

Modelagem de Redes Neurais Profundas com Keras

Caio Jordão Carvalho

Instituto Federal da Bahia

caiojcarvalho@gmail.com

6 de novembro de 2018

Caio Jordão Carvalho

- 6º semestre em ADS (**Instituto Federal da Bahia**);
- Back-end Developer no **Escavador**;
- Contribuidor na comunidade **KDE**;
- Visão Computacional, Inteligência Artificial e Machine Learning.
- Github: **@cjlcarvalho**

Agenda

1 Redes Neurais Artificiais

- Perceptron

2 Convolução e Pooling

3 Deep Learning

- Convolutional Neural Networks
- Generative Adversarial Networks
- Autoencoders

4 Keras

- Neurônio Artificial.
- Abstração do sistema nervoso biológico.
- Primeiro modelo de Neurônio Artificial (McCulloch e Pitts, 1943).
- Perceptron (Rosenblatt, 1957).

Componentes de um Neurônio

- Vetor de Entradas (Inputs).
- Vetor de Pesos (Weights).
- Bias term.
- Função de Ativação.

Mudança dos pesos == Aprendizado.

Organização de um Perceptron

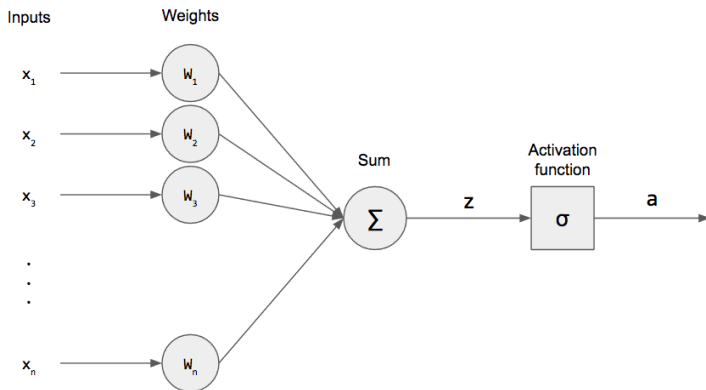


Figura: Entradas, pesos, produto escalar e função de ativação.

Dada uma entrada, define um estado para um neurônio.

Tipos

- ① Step Function.
- ② Linear.
- ③ Sigmoid.
- ④ Hyperbolic Tangent.
- ⑤ ReLU.

Funções de Normalização

- 1 Normaliza entre 0 e 1 as saídas dado um problema multi-classes processado por um neurônio.
- 2 É aplicada na camada de saída (output layer).
- 3 Tem como objetivo encaixar a saída obtida dentro de uma classe presente em um conjunto de classes previamente estabelecido.

$$f(v_i) = \frac{e^{v_i}}{\sum_j e^{v_j}}$$

Figura: Função softmax.

- Operação matemática entre duas funções f e g , produzindo uma terceira função, que pode ser interpretada como uma função modificada de f .
- Input (imagem) X Kernel.
- Gera Feature Maps.

Convolução

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

 \times

	0	1	0	
	0	0	0	
	0	0	0	

 $=$

		42		

Figura: Operação de Convolução.

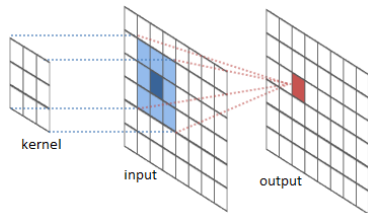


Figura: Relação entre o kernel e a entrada.

- Operação de Agregação.
- Downsampling.
- Redução de escala através de Convolução.
- Max, Avg, Mediana.

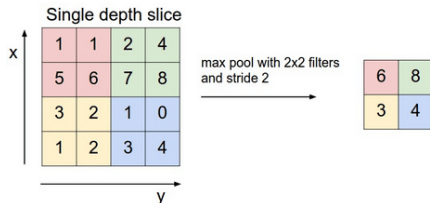


Figura: Pooling de Máximo.

- Estudar um dia antes da prova == Shallow Learning (Aprendizado Raso).
- Estudar todo dia desde o início da disciplina == Deep Learning (Aprendizado Profundo).
- Maior quantidade de dados e processamento em GPU.

- Representação em mais níveis de abstração e maior complexidade de classificação.
- Quanto maior quantidade de dados para o treino e mais épocas, maior o aprendizado.
- Visão Computacional, Processamento de Áudio, Big Data.

- Inspiradas no modelo biológico da visão.
- Camadas de Convolução e Pooling.
- LENET-5 (Yann LeCun, 1998).
- Veio a ter popularidade a partir de 2006 por conta das GPUs.
- Crescimento do uso a partir de 2010 por conta da disponibilidade de grandes massas de dados.

Camadas de uma CNN

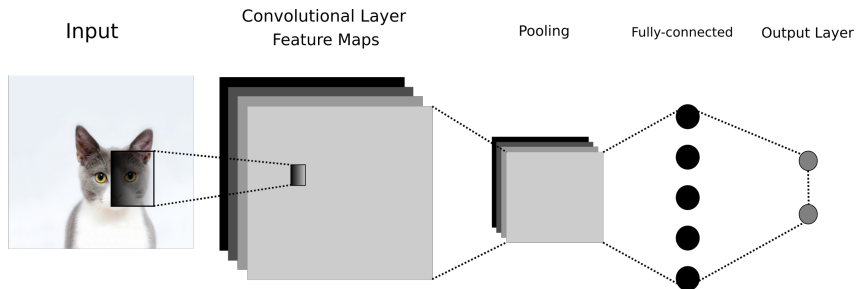


Figura: Camada de entrada, camada convolucional, camada de pooling, camada fully-connected (densa), camada de saída.

Generative Adversarial Networks

- Generative
 - Aprenda um modelo generativo.
- Adversarial
 - A rede é treinada juntamente com uma outra rede adversária, aprendendo a partir de uma competição.
- Networks
 - Redes de Aprendizado Profundo.

Generative Adversarial Networks

- Ian Goodfellow et al, 2014.
- Gerador contra Discriminador.
- Gerador irá gerar dados a partir de um dataset inicial.
- Discriminador irá avaliar esses dados de acordo com o dataset inicial.

Arquitetura GANs

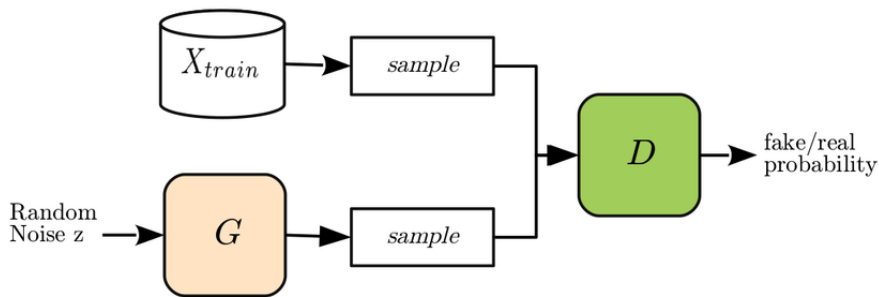


Figura: Gerador procura aproximar o ruído inicial de uma imagem real, enquanto que o Discriminador avalia com base nos dados reais.

- Geração de imagens e dados.
- Super resolução de imagens.
- Recuperação de imagens.
- Extração de objetos presentes numa imagem.
- Texto para imagem.
- Detecção de anomalias.

- Aprendizado não-supervisionado.
- Tenta aprender uma função de identidade.
- Comprime a entrada e tenta descomprimir para que a saída seja igual a ela.
- $y^{(i)} = x^{(i)}$

Arquitetura Autoencoders

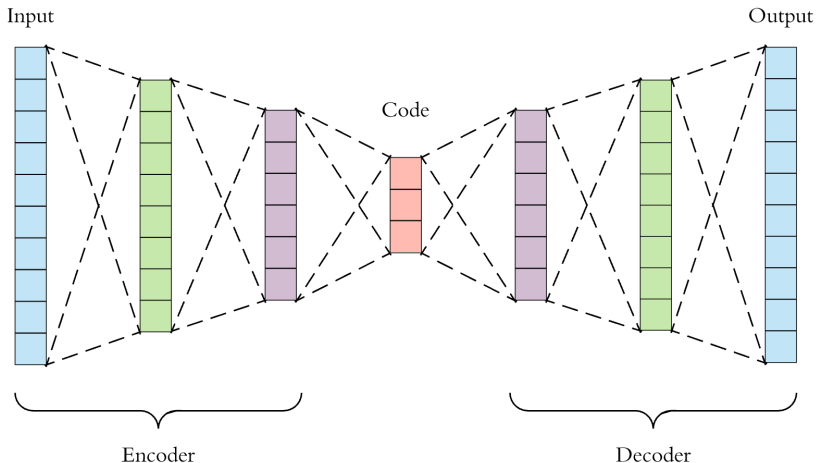


Figura: Encoder irá comprimir os dados, enquanto que o Decoder irá descomprimir.

- Biblioteca de alto nível para Deep Learning.
- Feita em Python.
- Tensorflow ou Theano como backends.

Fluxo no Keras

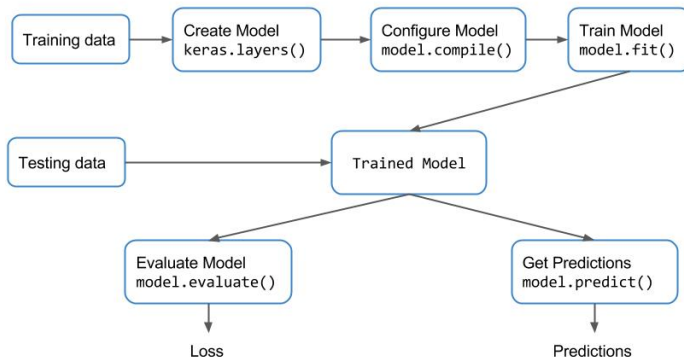


Figura: Relação entre o treino e teste no Keras.

Camadas presentes no Keras

- Dense (`keras.layers.Dense`).
- Activation (`keras.layers.Activation`).
- Dropout (`keras.layers.Dropout`).
- Flatten (`keras.layers.Flatten`).
- Convolution layer (`keras.layers.Conv2D`).
- Pooling layer (`keras.layers.MaxPooling2D`).
- Upsampling layer (`keras.layers.UpSampling`).

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Activation

model = Sequential()
model.add(Dense(10, input_shape=(48, 48, 3)))
model.add(Activation('linear'))
```

```
model.compile(optimizer='adam',  
              loss='categorical_crossentropy',  
              metrics=['accuracy'])  
  
model.fit(X, Y, batch_size=4, epochs=100,  
          validation_data=(X_test, Y_test))  
  
model.save_weights('saved.weights')  
model.save('saved.model')
```

```
model.load_weights('saved.weights')

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

prediction = model.predict(data)
```

- How Are You: [cjlcarvalho/how-are-you](#)
- ulcer-image-segmentation: [cjlcarvalho/ulcer-image-segmentation](#)
- Perceptron: [cjlcarvalho/hands-on-ann/perceptron_keras](#)

Perguntas?

<https://github.com/cjlcarvalho>