

# Machine Learning Tutorial

Feito por Thyago Capitaniao

O objetivo aqui foi justamente treinar a aplicação prática de como se criar uma rede neural do tipo e tentar entender a teoria e com ela fazer os comentários. Minha base teórica ainda é muito superficial e não saberia explicar muitas das decisões tomadas aí, principalmente quando se trata das camadas, mas esse tutorial me fez ter uma compreensão maior do que é o desafio de criar redes neurais e deep learning.

## ***Passo 1: Imports***

1. Tensorflow para o modelo
2. Matplotlib para plotar as imagens

In [2]:

```
import tensorflow as tf
import matplotlib.pyplot as plt

tf.__version__
```

Out[2]:

'2.0.0'

## ***Passo 2: obter base de dados e especificar listas de treinamento do modelo***

Pegar o dataset com o qual eu vou trabalhar. No caso, se trata de um dataset de números escritos na maneira cursiva. Meu objetivo é fazer o modelo aprender a identificar os números.

In [3]:

```
mnist = tf.keras.datasets.mnist
```

In [4]:

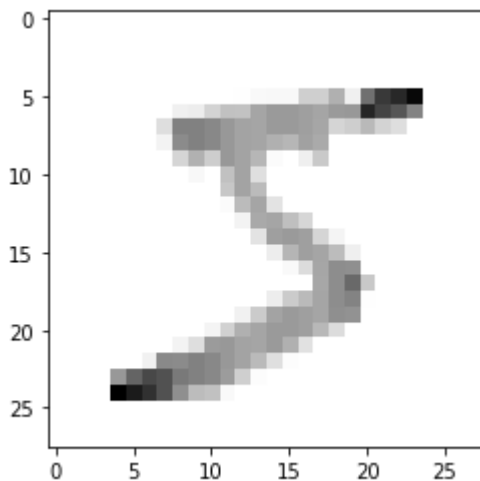
```
(x_train,y_train),(x_test,y_test) = mnist.load_data()

x_train = tf.keras.utils.normalize(x_train,axis=1)
x_test = tf.keras.utils.normalize(x_test,axis=1)
```

Exemplo de um dos items do dataset

In [5]:

```
plt.imshow(x_train[0], cmap = plt.cm.binary)
plt.show()
```



### ***Passo 3: Especificação do modelo de treino***

Onde o treinamento do modelo de fato acontece, passando por três camadas

In [ ]:

```
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(128,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(128,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(10,activation=tf.nn.softmax))

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

model.fit(x_train, y_train)
```

Train on 60000 samples

Teste para saber se houve overfit

In [ ]:

```
val_loss, val_acc = model.evaluate(x_test, y_test)

print(val_loss, val_acc)
```

In [ ]:

```
import matplotlib.pyplot as plt

plt.imshow(x_train[0])
plt.show()
```

Salvando o Modelo

In [ ]:

```
model.save('epic_num_reader.model')
```

In [ ]:

```
new_model = tf.keras.models.load_model('epic_num_reader.model')
```

Exemplo de um item para verificar se o modelo foi bem sucedido

In [ ]:

```
predictions = new_model.predict([x_test])
```

In [ ]:

```
import numpy as np
```

In [ ]:

```
print(np.argmax(predictions[0]))
```

In [ ]:

```
plt.imshow()
```