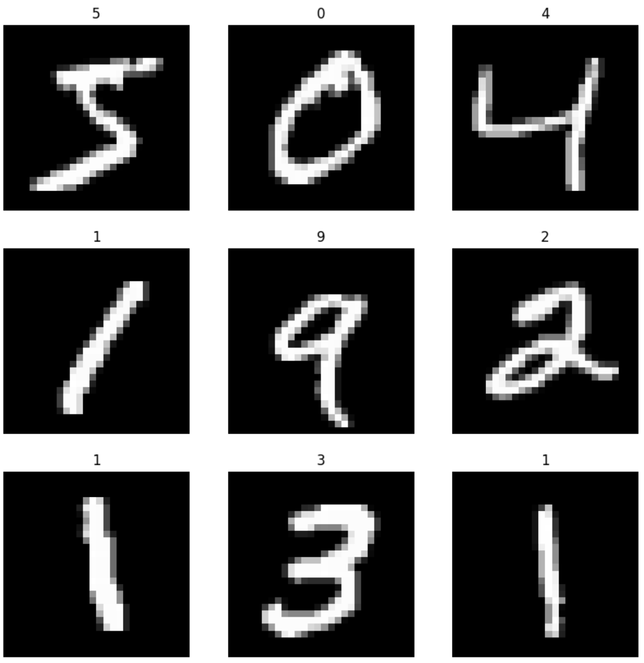
Open in Colab

In [1]:
```
import tensorflow as tf
import matplotlib.pyplot as plt
# Load the MNIST dataset
mnist = tf.keras.datasets.mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images.shape
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

Out[1]:
```
(60000, 28, 28)
```

In [5]:
```
plt.figure(figsize=(10, 10))
for i in range(9):
 ax = plt.subplot(3, 3, i + 1)
 plt.imshow(train_images[i], cmap='gray')
 plt.title(train_labels[i])
 plt.axis('off')
```



In [6]:
```
import tensorflow as tf
model = tf.keras.Sequential([
 tf.keras.layers.Flatten(input_shape=(28, 28)),
 tf.keras.layers.Dense(128, activation='relu'),
 tf.keras.layers.Dense(10, activation='softmax')
])
```

In [9]:
```
model.compile(tf.keras.optimizers.Adam(learning_rate=0.001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

In [10]:
```
model.fit(train_images, train_labels, epochs=3)
```

```
Epoch 1/3
1875/1875 [==============================] - 6s 3ms/step - loss: 2.2096 - accuracy: 0.8477
Epoch 2/3
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3872 - accuracy: 0.9078
Epoch 3/3
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2854 - accuracy: 0.9308
```

Out[10]:
```
<keras.callbacks.History at 0x7e007ccd2b60>
```

In [11]:
```
train_accuracy=model.evaluate(train_images , train_labels)
```

```
1875/1875 [==============================] - 3s 1ms/step - loss: 0.2513 - accuracy: 0.9342
```

In [12]:
```
test_accuracy=model.evaluate(test_images , test_labels)
```

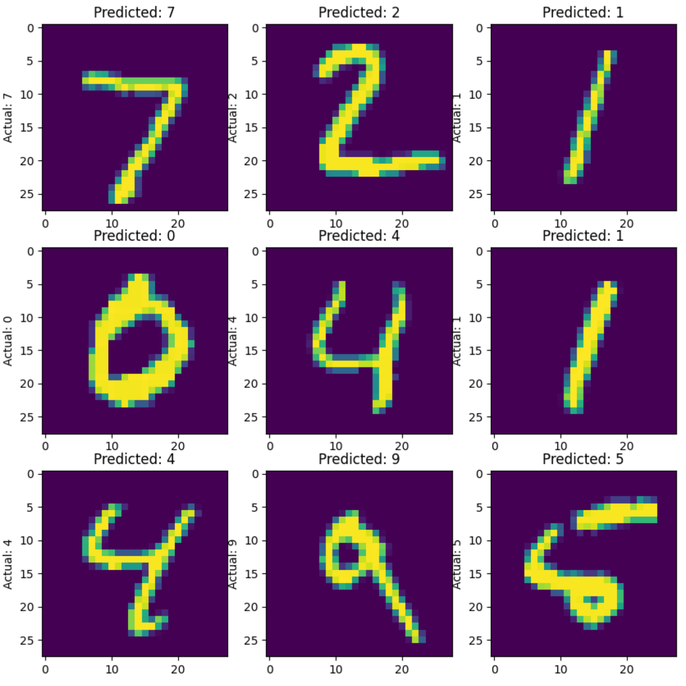
```
313/313 [==============================] - 0s 1ms/step - loss: 0.3347 - accuracy: 0.9234
```

In [13]:
```
print("train_accuracy",train_accuracy)
print("test_accuracy",test_accuracy)
```

```
train_accuracy [0.2513413429260254, 0.934166669845581]
test_accuracy [0.334721177816391, 0.9233999848365784]
```

In [15]:
```
import numpy as np
test_images_9 = test_images[:9]
fig, axes = plt.subplots(3, 3, figsize=(10, 10))
for i in range(9):
 test_images_9_1 = np.reshape(test_images_9[i], (1, 28, 28))
 prediction = model.predict(test_images_9_1)
 axes[i // 3, i % 3] = plt.subplot(3, 3, i + 1)
 axes[i // 3, i % 3].imshow(test_images_9[i])
 axes[i // 3, i % 3].set_title(f"Predicted: {prediction.argmax()}")
 axes[i // 3, i % 3].set_ylabel(f"Actual: {test_labels[i]}")
plt.show()
```

```
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 22ms/step
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: