No primeiro vídeo é abordado a origem da inteligência artificial e uma metáfora sobre seu funcionamento utilizando especialistas numa loja de penhores, onde cada especialista seria um nodo da rede neural.

Uma rede neural possui vantagens e desvantagens sobre algoritmos programados, no caso os algoritmos possuem um tempo menor de execução, porém deixam a desejar no quesito adaptabilidade, pois o algoritmo programado não é capaz de se adequar a uma mudança no prompt, já a rede neural consegue criar novos caminhos para chegar ao resultado desejado, se assemelhando ao processo de tomada de decisão efetuado pelo cérebro.

Como utilizar uma rede neural muito grande seria um processo caro e lento, existem alguns métodos de otimizar as redes, um exemplo seria uma rede neural convolucional, onde se tem um grupo de nodos que podem processar melhor um tipo de entrada específica, caso os mesmos demonstrem uma saída favorável, o processo ja é encerrado antes de passar pela rede inteira, economizando tempo e poder de processamento

O segundo vídeo começa mostrando o que difere o Machine Learning da programação tradicional, no caso em vez do computador receber os dados e o que ele deve fazer com eles para obter o resultado, a entrada será dos dados e do resultado a ser obtido, então o computador deverá encontrar uma forma de tratar os dados para obter o resultado.

Foi apresentado um código utilizando tensorflow e o framework keras para montar uma rede neural básica que é capaz de prever qual será o valor de y utilizando o valor de x e um array de x e y, a predição é realizada quantas vezes for especificado no parâmetro epochs afim de “treinar” a máquina

```
import tensorflow as tf
import numpy as np
from tensorflow import keras
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([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)

model.fit(xs, ys, epochs=300)
print(model.predict([10.0]))
```

Foram realizados dois testes, um com 500 iterações e outro com 300, observa-se que com 500 iterações o valor chegou mais perto de 31, porém a diferença entre 500 e 300 não foi tão grande, portanto deve-se avaliar se é realmente necessário efetuar 500 predições para obter exatidão ou se é mais adequado realizar apenas 300 e economizar tempo/recursos

Teste com 500 iterações

```
1/1 [==============================] - 0s 6ms/step - loss: 2.079
Epoch 491/500
1/1 [==============================] - 0s 9ms/step - loss: 2.0379e-07
Epoch 492/500
1/1 [==============================] - 0s 6ms/step - loss: 1.9952e-07
Epoch 493/500
1/1 [==============================] - 0s 9ms/step - loss: 1.9552e-07
Epoch 494/500
1/1 [==============================] - 0s 7ms/step - loss: 1.9145e-07
Epoch 495/500
1/1 [==============================] - 0s 11ms/step - loss: 1.8756e-07
Epoch 496/500
1/1 [==============================] - 0s 6ms/step - loss: 1.8372e-07
Epoch 497/500
1/1 [==============================] - 0s 6ms/step - loss: 1.7994e-07
Epoch 498/500
1/1 [==============================] - 0s 10ms/step - loss: 1.7623e-07
Epoch 499/500
1/1 [==============================] - 0s 6ms/step - loss: 1.7261e-07
Epoch 500/500
1/1 [==============================] - 0s 6ms/step - loss: 1.6903e-07
1/1 [==============================] - 0s 144ms/step
[[30.998802]]
```

Teste com 300 iterações

```
Epoch 289/300
1/1 [==============================] - 0s 6ms/step - loss: 8.399
Epoch 290/300
1/1 [==============================] - 0s 6ms/step - loss: 8.2273e-06
Epoch 291/300
1/1 [==============================] - 0s 9ms/step - loss: 8.0587e-06
Epoch 292/300
1/1 [==============================] - 0s 10ms/step - loss: 7.8931e-06
Epoch 293/300
1/1 [==============================] - 0s 9ms/step - loss: 7.7312e-06
Epoch 294/300
1/1 [==============================] - 0s 6ms/step - loss: 7.5722e-06
Epoch 295/300
1/1 [==============================] - 0s 8ms/step - loss: 7.4165e-06
Epoch 296/300
1/1 [==============================] - 0s 6ms/step - loss: 7.2645e-06
Epoch 297/300
1/1 [==============================] - 0s 9ms/step - loss: 7.1155e-06
Epoch 298/300
1/1 [==============================] - 0s 6ms/step - loss: 6.9695e-06
Epoch 299/300
1/1 [==============================] - 0s 9ms/step - loss: 6.8259e-06
Epoch 300/300
1/1 [==============================] - 0s 7ms/step - loss: 6.6863e-06
1/1 [==============================] - 0s 92ms/step
[[30.992453]]
```