



1 - Vídeo: O Que é Aprendizado de Máquina (I)

Redes Neurais e Machine Learning

- Explicação da origem e do inverno das IAs onde houve uma grande baixa de investimento devido a falta de dados e tecnologia para continuar desenvolvendo. Ressurgiu nos anos 2000.
- Explicação sobre como as redes neurais imitam o funcionamento do cérebro e aprendem com o Perceptron.
- A importância das interações na calibração de redes neurais artificiais.
 - Deu um exemplo de avaliação de obras de arte, de criar um grupo com amigos para tentar avaliar uma obra de arte, no entanto contatar um especialista seria mais eficiente
 - Organizam os neurônios em várias camadas para otimizar a avaliação de itens.
 - Calibrar a rede neural com base em feedbacks pode ser mais demorado e caro do que seguir regras claras, mas resulta em soluções mais precisas
- Processo de rede convolucional
 - Imagens psicodélicas que exageram os parametros, foi o que ajudou a ter a IA que temos hoje.

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

- Explica como faria o computador entender o que é um jogo como Pedra, Papel, Tesoura.
- O que difere ML da programação convencional é que damos os dados e ele identifica os padrões, como identificar nas imagens o que é pedra, o que é papel e o que é tesoura
- Um exemplo de como podemos fazer isso em código:
 - Fazemos uma suposição inicial, e nesse caso mandamos o modelo tentar 500 vezes a se aproximar do padrão. Como se estivessemos tentando convergir pontos de X e Y em uma mesma reta. Dessa forma, vamos ajustando a precisão do modelo.
- 1. Linha - Define o modelo entre si, tem 1 único neurônio. Forma de entrada apenas 1 valor
- 2. Linha - Quando compilar temos o otimizador e a perda, o modelo faz uma suposição da relação entre os números. A perda fala quão ruim está, assim o otimizador consegue ir otimizando o máximo possível.
- 3. Os dados - são configurados em matriz de x e y. Os processo de combiná-los está no ajuste do modelo, nós literalmente dizemos: encaixe x no y e tente 500 vezes

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])  
model.compile(optimizer='sgd', loss='mean_squared_error')  
  
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)  
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)  
  
model.fit(xs, ys, epochs=500)  
  
print(model.predict([10.0]))
```