

## **Relatório 1 - O Que é Aprendizado de Máquina (I)**

### **Higor Miller Grassi**

Após os anos 2000, as redes neurais começaram a ascender novamente, por conta devido ao aumento do poder computacional, com essa nova onda, trouxeram avanços significativos com a popularização do mesmo.

As redes neurais agem como se fossem cérebros humanos consistindo basicamente em unidades que são chamadas de neurônios que ao se interligarem, são capazes de processar e apurar informações. Assim, agindo basicamente igual ao cérebro humano usando o aprendizado de máquina para obter resultados.

No contexto do aprendizado de máquina, dados são as informações utilizadas para treinar um modelo, já os rótulos são as respostas ou categorias associadas a esses dados, como por exemplo, em um conjunto de dados de imagens, as imagens seriam os dados, e os rótulos poderiam indicar se cada imagem contém x ou y.

O modo de backpropagation, faz com que cada conexão feita pelas redes neurais obtém pesos associados onde ele é moldado durante o treinamento, dando como mais ou menos importante uma entrada perante aos acertos mais frequentes, podendo impactar no resultado final.

Os teoremas de aproximação universal, na área matemática, existe uma sequência de redes neurais que pode se aproximar dessa função, resumindo, se você quiser que a rede aprenda a fazer algo, ela pode fazer isso se tiver neurônios suficientes, ou seja, para uma função Rede Neural Feedforward (fff), é possível encontrar sequências de redes neurais que se aproximam de fff .

A rede neural feedforward seria um modelo unidirecional, com entrada para a saída, sendo usada mais comumente em reconhecimento de padrões e previsão de séries.

Visão computacional é um campo da inteligência artificial que ensina os computadores a interpretar e compreender o conteúdo visual do mundo, como imagens e vídeos. Isso inclui tarefas como detecção de objetos e reconhecimento facial. As redes neurais, especialmente as CNNs, têm sido fundamentais no avanço da visão computacional, permitindo que máquinas alcancem desempenho de nível humano em várias tarefas.

Redes neurais convolucionais (CNNs) são um tipo de rede neural projetada para entender e classificar imagens, funcionam bem porque podem lidar com dados tridimensionais, obtendo várias camadas:

- Camada de entrada: recebe a imagem
- Camadas Convolucionais: Elas “convoluem” a imagem, procurando padrões e características
- Camadas de Agrupamento: Simplificam a informação, resumindo as características mais importantes
- Camadas Totalmente Conectadas: São as camadas finais que tomam a decisão sobre o que a imagem representa

Sendo a mais importante e convolucionais, utilizando um filtro que ao passar pela imagem conseguem identificar características específicas como padrões e texturas, realizando cálculos entre os pixels da imagem e os valores, esse processo é chamado de convolução.

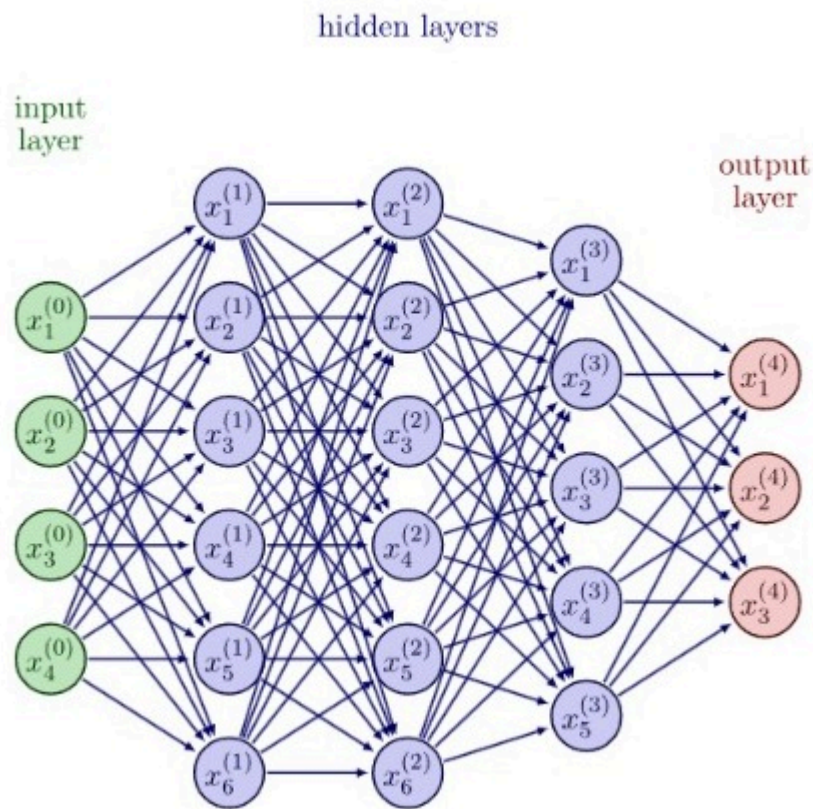


Figura 1 (referenciada): Estrutura típica de uma RNA totalmente conectada, compreendendo uma camada de entrada, várias camadas ocultas e uma camada de saída. Cada seta de uma fonte para um neurônio alvo indica que o valor de saída da fonte é usado como um valor de entrada para o alvo.

**Camada de Entrada:** responsável por receber os dados de entrada. Ela não realiza cálculos complexos; simplesmente passa as informações para as camadas subsequentes.

**Camada Oculta:** ocorre a maior parte da computação na rede, responsáveis principalmente por processar as informações recebidas da camada de entrada, realizando operações que permitem aprender padrões complexos, podendo ter várias camadas ocultas, e o número de neurônios em cada camada pode variar.

**Camada de Saída:** recebe os resultados das camadas ocultas e faz os cálculos finais para produzir a saída da rede

Código do vídeo:

```
1 import tensorflow as tf
2 import numpy as np
3 from tensorflow import keras
4
5 model = tf.keras.Sequential([
6     keras.Input(shape=(1,)),
7     keras.layers.Dense(units=1)
8 ])
9
10 model.compile(optimizer='sgd', loss='mean_squared_error')
11
12 xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
13 ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)
14
15 model.fit(xs, ys, epochs=500)
16
17 print(model.predict(np.array([10.0])))
```

O código cria um modelo de rede neural que tenta aprender a relação entre os dados de entrada designados com xs e os dados de saída designados como ys. Depois de treinar o modelo com os dados, ele é usado para prever o resultado para um novo valor (10.0), sendo assim, a saída deve conter um número que representa essa previsão, baseada no que o modelo aprendeu durante o treinamento.

Resultado:

```
Epoch 488/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 3.4423e-07
Epoch 489/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 3.3708e-07
Epoch 490/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 3.3016e-07
Epoch 491/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 11ms/step - loss: 3.2335e-07
Epoch 492/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 3.1674e-07
Epoch 493/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 3.1024e-07
Epoch 494/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 3.0388e-07
Epoch 495/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 13ms/step - loss: 2.9765e-07
Epoch 496/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 2.9152e-07
Epoch 497/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 13ms/step - loss: 2.8556e-07
Epoch 498/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 2.7962e-07
Epoch 499/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 2.7390e-07
Epoch 500/500
+-----+
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 12ms/step - loss: 2.6828e-07
+1m1/1+0m +32m+-----+0m+37m+0m +1m0s+0m 18ms/step
[[30.998487]]
```

Referências: Alexandre, The Universal approximation theorem. Deep-mind, 2023.  
Disponível em:

<https://www.deep-mind.org/2023/03/26/the-universal-approximation-theorem/>

Referências: IBM,What are convolutional neural networks?. IBM.

Disponível em: <https://www.ibm.com/topics/convolutional-neural-networks>