Relatório do bootcamp - 1

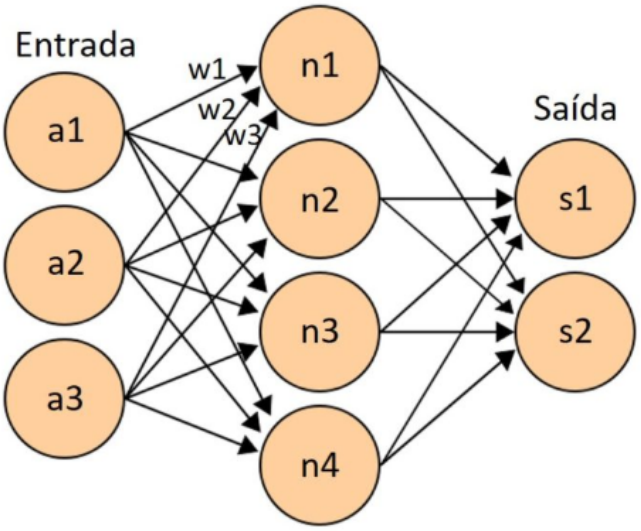
Thassiana C. A. Muller

Redes Neurais e Machine Learning | Nerdologia Tech :

Nesse vídeo ficou fácil de visualizar a interação entre os neurônios, o peso da decisão de cada um e a integração entre as camadas de decisões, também deixa mais claro o poder das redes neurais para problemas não claramente definidos, como uma corrida que pode ter seu trajeto alterado.

Os neurônios são pequenas unidades de processamento que realizam operações simples e repassam seus resultados adiante.

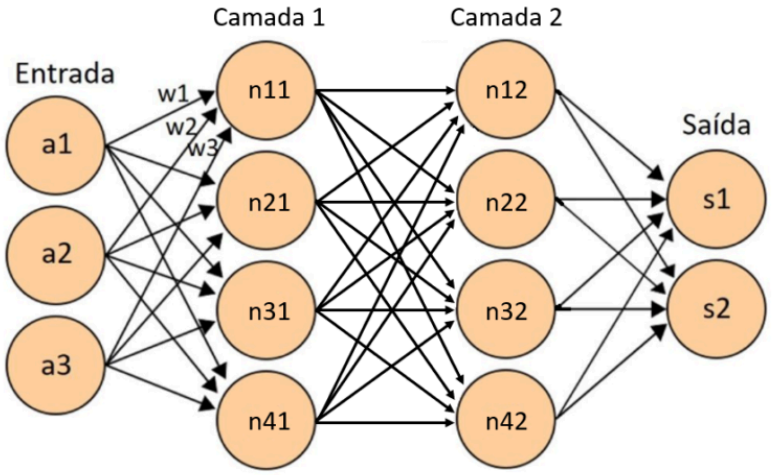
As redes neurais podem ser desenvolvidas com apenas uma camada de neurônios, conhecidas como redes neurais rasas, de forma que cada neurônio dará seu peso, suas operações e seu *bias*[1] para cada dado da entrada. Dessa forma, a resposta se dará a partir do neurônio de saída com o maior valor ativado. Esse tipo de rede neural, resolve problemas mais simples como a ação de comprar ou não a obra de arte no contexto do vídeo.



Porém para problemas mais complexos o *deep learning* passa a ser necessário. Esse tipo de rede neural possui várias camadas intermediárias de neurônios que vão proporcionar um ajuste mais fino ao problema, porém ao se projetar esse tipo de solução é importante atentar-se para o fenômeno chamado de overfitting que é quando o modelo se ajusta demais ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados.

---

[1] Constante que é adicionada ao somatório ponderado das entradas de um neurônio antes de passar pela função de ativação. O bias ajuda a ajustar a saída do neurônio de maneira que a rede neural possa modelar melhor os dados e resolver problemas de aprendizado.



Introdução ao Machine Learning (ML de Zero a 100, parte 1)

Esse vídeo mostra a diferença entre os algoritmos e programas tradicionais, que entra-se com dados e o que deve ser feito e obtém-se uma resposta, e a programação para machine learning, que entra-se com dados e respostas esperadas e espera-se como saída as regras. Pude ver no código de exemplo do vídeo a reação do programa a tentativa e erro para melhor ajuste preditivo. Dessa forma, escrevi outro código para, desta vez, analisar o aprendizado em cima da equação **y = (x*6) - (x-3)**

```
x = [-1.0, 0.0, 1.0, 2.0, 3.0, 4.0, 8.0, 10.0, 15.0]

y = []
for num in x:
    y.append((num*6)-(num-3))

print(y)
#y = [-2.0, 3.0, 8.0, 13.0, 18.0, 23.0, 43.0, 53.0, 78.0]
```

```
import tensorflow as tf
from tensorflow import keras
import numpy as np

model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

x = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0, 8.0, 10.0, 15.0], dtype=float)
```

```
y = np.array([-2.0, 3.0, 8.0, 13.0, 18.0, 23.0, 43.0, 53.0, 78.0],
dtype=float)

model.fit(x,y, epochs=500)

print(model.predict([12.0]))
```

Saída:

```
[[63.002693]]
```

Houve um erro de 0.002693 do resultado esperado

Referencias:

Tudo sobre Redes Neurais e Deep Learning - Entenda. Disponível em: <https://didatica.tech/introducao-a-redes-neurais-e-deep-learning/>.