**Inteligencia artificial (calcular gados Celsius a fahrenheit)**

Para crear nuestra red neuronal debemos de hacer uso del Python y Tensorflow (esta es una librería para uso de Python).

Importamos la librería en Python:

import tensorflow as tf

Además importaremos **Numopy** para trabajar con arreglos numéricos:

import numpy as np

Creamos dos arreglos para ingresar nuestros datos de prueba en los dos tipos de grados que deseamos dar y que nuestra red neuronal aprenda.

*celsius = np.array([-40, -10, 0, 8, 15, 22, 38], dtype=float)*

*fahrenheit = np.array({-40, 14, 32, 46, 59, 72, 100], dtype=float))*

en este punto estaremos usando Keras, framework que se usa para crear redes neuronales mas simples. Podemos espesificar las dos capas pero con Keras podemos solo especificar la capa de salida.

Creamos una capa de salida de dipo Densa. Densa nos permitirá establecer esa conexión entre las neuronas de nuestra inteligencia

C*apa = tf.kera.layers.Dense(units=1, input\_shape=[1])*

Indicamos la cantidad de neuronas de la capa (capas de salida)

Aquí indicamos la cantidad de neuronas que tenemos de entrada

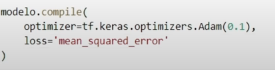
Daremos un tipo de modelo de keras, en este ejemplo se usara el tipo secuencial (pero existen varios tipos mas robustos).

*modelo = tf.keras.Sequential([capal])*

Indicamos la capa que se creo

empezamos a crear la parte donde empezara a aprender nuestra inteligencia de la siguiente manera.

Le permitirá a la red saber como ajustara los pesos de manera eficiente



Permite saber que una gran cantidad de errores grandes es mejor que una cantidad de errores pequeños

Valor de aprendizaje, le permite saber que tanto debe de ajustar los pesos y sesgos

Para empezar el entrenamiento usaremos la función Fit, indicaremos los datos de entrada y salida y le indicaremos la cantidad de vueltas que dará.



Vueltas que dará

Salida

Entrada



Vemos los resultados de la función de perdida, nos indicara que tan mal le fue a la red.



Ahora pongamos a prueba la predicción:



Podemos agregar más capas (neuronas):

