa de características é aplicado a função 'relu' onde os números negativos retornam 0, e os números positivos retornam eles mesmos, com o intuito de melhorar a qualidade da imagem.

Aula 4 - Etapa 2 - Pooling:

A ideia dessa segunda etapa é que a rede consiga se adaptar aos diversos tipos de ambiente de uma imagem, ainda deve ser possível identificar que seja um cachorro em tal imagem mesmo que ele esteja na neve ou na água.

Existem vários tipos de pooling, e o mais utilizado é o max pooling, onde é feita a extração do maior valor dentro do mapa de características:



Aula 5 - Etapa 3 - Flattening:

Após o pooling é retornando uma matriz menor ainda, e transformamos essa matriz em um vetor, é necessário transformar em vetor, pois esses dados vão ser enviados para uma camada de entrada de uma rede neural densa.

Após essas aulas o curso se torna muito mais prático do que teórico, trazendo códigos do que foi apresentado anteriormente:

```
[1]: import tensorflow as tf
import keras
import matplotlib
import numpy as np

[2]: from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, Dense, Flatten, Dropout, Conv2D, MaxPooling2D, BatchNormalization
from tensorflow.keras import utils as np_utils
import matplotlib.pyplot as plt

[ ]: Carregamento da base de dados.

[3]: (X_treinamento, y_treinamento), (X_teste, y_teste) = mnist.load_data()

[4]: X_treinamento.shape, X_teste.shape

[4]: ((60000, 28, 28), (10000, 28, 28))

[5]: X_treinamento, X_treinamento.max()
```

Aqui ele mostra como importar as bibliotecas necessárias para podermos utilizar e criar uma rede neural convolucional, todas essas bibliotecas só são possíveis graças ao tensorflow.

E aqui é como fica a ‘cara’ de uma rede neural convolucional:

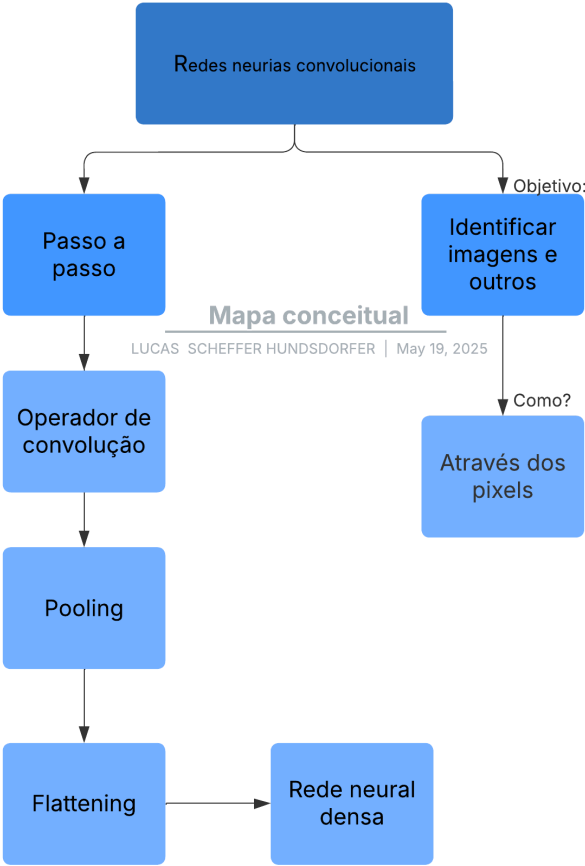
rede_neural.summary()

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 32)	320
batch_normalization (BatchNormalization)	(None, 26, 26, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_2 (Conv2D)	(None, 11, 11, 32)	9,248
batch_normalization_1 (BatchNormalization)	(None, 11, 11, 32)	128
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_2 (Dense)	(None, 128)	102,528
dropout (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 128)	16,512
dropout_1 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 10)	1,290

Total params: 130,154 (508.41 KB)
Trainable params: 130,026 (507.91 KB)
Non-trainable params: 128 (512.00 B)

Insight visual original feito em forma de mapa mental utilizando o LucidChart:



Conclusões

A conclusão que eu chego após o curso e os códigos é que uma rede neural densa serve muito bem para o intuito dela, e funciona de forma bem parecida com uma rede neural comum, porém com alguns parâmetros a mais dentro dela uma normalização diferente dos dados, mas ela útil em outras áreas se diferenciando da rede neural, a rede neural convolucional é melhor identificando objetos em imagens e até em áudios ela pode ser muito boa.