

# Relatório 01 - O Que é Aprendizado de Máquina

Guilherme Loan Schneider

## Descrição da atividade

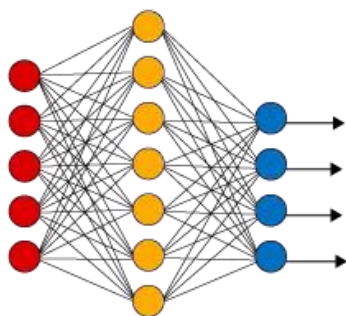
A ideia fundamental do aprendizado de máquina consiste em utilizar dados computacionais e algoritmos com o objetivo de simular e replicar o processo de pensamento e tomada de decisões de um ser humano. Esta abordagem busca criar sistemas inteligentes capazes de aprender padrões, fazer previsões e se adaptar a novas informações, de forma semelhante ao cérebro humano, sem serem explicitamente programados para cada situação.

É possível perceber no vídeo Redes Neurais e Machine Learning que, já na década de 50 estavam sendo feitas tentativas de criar um dispositivo computacional autônomo, como é o caso do PERCEOTRON, um equipamento destinado ao reconhecimento de palavras escritas e voz. No entanto o seu desenvolvimento estava fadado as limitações tecnológicas da época.

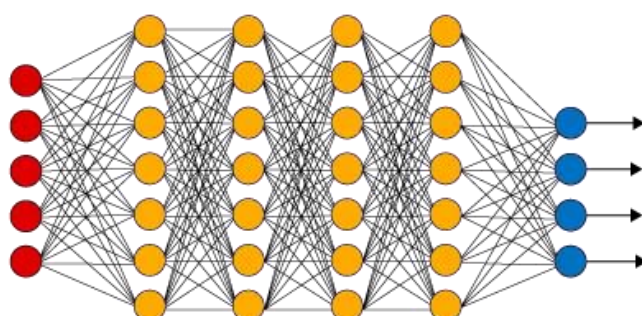
A analogia utilizada pelo autor do vídeo, onde um grupo de WhatsApp com várias pessoas decidindo se vale a pena comprar um produto por determinado preço, sem muitas vezes sequer conhecer o quê seja, demonstra o exato funcionamento dessas redes neurais, onde os indivíduos do grupo são a nossa rede, nós passamos como entrada um determinado produto e como resultado, desejamos saber se vale a pena aquele produto ou não.

Na figura abaixo, é representado o que é dito no vídeo, onde no modelo Simple Neural Network, é utilizado apenas uma camada oculta (consiste nos indivíduos da nossa rede), indicando que ali ocorrerá todo o processamento da rede. Já no caso do modelo Deep Learning Neural Network, são utilizadas várias camadas ocultas, onde cada uma delas desempenhará um papel de decisão diferente na rede, ou seja, existirão mais conceitos de avaliação de determinada entrada.

### Simple Neural Network



### Deep Learning Neural Network



● Input Layer    ● Hidden Layer    ● Output Layer

Agora, se nós quiséssemos melhorar essa rede, é possível fazer com que toda a rede passe a saber o que são produtos eletrônicos, carros, objetos variados, dentre outros, utilizando um treinamento que consiste em técnicas e algoritmos a fim de familiarizá-los àqueles dispositivos.

Na aula Introdução ao Machine Learning é explicitado que a ideia desse conteúdo é fazer uma programação reversa, onde você conhece as respostas de determinado problema, e você deseja que a máquina descubra, de forma independente, essas respostas estipuladas. Por exemplo, eu sei que um carro possui quatro rodas, quatro portas, um capô, uma tampa de porta-malas, um chassi, dentre outros, no entanto, um computador não possui esse conhecimento. Logo, para que seja possível ele reconhecer essa informação, por exemplo, do que é uma porta, é necessário que ele seja treinado com imagens, onde é especificado o formato de uma porta, a moldura, as borrachas vedantes, etc.

O código abaixo apresenta as descrições de cada seção utilizada.

```
import tensorflow as tf

import numpy as np

from tensorflow import keras

# Aqui é importado a API de redes neurais utilizada no exemplo da aula


model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
#Nessa seção de Código, é definido o modelo de rede neural, sendo a Sequential
#a mais simples possível.

#Ela possui apenas um neurônio (units) e um valor de entrada (input_shape).


model.compile(optimizer='sgd', loss='mean_squared_error')

#A função “optimizer” está desempenhando o papel de gerar uma suposição para o
#problema utilizando com base o valor obtido na função “loss”, que calcula a
#diferença entre a saída prevista pelo modelo e o valor real.


#Definição dos nossos dados

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

#Basicamente é definido um array de valores com uma relação entre si ( $Y = 3X + 1$ ) e a ideia é que a rede neural consiga entender essa relação.


model.fit(xs, ys, epochs=500)

#Essa linha de código é responsável por treinar um conjunto de dados de entrada
#(xs) e saída (ys) ao longo de várias iterações.

#O número de epochs define o tanto de vezes que o modelo passará pelos dados,
#ajustando seus parâmetros a cada iteração, a fim de diminuir o valor da função
#loss. Nesse caso não é necessário utilizar tantas iterações, visto que pode
#causar overfitting, ou seja, ele poderá perder a capacidade de prever novos
#resultados.


print(model.predict([10.0]))

#Por fim, após passar pela fase de treinamento, o modelo fará a predição de Y
#recebendo o valor X = 10.
```

Na figura abaixo estão os resultados da função loss, que atingiu uma taxa de erro baixa, após rebalancear os pesos da sua rede neural ao longo das iterações.

```
Epoch 484/500
1/1 ██████████ 0s 25ms/step - loss: 3.0183e-06
Epoch 485/500
1/1 ██████████ 0s 24ms/step - loss: 2.9563e-06
Epoch 486/500
1/1 ██████████ 0s 23ms/step - loss: 2.8956e-06
Epoch 487/500
1/1 ██████████ 0s 24ms/step - loss: 2.8361e-06
Epoch 488/500
1/1 ██████████ 0s 25ms/step - loss: 2.7778e-06
Epoch 489/500
1/1 ██████████ 0s 23ms/step - loss: 2.7209e-06
Epoch 490/500
1/1 ██████████ 0s 24ms/step - loss: 2.6648e-06
Epoch 491/500
1/1 ██████████ 0s 24ms/step - loss: 2.6103e-06
Epoch 492/500
1/1 ██████████ 0s 25ms/step - loss: 2.5566e-06
Epoch 493/500
1/1 ██████████ 0s 24ms/step - loss: 2.5039e-06
Epoch 494/500
1/1 ██████████ 0s 25ms/step - loss: 2.4525e-06
Epoch 495/500
1/1 ██████████ 0s 24ms/step - loss: 2.4020e-06
Epoch 496/500
1/1 ██████████ 0s 23ms/step - loss: 2.3525e-06
Epoch 497/500
1/1 ██████████ 0s 39ms/step - loss: 2.3042e-06
Epoch 498/500
1/1 ██████████ 0s 25ms/step - loss: 2.2573e-06
Epoch 499/500
1/1 ██████████ 0s 24ms/step - loss: 2.2107e-06
Epoch 500/500
1/1 ██████████ 0s 23ms/step - loss: 2.1654e-06
```

## Conclusões

Nesta aula foi possível compreender um pouco do funcionamento de uma rede neural, bem como ela funciona, alguns tipos de redes neurais e aplicações práticas desse conteúdo. Além disso, foi possível entender, de forma prática, um cenário de machine learning, onde a máquina precisou efetuar iterações a fim de descobrir os padrões daquele problema.

## Referencias

NERDOLOGIA. Redes Neurais e Machine Learning | Nerdologia Tech. YouTube, 12 set. 2019. Disponível em: [https://www.youtube.com/watch?v=1\\_c\\_MA1F-vU](https://www.youtube.com/watch?v=1_c_MA1F-vU). Acesso em: 08 out. 2024.

TENSORFLOW. Introdução ao Machine Learning (ML de Zero a 100, parte 1). YouTube, 7 jan. 2020. Disponível em: <https://www.youtube.com/watch?v=t5z5lyrb7s&list=RDCMUC0rqcBdTuFTjJiefW5t-IQ>. Acesso em: 08 out. 2024.