Handwritten Digit Recognition using TensorFlow

ISHAN AHMAD

October 13, 2023

1 Introduction

In this project, we will walk through the process of building a handwritten digit recognition model using the MNIST dataset and TensorFlow. We will load and preprocess the data, create a neural network, train the model, and evaluate its performance.

2 Data Loading and Preprocessing

We start by loading the MNIST dataset using TensorFlow's Keras API. The dataset contains 60,000 training images and 10,000 test images.

```
(X_train, y_train), (X_test, y_test) = keras.datasets.mnist.load_data()
```

Next, we normalize the pixel values of the images to the range [0, 1] by dividing by 255.

```
X_train = X_train / 255
X_test = X_test / 255
```

We also flatten the images from 28x28 to 784 pixels.

```
X_train_flattened = X_train.reshape(len(X_train), 28*28)
X_test_flattened = X_test.reshape(len(X_test), 28*28)
```

3 Neural Network Model

We create a simple feedforward neural network with two dense layers. The first layer has 150 neurons with ReLU activation, and the output layer has 10 neurons with a softmax activation.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(150, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

We compile the model with the Adam optimizer and sparse categorical cross-entropy loss.

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

4 Model Training

We train the model on the training data for 40 epochs.

```
model.fit(X_train, y_train, epochs=40)
```

5 Model Evaluation

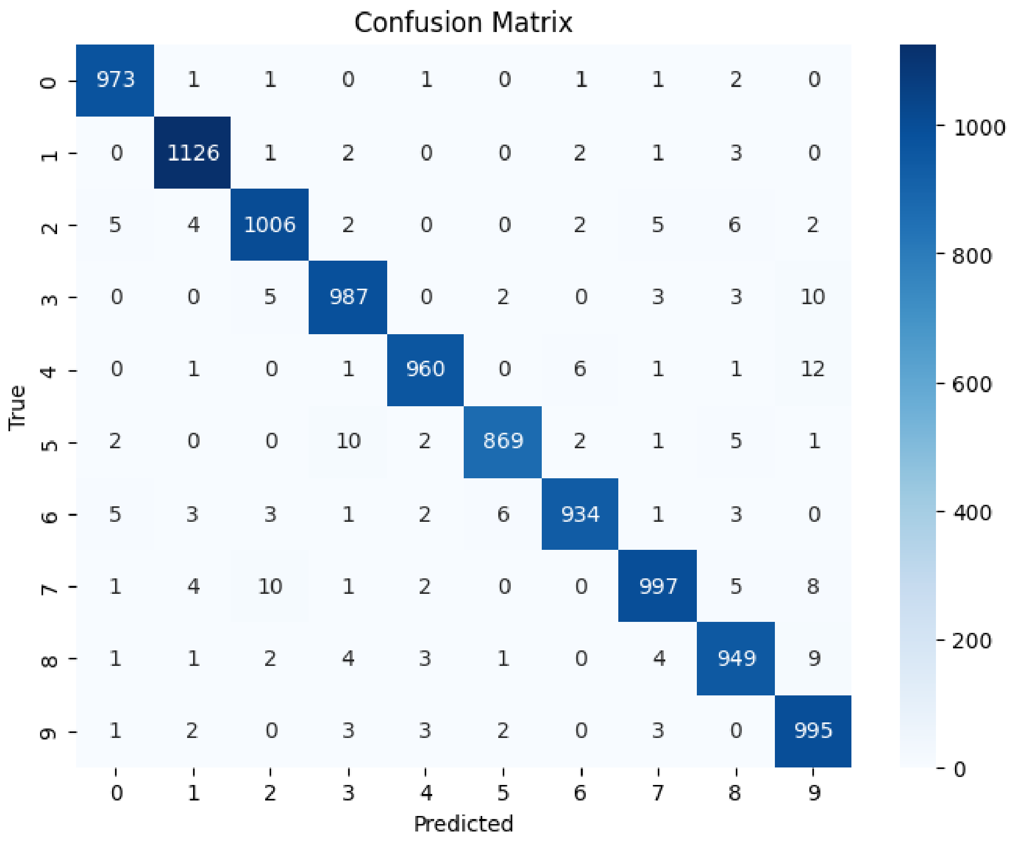
We evaluate the model on the test data.

```
model.evaluate(X_test, y_test)
```

6 Confusion Matrix

We create a confusion matrix to visualize the model's performance.

```
y_pred = model.predict(X_test)
y_pred_labels = np.argmax(y_pred, axis=1)
conf_matrix = confusion_matrix(y_test, y_pred_labels)
```



True \ Predicted	0	1	2	3	4	5	6	7	8	9
0	973	1	1	0	1	0	1	1	2	0
1	0	1126	1	2	0	0	2	1	3	0
2	5	4	1006	2	0	0	2	5	6	2
3	0	0	5	987	0	2	0	3	3	10
4	0	1	0	1	960	0	6	1	1	12
5	2	0	0	10	2	869	2	1	5	1
6	5	3	3	1	2	6	934	1	3	0
7	1	4	10	1	2	0	0	997	5	8
8	1	1	2	4	3	1	0	4	949	9
9	1	2	0	3	3	2	0	3	0	995

Figure 1: Confusion Matrix

7 Conclusion

In this project, we successfully built and trained a neural network model for handwritten digit recognition. The model achieved an accuracy of 98% on the MNIST dataset.