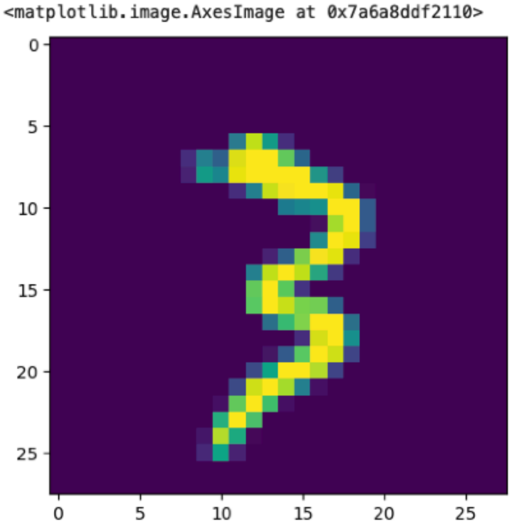
1. **Ejecuta la predicción de una muestra y observa que el resultado es una lista de**

```
Image:
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 108 232 137  33   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  33 111  77 241 254 254 195  87   2   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  27 140 117 245 254 254 254 254 133  48   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0  33 112 233 251 254 254 245  83   6   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0 106 117 129 254 254  75   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  10 221 254  75   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 122 254 246  50   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0  24  82 203 252 243  36   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 111 230 254 232 153  43   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 195 254 192  41   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 191 254 234 205 199  78   1   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0  93 175 208 254 254  93   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  37 239 254 129   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0  14  73 190 254 235  55   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  57 150 254 254 227  55   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0  61 237 254 221 112  12   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0  12 195 254 180  12   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0 160 254 183   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  18 225 158   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  31 148  25   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]]
<matplotlib.image.AxesImage at 0x7a6a8ddf2110>
```



números.

2. ¿Por qué crees que es así y qué representan esos números?

Es la representación de características que definen una forma definida de un carácter en un array que representa el area de 28x28 pixeles.

3. Experimenta con diferentes valores para el número de neuronas en la capa oculta de la red neuronal

10:

```
Perdida en el conjunto de prueba: 0.18608200550079346
```

Precisión en el conjunto de prueba: 0.9451833367347717

90:
Perdida en el conjunto de prueba: 0.015796812251210213
Precisión en el conjunto de prueba: 0.9957333207130432

240:
Perdida en el conjunto de prueba: 0.007639619521796703
Precisión en el conjunto de prueba: 0.9976333379745483

128, 40:
Perdida en el conjunto de prueba: 0.013722683303058147
Precisión en el conjunto de prueba: 0.9956666827201843

128,80:
Perdida en el conjunto de prueba: 0.011161230504512787
Precisión en el conjunto de prueba: 0.9972000122070312

4. Cambia la función de activación de la capa oculta a "sigmoid" o "tanh" y observa cómo afecta el rendimiento del modelo.

Sigmoid

Perdida en el conjunto de prueba: 0.030255744233727455
Precisión en el conjunto de prueba: 0.9926833510398865

Tanh

Perdida en el conjunto de prueba: 0.011531858704984188
Precisión en el conjunto de prueba: 0.9976999759674072

5. ¿Qué sucedería si eliminas la capa Flatten()?

ValueError: `labels.shape` must equal `logits.shape` except for the last dimension. Received: labels.shape=(32,) and logits.shape=(896, 10)

No se aplana la entrada al vector necesario

7. Prueba diferentes optimizadores, como "sgd" o "rmsprop", y observa cómo afectan el rendimiento del modelo.

Sgd

```
Perdida en el conjunto de prueba: 0.15661783516407013
Precisión en el conjunto de prueba: 0.9562000036239624
```

rmsprop

```
Perdida en el conjunto de prueba: 0.020149284973740578
Precisión en el conjunto de prueba: 0.9939833283424377
```

8. Aumenta el número de épocas de entrenamiento y observa cómo afecta el rendimiento del modelo.

40:

```
Perdida en el conjunto de prueba: 0.006380090024322271
Precisión en el conjunto de prueba: 0.9976166486740112
```

9. Modifica la arquitectura de la red neuronal agregando capas adicionales o cambiando el número de neuronas en cada capa, y observa cómo afecta el rendimiento del modelo.

```
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

```
Perdida en el conjunto de prueba: 0.0344943068921566
Precisión en el conjunto de prueba: 0.9902999997138977
```

10. Elimina la normalización de los valores de píxeles y observa cómo afecta el rendimiento del modelo.

```
Perdida en el conjunto de prueba: 0.14308802783489227
Precisión en el conjunto de prueba: 0.9633166790008545
```

11. Considera las capas finales (de salida). ¿Por qué hay 10 de ellas? ¿Qué pasaría si tuvieras una cantidad diferente a 10?

```
Perdida en el conjunto de prueba: 0.01710028201341629
Precisión en el conjunto de prueba: 0.9945833086967468
```