In [10]:

```
import tensorflow as tf
import numpy as np
import pandas as pd
from tensorflow.keras.datasets.fashion_mnist import load_data
```

In [11]:

```
# cargar datos
(X_train, y_train), (X_test, y_test) = load_data()
```

In [12]:

```
# procesamiento de datos
X_train = X_train.reshape((-1, 28, 28, 1))
X_train = X_train/255
X_test = X_test.reshape((-1, 28, 28, 1))
X_test = X_test/255
```

In [13]:

```
# parámetros de capacitación
batch_size = 8
n_epochs   = 20
learn_rate = 0.0001
```

In [14]:

```
# modelo
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

In [15]:

```
# Compilar el modelo
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=learn_rate),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
```

In [16]:

```
# Entrenar modelo
model.fit(X_train, y_train, batch_size=batch_size,
        epochs=n_epochs)
```

```
Epoch 1/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 28s 4ms/step - accuracy: 0.7269 - loss: 0.7677
Epoch 2/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 28s 4ms/step - accuracy: 0.8593 - loss: 0.3924
Epoch 3/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.8816 - loss: 0.3277
Epoch 4/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 28s 4ms/step - accuracy: 0.8903 - loss: 0.3002
Epoch 5/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.8997 - loss: 0.2725
Epoch 6/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 26s 3ms/step - accuracy: 0.9080 - loss: 0.2550
Epoch 7/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 26s 3ms/step - accuracy: 0.9152 - loss: 0.2367
Epoch 8/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.9179 - loss: 0.2256
Epoch 9/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.9203 - loss: 0.2179
Epoch 10/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.9241 - loss: 0.2062
Epoch 11/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.9290 - loss: 0.1981
Epoch 12/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.9312 - loss: 0.1837
Epoch 13/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 26s 4ms/step - accuracy: 0.9368 - loss: 0.1712
Epoch 14/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 28s 4ms/step - accuracy: 0.9368 - loss: 0.1731
Epoch 15/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 29s 4ms/step - accuracy: 0.9399 - loss: 0.1630
Epoch 16/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 27s 4ms/step - accuracy: 0.9446 - loss: 0.1545
Epoch 17/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 26s 4ms/step - accuracy: 0.9466 - loss: 0.1469
Epoch 18/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 26s 4ms/step - accuracy: 0.9479 - loss: 0.1421
Epoch 19/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 25s 3ms/step - accuracy: 0.9499 - loss: 0.1362
Epoch 20/20
7500/7500 ━━━━━━━━━━━━━━━━━━━━ 25s 3ms/step - accuracy: 0.9514 - loss: 0.1296
```

```
Out[16]: <keras.src.callbacks.history.History at 0x176a005cb60>
```

```
In [17]: # Evaluar en test
         test_loss, test_acc = model.evaluate(X_test, y_test)
         print(f"\nPrecisión en datos de prueba: {test_acc:.4f}")
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9220 - loss: 0.2457

Precisión en datos de prueba: 0.9225
```