```
import numpy as np
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Conv2D,MaxPooling2D,Flatten,Dropout
import matplotlib.pyplot as plt
```

```
print("[INFO] accessing MNIST...")
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape((x_train.shape[0], 28, 28, 1)).astype('float32') / 255
x_test = x_test.reshape((x_test.shape[0], 28, 28, 1)).astype('float32') / 255
```

```
[INFO] accessing MNIST...
```

```
model = Sequential()

# Convolutional layers
model.add(Conv2D(28,kernel_size=(3, 3), input_shape=(28, 28, 1)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())

# Fully connected layers
model.add(Dense(200,activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(10,activation="softmax"))
model.summary()
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 28)	280
max_pooling2d (MaxPooling2D)	(None, 13, 13, 28)	0
flatten (Flatten)	(None, 4732)	0
dense_3 (Dense)	(None, 200)	946,600
dropout (Dropout)	(None, 200)	0
dense_4 (Dense)	(None, 10)	2,010

```
 Total params: 948,890 (3.62 MB)
 Trainable params: 948,890 (3.62 MB)
 Non-trainable params: 0 (0.00 B)
```

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

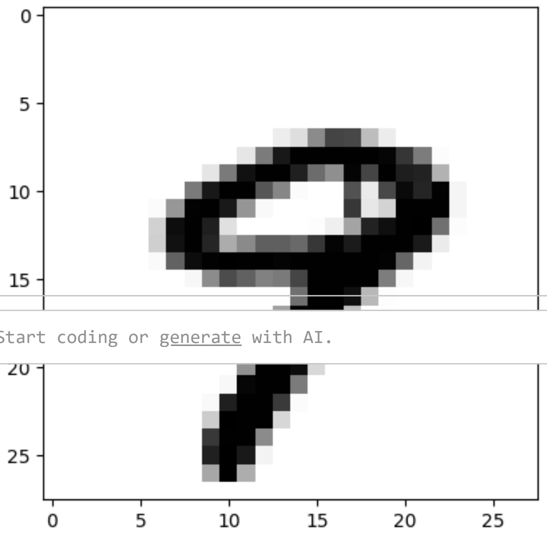
```
model.fit(x_train, y_train, epochs=2)
```

```
Epoch 1/2
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 48s 25ms/step - accuracy: 0.8949 - loss: 0.3462
Epoch 2/2
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 45s 24ms/step - accuracy: 0.9715 - loss: 0.0916
<keras.src.callbacks.history.History at 0x7ef3103ac8f0>
```

```
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_accuracy*100:.2f}%')
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 11ms/step - accuracy: 0.9755 - loss: 0.0760
Test accuracy: 98.02%
```

```
image = x_test[9]
plt.imshow(image,cmap='Greys')
plt.show()
```



Start coding or generate with AI.