CLASIFICADOR CON AUMENTO DE DATOS (DATA AUGMENTATION) UTILIZANDO UNA RED CONVOLUCIONAL EN EL DATASET FASHION MNIST

Objetivo del ejercicio

Aprender cómo mejorar la generalización de un modelo convolucional aplicando **técnicas de aumento de datos** mediante ImageDataGenerator.

¿Qué es ImageDataGenerator?

ImageDataGenerator es una clase de Keras (en TensorFlow) que se usa para **preprocesar imágenes** antes de entrenar una red neuronal. Su principal utilidad es permitir el **aumento de datos (data augmentation)**, lo que ayuda a que tu modelo **aprenda mejor y generalice más**.

Paso 1: Cargar y preparar los datos

```
from tensorflow.keras.datasets import fashion_mnist

(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

¿Qué es Fashion MNIST?
Es un dataset de imágenes en escala de grises de 28x28 píxeles de ropa: camisetas, pantalones, zapatos, etc. Tiene 10 clases y es similar a MNIST, pero más complejo.

Paso 2: Normalizar y agregar canal

```
x_train = x_train.astype('float32') / 255

x_test = x_test.astype('float32') / 255


x_train = x_train.reshape(-1, 28, 28, 1)

x_test = x_test.reshape(-1, 28, 28, 1)
```

¿Por qué hacemos esto?

- Normalizamos los valores de píxeles para que estén entre 0 y 1 (esto ayuda al entrenamiento).
- Añadimos un canal (1) para que TensorFlow entienda que son imágenes de un solo canal (grises).

Paso 3: Codificar etiquetas con to_categorical

```
from tensorflow.keras.utils import to_categorical


y_train_cat = to_categorical(y_train, 10)

y_test_cat = to_categorical(y_test, 10)
```

¿Qué hace esto?
Convierte las etiquetas (por ejemplo 3) a un vector como [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]. Es necesario para clasificación multicategoría con softmax.

Paso 4: Crear el generador con aumento de datos

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator


datagen = ImageDataGenerator(
  rotation_range=15,
  width_shift_range=0.1,
  height_shift_range=0.1,
  zoom_range=0.1
)
datagen.fit(x_train)
```

¿Qué hace ImageDataGenerator?
Crea versiones **aumentadas** (ligeramente modificadas) de las imágenes para simular variabilidad real. Esto **reduce el sobreajuste**.

- rotation_range=15: Rota la imagen aleatoriamente hasta ±15 grados.
- width_shift_range=0.1: Desplaza horizontalmente hasta el 10% del ancho.
- height_shift_range=0.1: Igual pero vertical.
- zoom_range=0.1: Hace zoom aleatorio hasta un 10%.

Paso 5: Visualizar imágenes aumentadas

```
for x_batch, y_batch in datagen.flow(x_train, y_train_cat, batch_size=9):
  # Muestra 9 imágenes aumentadas
  ...
  break
```

¿Por qué lo visualizamos?
Para comprobar que el generador está funcionando y las imágenes se ven razonables (no deformadas o mal etiquetadas).

Paso 6: Crear modelo CNN simple

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense


model = Sequential([
```

```
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])
```

¿Qué hace esta red?
Es una red convolucional **básica pero funcional**:

1. Capa convolucional (extrae características)
2. Max Pooling (reduce dimensiones)
3. Flatten (aplasta los datos para Dense)
4. Capa densa oculta
5. Capa de salida softmax (10 clases)

Paso 7: Compilar y entrenar el modelo usando el generador

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(datagen.flow(x_train, y_train_cat, batch_size=32),
          epochs=5,
          validation_data=(x_test, y_test_cat))
```

¿Qué hace fit(datagen.flow(...))?
Entrena el modelo **alimentándolo con las imágenes aumentadas** generadas en tiempo real.

Paso 8: Evaluar en los datos de prueba

```
test_loss, test_acc = model.evaluate(x_test, y_test_cat)
print(f"Precisión en test: {test_acc:.4f}")
```

¿Qué obtenemos?
Una métrica de rendimiento (precisión final) usando **datos reales de prueba no aumentados**.
Compara este resultado con el mismo modelo sin aumento.