NTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

Nombre de los alumnos: Sandoval Muñoz Emanuel, Valerio López José

Eduardo, Trejo Sierra Héctor.

Matriculas: 2022670104, 2022670002, 2021630065.

Maestr@: CAMACHO VAZQUEZ VANESSA ALEJANDRA

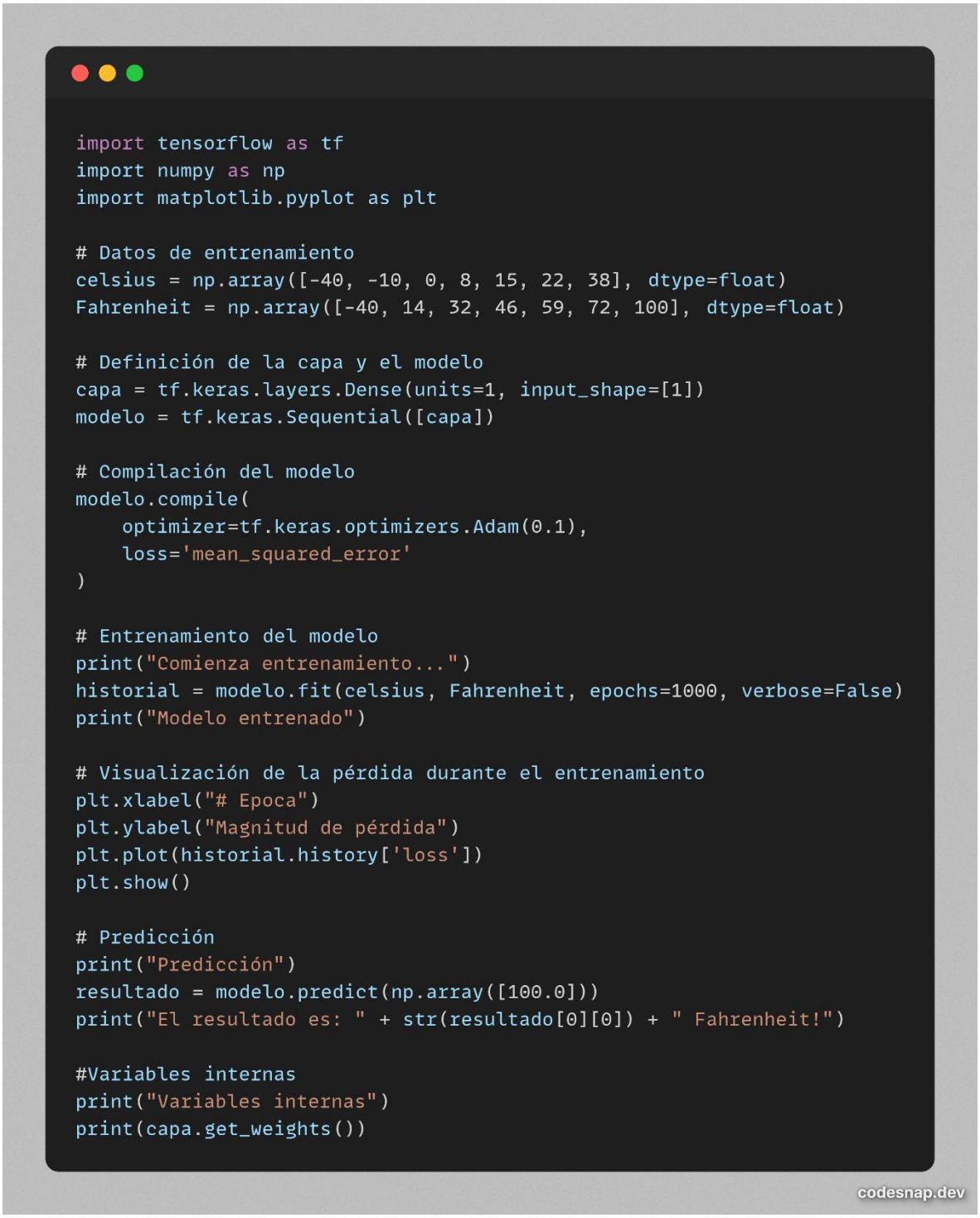
Materia: MACHINE LEARNING

Grupo: 6CV1

Trabajo: Práctica 8. Redes neuronales usando Python y Tensorflow

Código completo de las redes neuronales

Red 1, esta red es la primera con la que se inicia, para poder hacer las otras 2 solicitadas

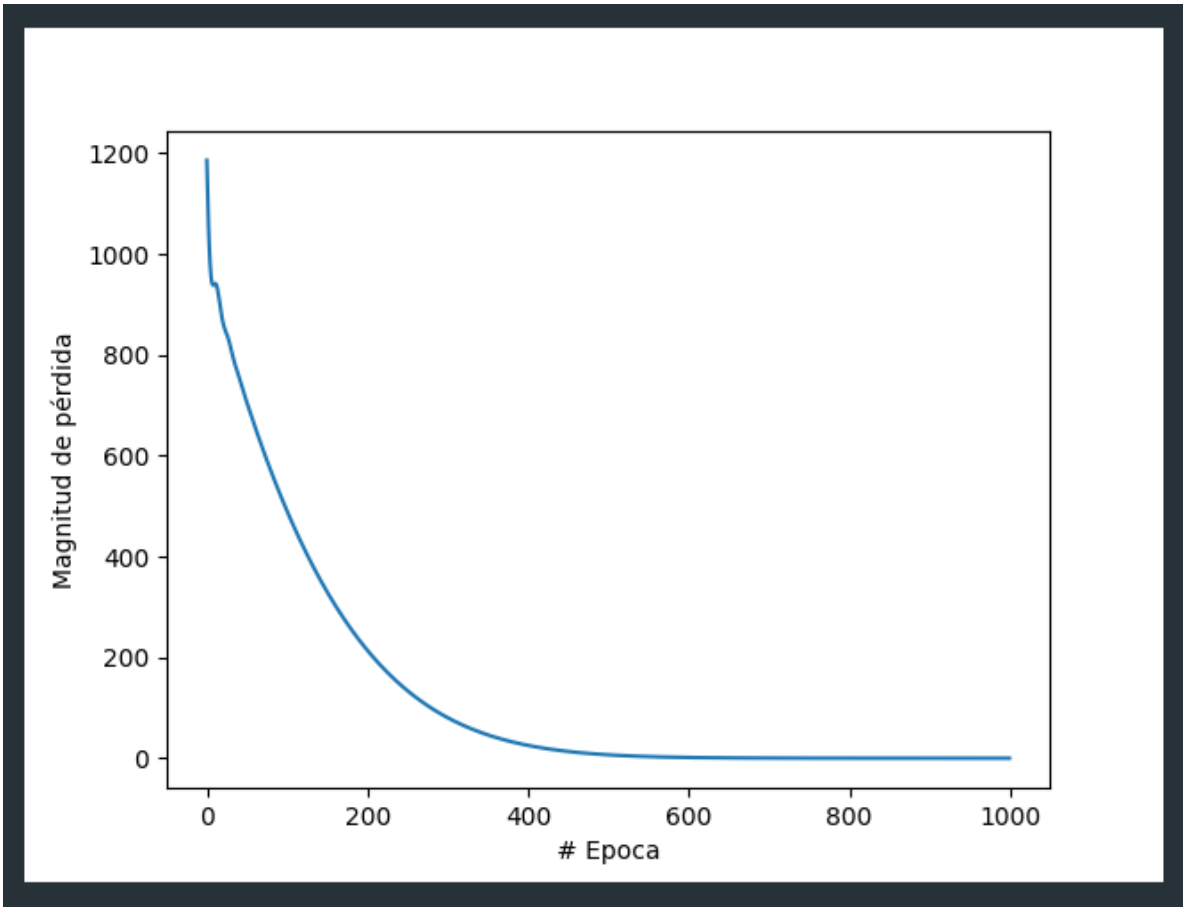


```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# Datos de entrenamiento
celsius = np.array([-40, -10, 0, 8, 15, 22, 38], dtype=float)
Fahrenheit = np.array([-40, 14, 32, 46, 59, 72, 100], dtype=float)

# Definición de la capa y el modelo
capa = tf.keras.layers.Dense(units=1, input_shape=[1])
modelo = tf.keras.Sequential([capa])

# Compilación del modelo
modelo.compile(
    optimizer=tf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)

# Entrenamiento del modelo
print("Comienza entrenamiento...")
historial = modelo.fit(celsius, Fahrenheit, epochs=1000, verbose=False)
print("Modelo entrenado")

# Visualización de la pérdida durante el entrenamiento
plt.xlabel("# Epoca")
plt.ylabel("Magnitud de pérdida")
plt.plot(historial.history['loss'])
plt.show()

# Predicción
print("Predicción")
resultado = modelo.predict(np.array([100.0]))
print("El resultado es: " + str(resultado[0][0]) + " Fahrenheit!")

#Variables internas
print("Variables internas")
print(capa.get_weights())
```

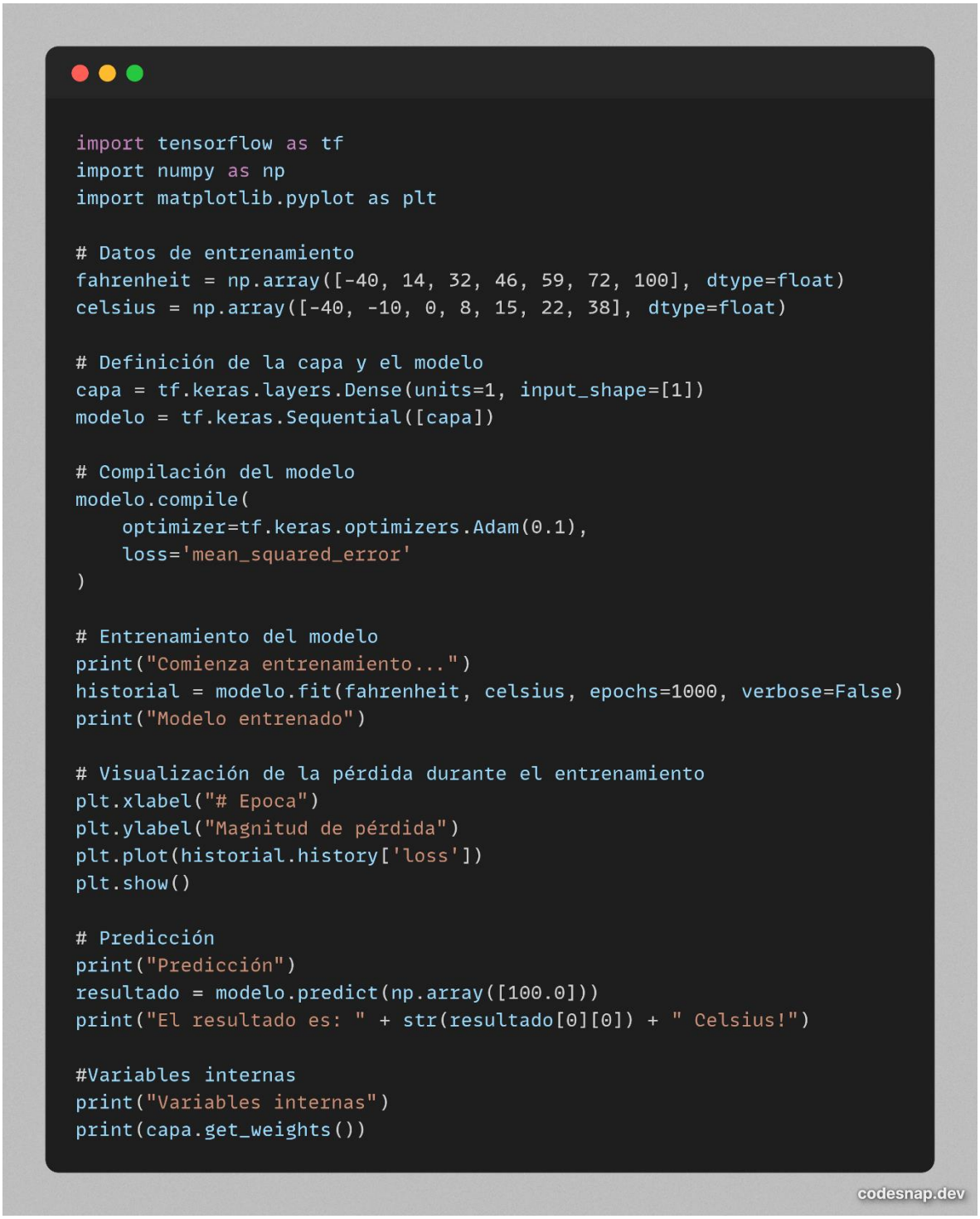
Ejecución

```
Comienza entrenamiento...
Modelo entrenado
Predicción
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
El resultado es: 211.74405 Fahrenheit!
Variables internas
[array([[1.7981333]], dtype=float32), array([31.930733], dtype=float32)]
```

Grafica



Red neuronal 1 convertir de grados Fahrenheit a Celsius

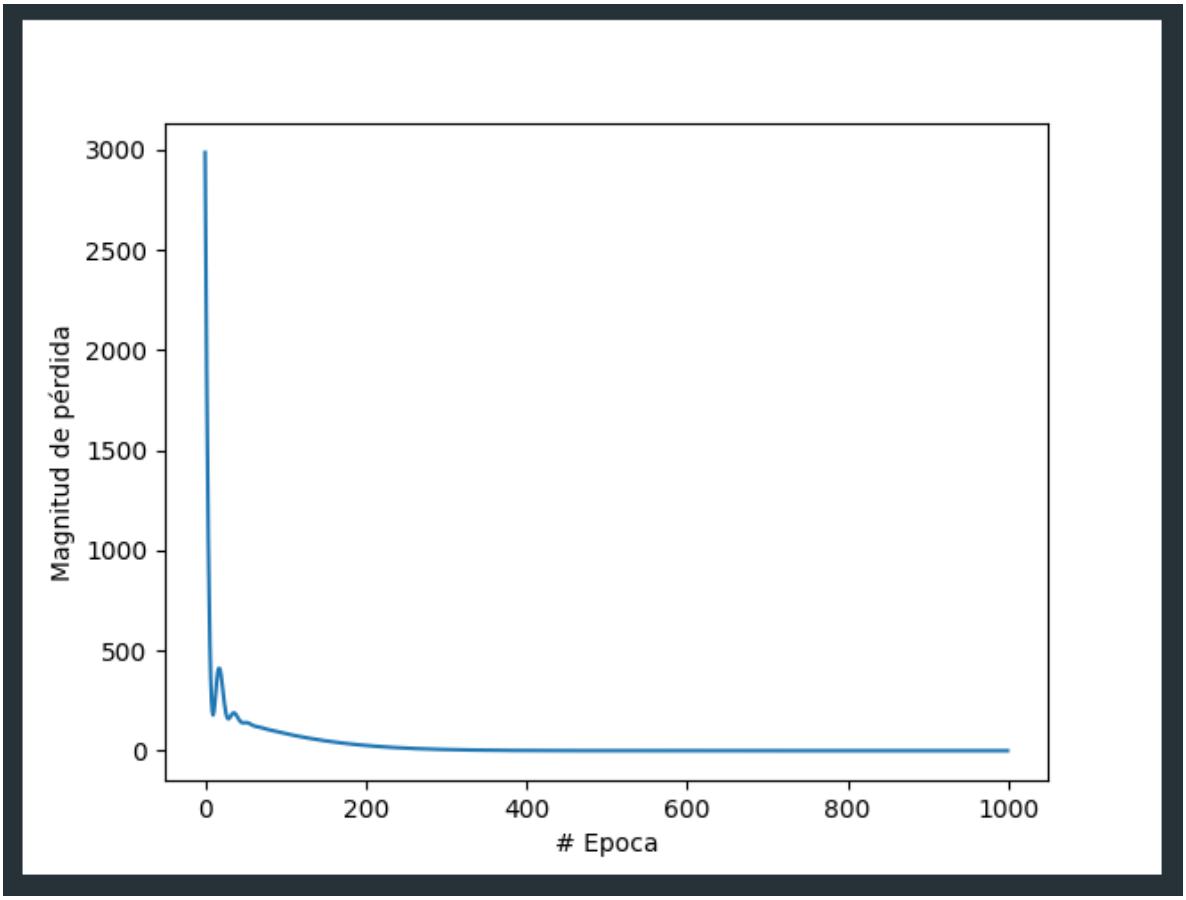


```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# Datos de entrenamiento
fahrenheit = np.array([-40, 14, 32, 46, 59, 72, 100], dtype=float)
celsius = np.array([-40, -10, 0, 8, 15, 22, 38], dtype=float)

# Definición de la capa y el modelo
capa = tf.keras.layers.Dense(units=1, input_shape=[1])
modelo = tf.keras.Sequential([capa])

# Compilación del modelo
modelo.compile(
    optimizer=tf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)

# Entrenamiento del modelo
print("Comienza entrenamiento...")
historial = modelo.fit(fahrenheit, celsius, epochs=1000, verbose=False)
print("Modelo entrenado")

# Visualización de la pérdida durante el entrenamiento
plt.xlabel("# Epoca")
plt.ylabel("Magnitud de pérdida")
plt.plot(historial.history['loss'])
plt.show()

# Predicción
print("Predicción")
resultado = modelo.predict(np.array([100.0]))
print("El resultado es: " + str(resultado[0][0]) + " Celsius!")

#Variables internas
print("Variables internas")
print(capa.get_weights())
```

Ejecución

```
Comienza entrenamiento...
Modelo entrenado
Predicción
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
El resultado es: 37.846046 Celsius!
Variables internas
[array([[0.55616736]], dtype=float32), array([-17.770693], dtype=float32)]
```

Grafica



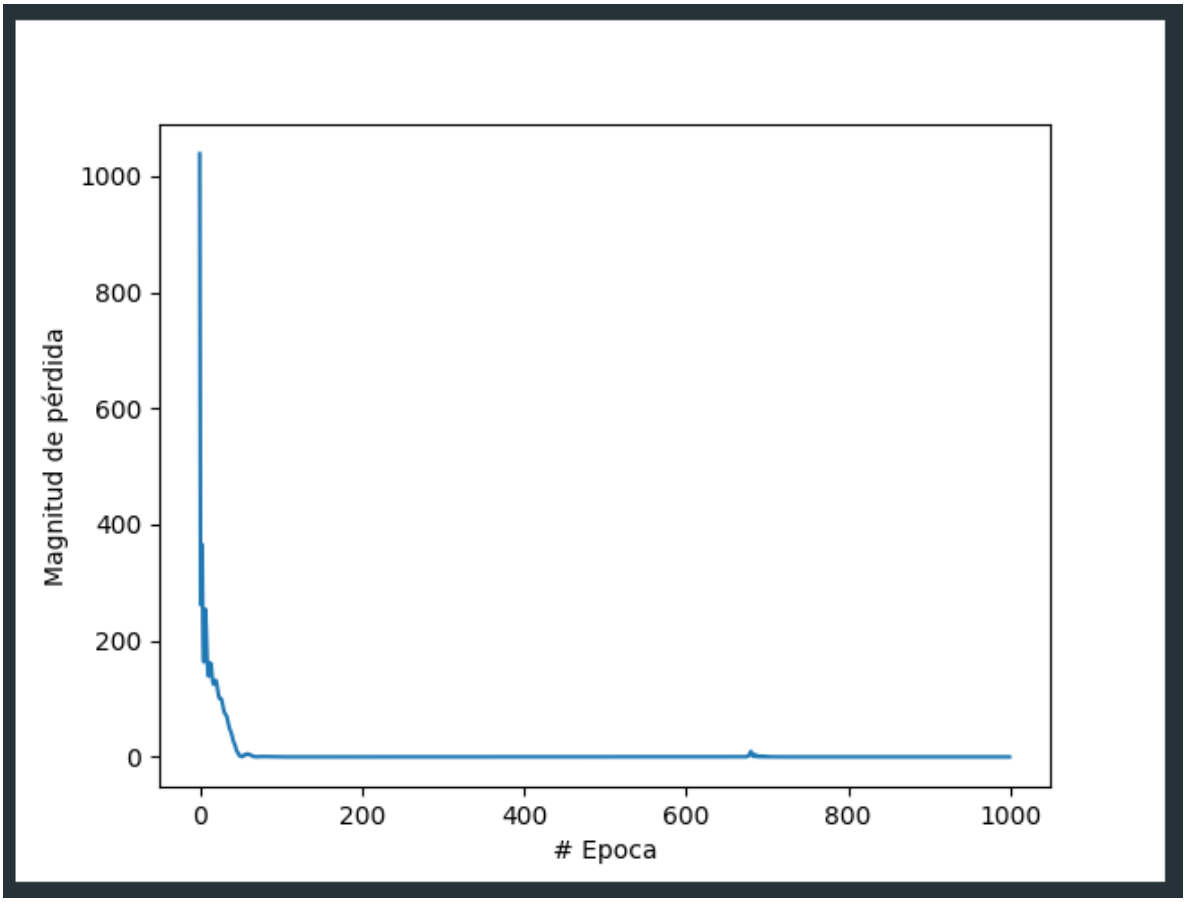
Red neuronal 2

```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# Datos de entrenamiento
fahrenheit = np.array([-40, 14, 32, 46, 59, 72, 100], dtype=float)
celsius = np.array([-40, -10, 0, 8, 15, 22, 38], dtype=float)

# Definición de la capa y el modelo
oculta1 = tf.keras.layers.Dense(units=3, input_shape=[1])
oculta2 = tf.keras.layers.Dense(units=3)
salida = tf.keras.layers.Dense(units=1)
modelo = tf.keras.Sequential([oculta1,oculta2,salida])

# Compilación del modelo
modelo.compile(
    optimizer=tf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)

# Entrenamiento del modelo
print("Comienza entrenamiento...")
historial = modelo.fit(fahrenheit, celsius, epochs=1000, verbose=False)
print("Modelo entrenado")

# Visualización de la pérdida durante el entrenamiento
plt.xlabel("# Epoca")
plt.ylabel("Magnitud de pérdida")
plt.plot(historial.history['loss'])
plt.show()

# Predicción
print("Predicción")
resultado = modelo.predict(np.array([100.0]))
print("El resultado es: " + str(resultado[0][0]) + " Celsius!")

#Variables internas
print("Variables internas")
print(oculta1.get_weights())
print(oculta2.get_weights())
print(salida.get_weights())
```

Ejecución

```
Comienza entrenamiento...
Modelo entrenado
Predicción
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
El resultado es: 37.8461 Celsius!
Variables internas
[array([[-0.0600067 ,  0.45854184, -0.1032307 ]], dtype=float32), array([-3.4929903, -3.3314588,  1.2664479], dtype=float32)]
[array([[-0.58691114,  0.04480637, -0.8916987 ],
       [-0.24449052, -0.59302366, -0.78185195],
       [ 0.83232725,  0.20785694,  0.00677548]], dtype=float32), array([0.86248815, 2.9356275 , 3.3355174 ], dtype=float32)]
[array([[-0.20345353],
       [-0.6522517 ],
       [-1.0792526 ]], dtype=float32), array([-3.743288], dtype=float32)]
```

Grafica



Nombre:

Tema: Aprendizaje Automático vs Programación Regular

Día | Mes | Año | Folio

Estos dos son 2 paradigmas distintos en el desarrollo del software, la programación regular implica escribir explícitamente cada paso que el programa debe seguir para resolver un problema.

En contraste el aprendizaje automático no requiere que el programador defina reglas explícitas. En lugar de esto el programador proporciona un conjunto de datos y el modelo de aprendizaje automático aprende a partir de estos datos, utilizando algoritmos como redes neuronales, árboles de decisión o máquinas de soporte vectorial, el modelo identifica patrones en los datos que le permiten hacer predicciones o tomar decisiones. Este aprendizaje es especialmente útil para problemas donde definir reglas explícitas es complejo o inviable debido a la naturaleza variable y extensa de los datos.

Escenarios que resolvemos,

Supongamos que se está desarrollando un sistema para predecir el éxito académico de los estudiantes en una universidad basado en varios factores, el objetivo es identificar a los estudiantes que podrían necesitar apoyo adicional para mejorar su rendimiento y aumentar sus posibilidades de éxito.

Nombre:

Tema:

Día | Mes | Año

Folio

En un escenario estudial, se resuelve definir una serie de reglas y condiciones basadas en los factores, lo cual podría ser muy complejo y poco preciso, debido a la cantidad de variables y su interacción, sin embargo con el aprendizaje automatico, podemos entrenar un modelo con datos historicos de estudiantes y sus resultados académicos para que el sistema aprenda a predecir el exito academico basado en los factores sin la necesidad de definir reglas explisitas.

Conceptos a conocer,

- Datos de entrenamiento.
- Algoritmos de aprendizaje
- Función de pérdida
- Proceso de entrenamiento
- Validación y prueba
- Red Neuronal

Nombre:

Tema: Proceso que sigue la Red neuronal

Día | Mes | Año | Folio

Recopilación y procesamiento de datos: Recolectar datos relevantes y limpiarlos, normalizar y transformar los datos para que sean adecuados para el modelo.

Definición de la arquitectura de red: Decidir la cantidad de neuronas y capas.

Inicialización de pesos: Asignar valores iniciales aleatorios a los pesos de las conexiones entre neuronas

Forward propagation: Pasar los datos de entrada a través de la red, hasta obtener una salida.

Calculo de la perdida: Comparar el valor de salida con el valor real.

Back ward Propagation: Calcular el gradiente.

Actualización de pesos: Ajustar los pesos utilizados utilizando un algoritmo de optimización.

Iteración del proceso: Repetir los pasos de propagación hacia adelante.

Cambios en la implementación

La pincipal diferencia que se pudo notar a la hora de implementar la red Neurona1 sen alla y la segun de red neuronal más compleja es la capacidad de la red para capturar y aprender patrones complejos en los datos.

la red sencilla en su piper capa (unica) es facil de entender pero limita su capacidad, la segunda con multiples capas ofrece una mayor capacidad de aprendizaje y precisión esto a costa de una mayor carga costo computacional). y tiempo de entrenamiento.

Nombre:

Tema: ¿Como va a aprender?

Día	Mes	Año	Folio

Aprendera a través del proceso, resultado descrito anteriormente, lo podemos resumir en los siguientes pasos,

- Entrenamiento inicial.
- Evaluación de Error
- Repropagación
- Actualización de pesos
- Repetición