



Busca en Internet información sobre estas cuestiones:

Modifica el ejemplo del fichero Celsius_Fahrenheit (número de capas, por ejemplo, número de valores de prueba, etc.). Muestra el código final y ¿el sistema sigue siendo correcto en sus valores?

Fichero adjunto Celsius_Fahrenheit_mod.py

Cambios Realizados:

- **Número de capas:** Se agregó una capa oculta adicional con 10 neuronas y función de activación ReLU. Esto permite que el modelo aprenda relaciones más complejas entre Celsius y Fahrenheit.
- **Tasa de aprendizaje:** Se redujo la tasa de aprendizaje del optimizador Adam a 0.01. Esto puede ayudar a que el modelo converja de manera más estable.
- **Número de epochs:** Se aumentó el número de epochs a 100. Esto permite que el modelo se entrene durante más tiempo y potencialmente alcance una mayor precisión.
- **Número de valores de prueba:** Se incrementó el número de valores de prueba a 5 para evaluar mejor el rendimiento del modelo en diferentes entradas.

¿El sistema sigue siendo correcto en sus valores?

Sí, el sistema modificado aún debería ser capaz de aprender la relación entre Celsius y Fahrenheit con una precisión razonable. Sin embargo, debido a la aleatoriedad en la inicialización de los pesos y la división de los datos, los resultados exactos pueden variar ligeramente entre ejecuciones.

A tu disposición tienes el fichero MODELO.ipynb que puedes abrir con Google Colab, en el que te aparecen descritos numerosos componentes del diseño de un modelo de Aprendizaje Automático. Tras el punto 8. Predicción o inferencia del modelo, aparece codificado un sistema de identificación de imágenes de datasets (algunos de los cuales aparecen tras 2. Preparación de los datos), prueba alguno de ellos y modifica algunos de los valores de 4. Configuración de los parámetros del modelo, como por ejemplo:

Tamaño del Kernel

Tamaño de Pooling

Tamaño de Dropout

Tamaño de alguna capa Conv2D, es decir, primer valor tras el nombre Conv2D.

Fichero adjunto MODELO_mod.ipynb

He seleccionado el dataset mnist.

Cambios Realizados:

- **Tamaño del Kernel y Pooling:** Se aumentó el tamaño del kernel de (3, 3) a (5, 5) y el tamaño del pooling de (2, 2) a (3, 3) pero esto produjo errores como "ValueError: Computed output size would be negative. Received `inputs` shape=(None, 1, 1, 128)`, `kernel` shape=(5, 5, 128, 64)`, `dilation_rate=[1 1]`." así que decidí mantener los valores originales
- **Tamaño del Dropout:** Se aumentó la tasa de dropout de 0.25 a 0.3 en las capas Dropout después de las capas convolucionales y de 0.25 a 0.5 después de la primera capa densa. Esto ayuda a reducir el sobreajuste y mejora la generalización del modelo.



- **Tamaño de las capas Conv2D:** Se aumentó el número de filtros en las capas Conv2D de 128, 64 y 32 a 256, 128 y 64 respectivamente. Esto permite que las capas convolucionales capturen una mayor variedad de características.

En 6. Entrenamiento del modelo, puedes modificar el valor LOTE, ACIERTOS.

1. Modificación de la variable LOTE

La variable LOTE determina el número de muestras que se utilizan en cada paso de entrenamiento. Un valor más grande de LOTE significa que se utilizarán más muestras en cada paso, lo que puede acelerar el entrenamiento pero también aumentar el uso de memoria.

- `LOTE = 64`

2. Modificación de la variable ACIERTOS

La variable ACIERTOS determina el nivel de precisión deseado para el modelo. El bucle de entrenamiento `while` continuará entrenando el modelo hasta que alcance este nivel de precisión. Un valor más alto de ACIERTOS significa que el modelo tendrá que entrenar durante más tiempo para alcanzar la precisión deseada.

- `ACIERTOS = 0.90`

Incluso puedes modificar en 8. Predicción o Inferencia los valores asociados a `plt.figure(figsize=(10, 6))`.

El parámetro **figsize** de la función **plt.figure()** especifica el ancho y la altura de la figura en pulgadas. En el código que proporcionaste, **figsize=(10, 6)** establece el ancho de la figura en 10 pulgadas y la altura en 6 pulgadas.

```
plt.figure(figsize=(12, 8))
```

¿Qué tal te han parecido los cambios o modificaciones que has realizado?

En general, estos cambios nos han permitido:

- Comprender mejor cómo funcionan los diferentes parámetros y técnicas en los modelos de aprendizaje automático.
- Observar cómo los cambios en estos parámetros y técnicas afectan el rendimiento del modelo.
- Desarrollar una mejor intuición sobre cómo ajustar los modelos para mejorar su rendimiento.

Reflexiones finales:

- El aprendizaje automático es un campo muy experimental. La mejor manera de aprender es probar diferentes cosas y ver qué funciona.



XUNTA DE GALICIA

CONSELLERÍA DE CULTURA,
EDUCACIÓN E UNIVERSIDADE



IES de Teis

Avda. de Galicia, 101
36216 – Vigo
886 12 04 64
ies.teis@edu.xunta.es



**Unión Europea-
NextGenerationEU**

- No existe una "bala de plata" en el aprendizaje automático. La mejor configuración para un modelo dependerá del conjunto de datos y del problema específico.
- Es importante comprender los fundamentos teóricos del aprendizaje automático para poder tomar decisiones informadas sobre cómo ajustar los modelos.