

```

1  # -*- coding: utf-8 -*-
2  """Red neuronal en Python y Tensorflow
3
4  Automatically generated by Colaboratory.
5
6  """
7
8  import tensorflow as tf
9  import numpy as np
10
11  celsius = np.array([-40, -10, 0, 8, 15, 22, 38], dtype=float)
12  fahrenheit = np.array([-40, 14, 32, 46, 59, 72, 100], dtype=float)
13
14  #capa = tf.keras.layers.Dense(units=1, input_shape=[1])
15  #modelo = tf.keras.Sequential([capa])
16
17  oculta1 = tf.keras.layers.Dense(units=3, input_shape=[1])
18  oculta2 = tf.keras.layers.Dense(units=3)
19  salida = tf.keras.layers.Dense(units=1)
20  modelo = tf.keras.Sequential([oculta1, oculta2, salida])
21
22  modelo.compile(
23      optimizer=tf.keras.optimizers.Adam(0.1),
24      loss='mean_squared_error'
25  )
26
27  print("Comenzando entrenamiento...")
28  historial = modelo.fit(celsius, fahrenheit, epochs=1000, verbose=False)
29  print("Modelo entrenado!")
30
31  import matplotlib.pyplot as plt
32  plt.xlabel("# Epoca")
33  plt.ylabel("Magnitud de pérdida")
34  plt.plot(historial.history["loss"])
35
36  print("Hagamos una predicción!")
37  resultado = modelo.predict([100.0])
38  print("El resultado es " + str(resultado) + " fahrenheit!")
39
40  print("Variables internas del modelo")
41  #print(capa.get_weights())
42  print(oculta1.get_weights())
43  print(oculta2.get_weights())
44  print(salida.get_weights())

```