

Double-click (or enter) to edit

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

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images = train_images / 255.0
test_images = test_images / 255.0

train_images = train_images.reshape(-1, 28, 28, 1)
test_images = test_images.reshape(-1, 28, 28, 1)

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

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

history = model.fit(train_images, train_labels,
                    epochs=5,
                    batch_size=64,
                    validation_split=0.2)

test_loss, test_acc = model.evaluate(test_images, test_labels)
print(f"\nTest accuracy: {test_acc:.4f}")
print(f"Test loss: {test_loss:.4f}")

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy', marker='o')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy', marker='o')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss', marker='o')
plt.plot(history.history['val_loss'], label='Validation Loss', marker='o')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()

predictions = model.predict(test_images)
predicted_labels = np.argmax(predictions, axis=1)

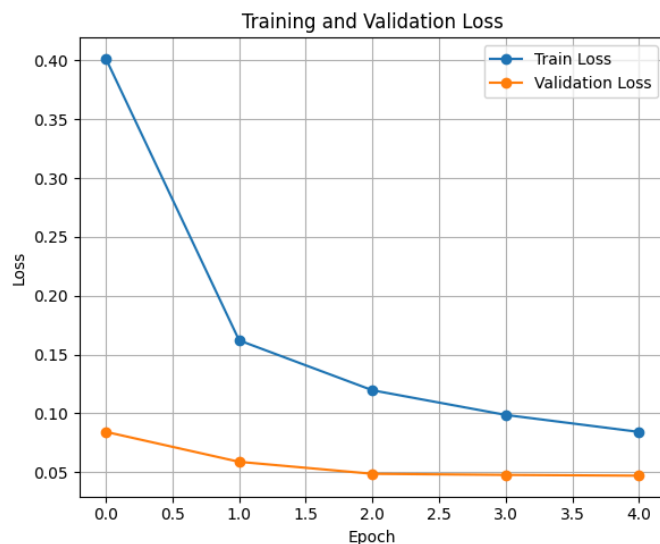
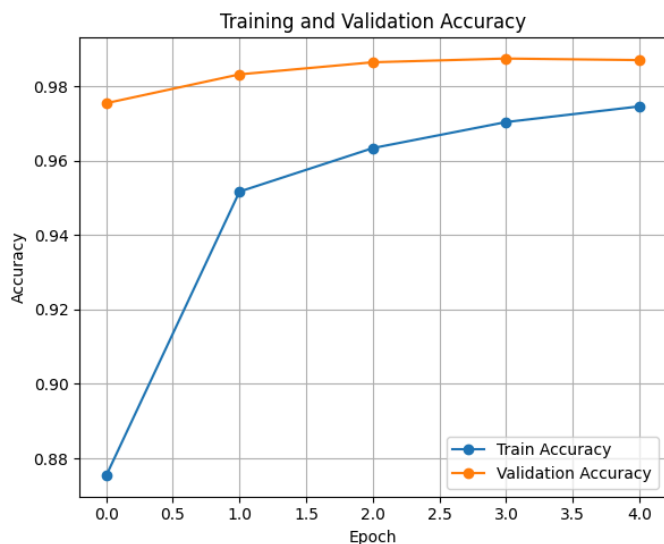
num_samples = 10
plt.figure(figsize=(15, 4))
for i in range(num_samples):
    plt.subplot(1, num_samples, i + 1)
    plt.imshow(test_images[i].reshape(28, 28), cmap='gray')
    plt.title(f"Pred: {predicted_labels[i]}\nTrue: {test_labels[i]}")
    plt.axis('off')

plt.suptitle("Sample Predictions on Test Images", fontsize=16)
plt.show()
```

```
Epoch 1/5  
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_shape`/  
super().__init__(activity_regularizer=activity_regularizer, **kwargs)  
750/750 — 25s 31ms/step - accuracy: 0.7633 - loss: 0.7330 - val_accuracy: 0.9754 - val_loss: 0.0843  
Epoch 2/5  
750/750 — 40s 30ms/step - accuracy: 0.9479 - loss: 0.1751 - val_accuracy: 0.9832 - val_loss: 0.0588  
Epoch 3/5  
750/750 — 23s 30ms/step - accuracy: 0.9610 - loss: 0.1246 - val_accuracy: 0.9864 - val_loss: 0.0486  
Epoch 4/5  
750/750 — 41s 31ms/step - accuracy: 0.9688 - loss: 0.1004 - val_accuracy: 0.9874 - val_loss: 0.0477  
Epoch 5/5  
750/750 — 41s 31ms/step - accuracy: 0.9734 - loss: 0.0871 - val_accuracy: 0.9870 - val_loss: 0.0470  
313/313 — 2s 5ms/step - accuracy: 0.9849 - loss: 0.0470
```

Test accuracy: 0.9888

Test loss: 0.0391



WARNING:tensorflow:5 out of the last 5 calls to <function TensorFlowTrainer.make\_predict\_function.<locals>.one\_step\_on\_data\_distrib  
313/313 — 2s 5ms/step

### Sample Predictions on Test Images

