

Bioinformatics
Multidisciplinary
Environment

Centro
Multiusuário
de Bioinformática



Utilizando Keras para Construir Redes Neurais

Daniela Coelho Batista Guedes Pereira

Material disponível em:

```
git clone https://github.com/danielacbgp/GeracaodeAplicativos_Keras
```



Agenda

✓ Aprendizagem de Máquina

✓ Rede Neural

✓ Keras

✓ Google Colaboratory

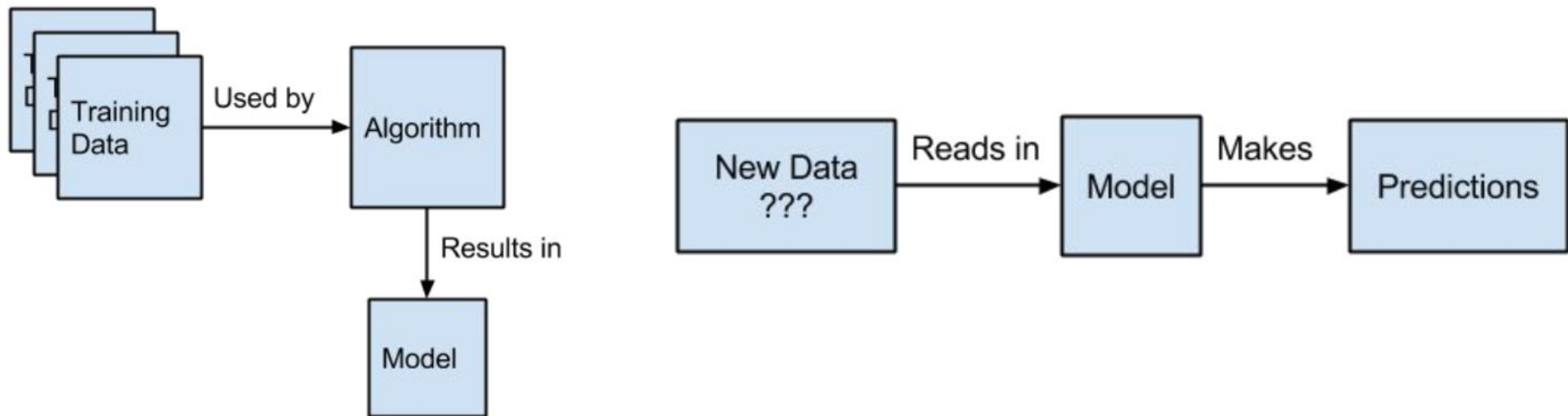
Aprendizagem de máquina



Aprendizagem de Máquina: Campo de estudo
que dá aos computadores a habilidade de aprender
sem serem explicitamente programados

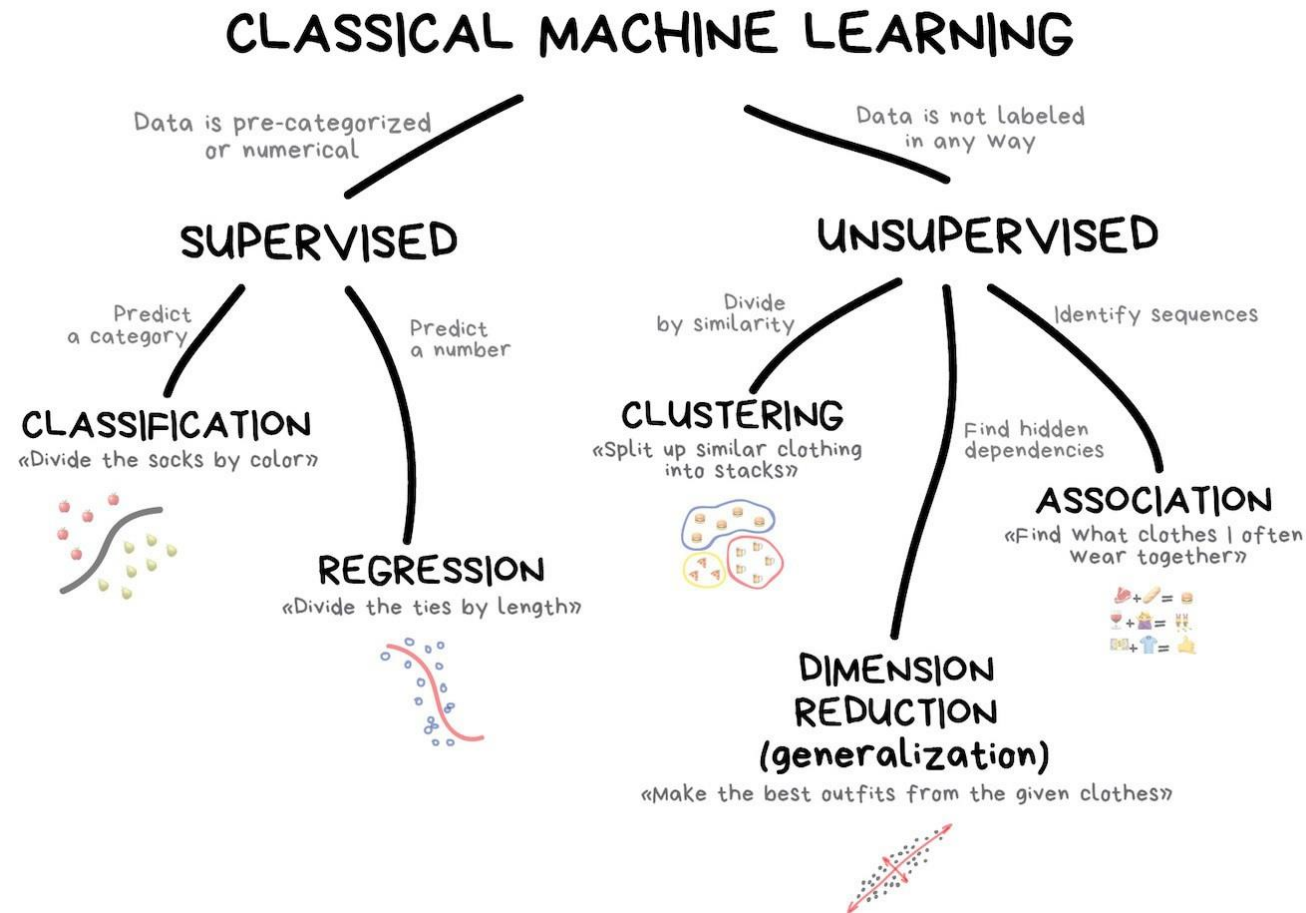
Arthur Samuel, 1959

Aprendizagem de Máquina

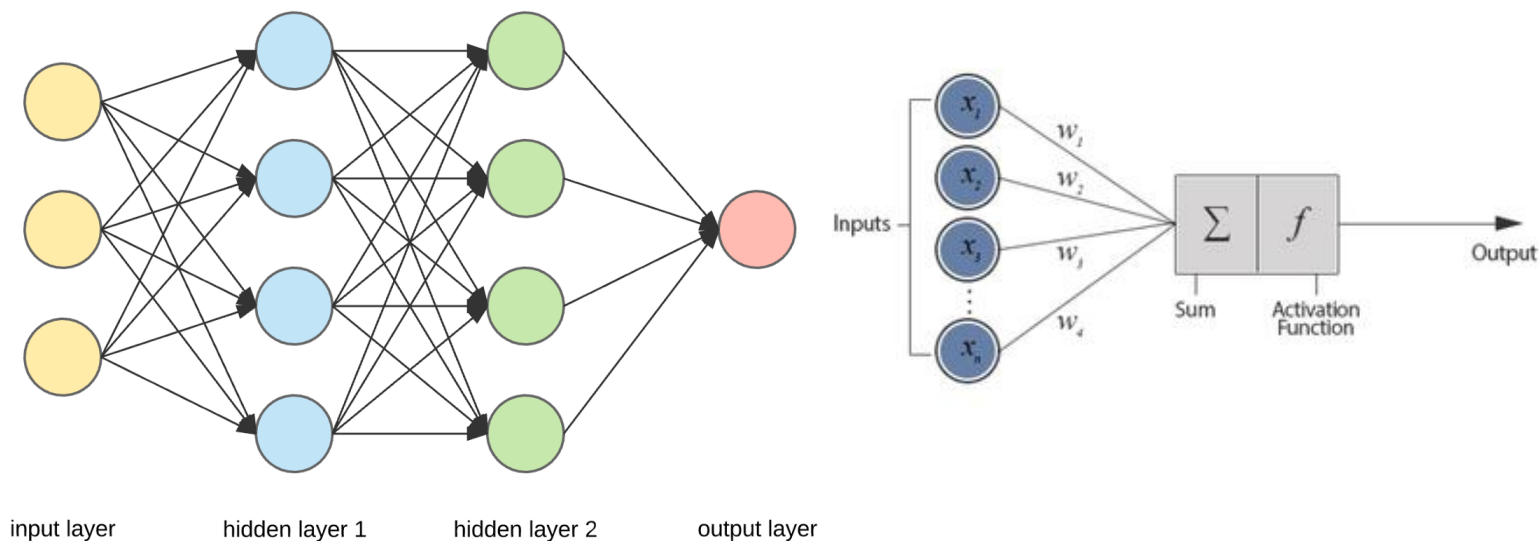


Tais algoritmos operam construindo um modelo a partir de inputs amostrais a fim de fazer previsões ou decisões guiadas pelos dados ao invés de simplesmente seguindo inflexíveis e estáticas instruções de programação tradicional

Tipos de Sistemas de Aprendizagem de Máquina



Rede Neural

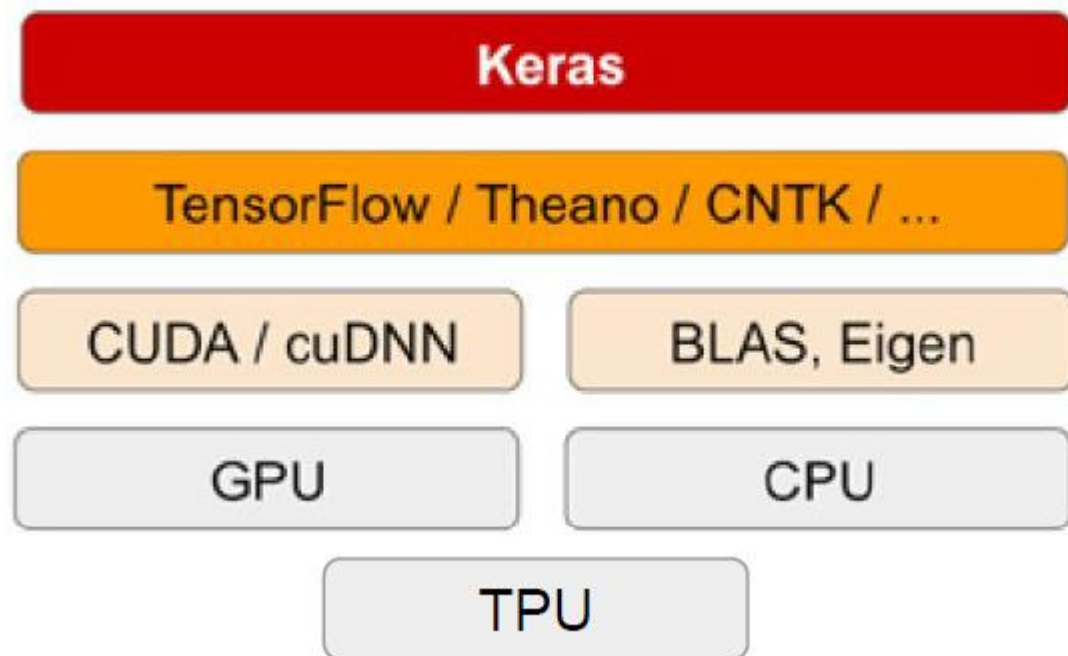


Os modelos de redes neurais foram inspirados na estrutura dos neurônios no nosso cérebro e nas mensagens passando entre neurônios

O que é Keras?

Keras é uma biblioteca, escrita em Python, capaz de executar sobre TensorFlow, CNTK ou Theano.

Fornece uma API de redes neurais de alto nível para desenvolver e avaliar modelos de *deep learning*, de forma rápida e fácil de entender.



Características:

- Permite que o mesmo código seja executado sem problemas em CPU, GPU ou TPU.
- Suporta vários tipos de arquiteturas de rede.
- Pode ser usado livremente em projetos comerciais.
- Compatível com qualquer versão do Python de 2.7 a 3.6

Keras - Workflow

❑ Especificação da Arquitetura

- Quantas camadas?
- Quantos nós em cada camada?
- Qual função de ativação será usado em cada camada?

```
In [1]: import numpy as np
```

```
In [2]: from keras.layers import Dense
```

```
In [3]: from keras.models import Sequential
```

```
In [4]: predictors = np.loadtxt('predictors_data.csv', delimiter=',')
```

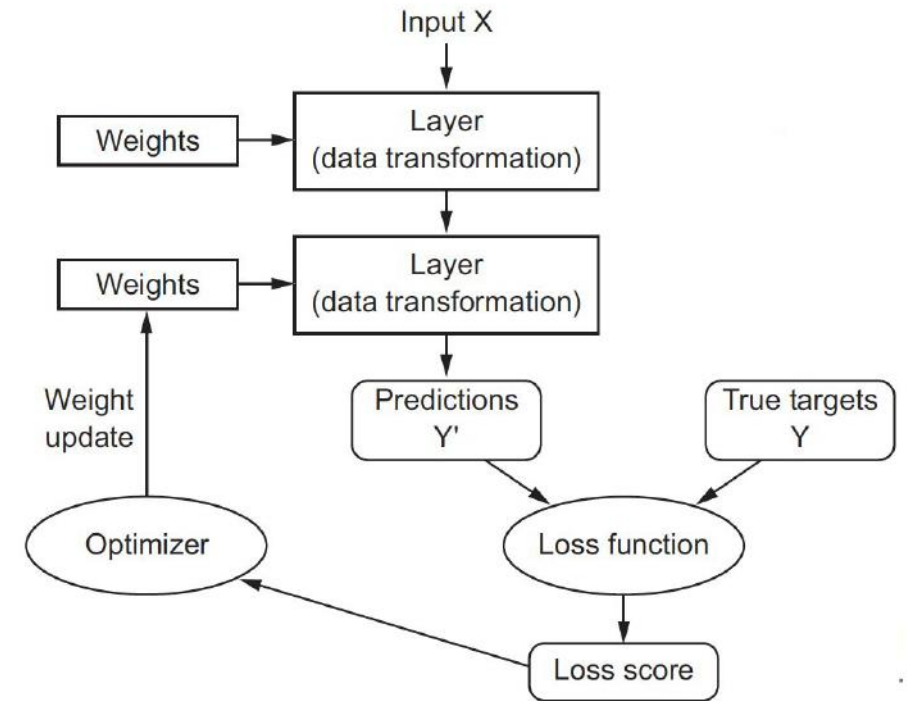
```
In [5]: n_cols = predictors.shape[1]
```

```
In [6]: model = Sequential()
```

```
In [7]: model.add(Dense(100, activation='relu', input_shape = (n_cols,)))
```

```
In [8]: model.add(Dense(100, activation='relu'))
```

```
In [9]: model.add(Dense(1))
```



Keras - Workflow

❑ Compilação do Modelo

- Loss function
- Qual otimizador

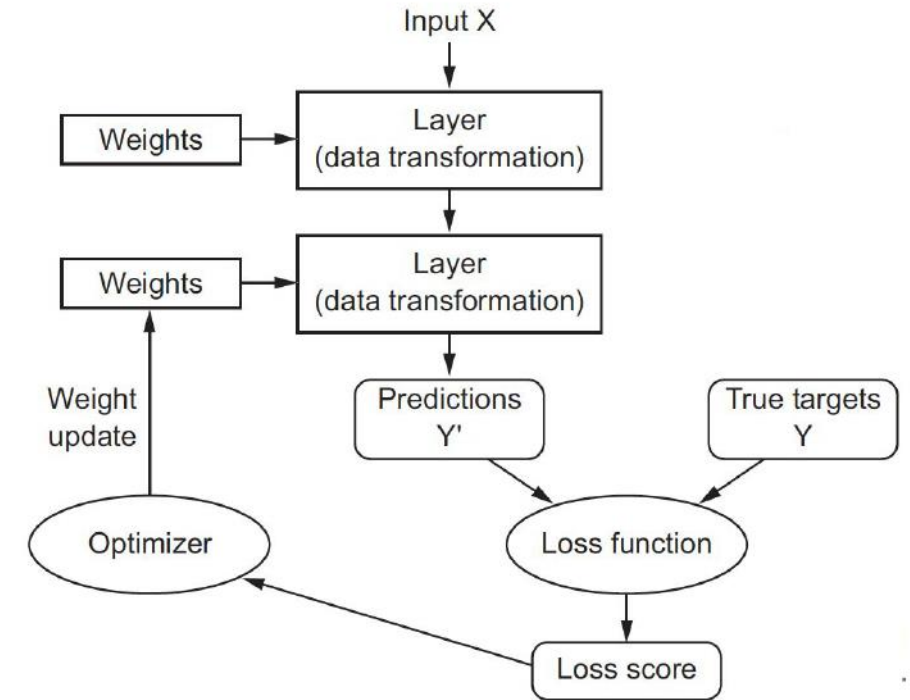
❑ Fit

- Ciclo de *back-propagation* e otimização dos pesos do modelo com seus dados.

```
In [1]: model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics=['accuracy'])
```

```
In [2]: model.fit(predictors, target, validation_split=0.3)
```

```
Epoch 1/10  
89648/89648 [=====] - 3s - loss: 0.7552 - acc: 0.5775 - val_loss: 0.6969 -  
val_acc: 0.5561  
Epoch 2/10  
89648/89648 [=====] - 4s - loss: 0.6670 - acc: 0.6004 - val_loss: 0.6580 -  
val_acc: 0.6102  
...  
Epoch 8/10  
89648/89648 [=====] - 5s - loss: 0.6578 - acc: 0.6125 - val_loss: 0.6594 -  
val_acc: 0.6037  
Epoch 9/10  
89648/89648 [=====] - 5s - loss: 0.6564 - acc: 0.6147 - val_loss: 0.6568 -  
val_acc: 0.6110  
Epoch 10/10  
89648/89648 [=====] - 5s - loss: 0.6555 - acc: 0.6158 - val_loss: 0.6557 -  
val_acc: 0.6126
```



Keras - *Workflow*

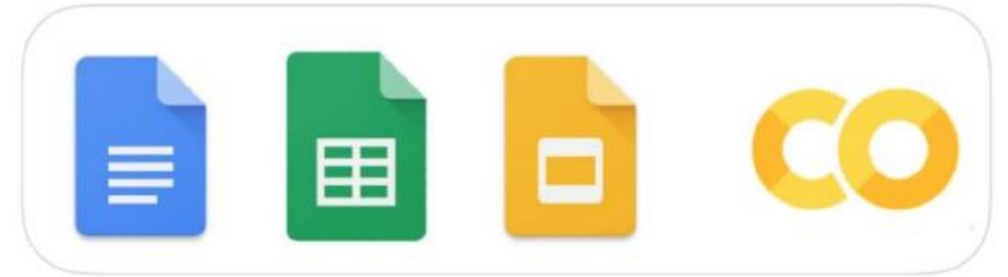
❑ Utilização do modelo para realizar previsões

Problem type	Last-layer activation	Loss function
Binary classification	sigmoid	binary_crossentropy
Multiclass, single-label classification	softmax	categorical_crossentropy
Multiclass, multilabel classification	sigmoid	binary_crossentropy
Regression to arbitrary values	None	mse
Regression to values between 0 and 1	sigmoid	mse or binary_crossentropy

Google Colaboratory



<https://colab.research.google.com/>



Colaboratory é um projeto de pesquisa do Google criado para ajudar a divulgar educação e pesquisa em aprendizagem de máquina. É um ambiente de notebook Jupyter que não requer configuração para ser executado e é executado inteiramente na nuvem.

Os *notebooks* colaborativos são armazenados no Google Drive e podem ser compartilhados da mesma forma que você faria com o Google Docs ou Sheets. O Google Colaboratory é gratuito para uso