Ex. No. 1	Multiclass Classification
Date of Exercise	15/07/2025

Aim:

To build and train a fully connected neural network for multiclass classification using the MNIST dataset.

Description:

The MNIST dataset consists of 70,000 grayscale images of handwritten digits (0-9). A neural network will be built using TensorFlow/Keras to classify these digits.

Classification is a type of supervised learning where the goal is to predict the category (or class) of an input based on the features of that input. The model is trained on labeled data, where the input data is associated with a specific class label. Once trained, the model can predict the class of new, unseen input data. Types of Classification:

1. **Binary Classification**: The task involves predicting one of two possible classes.
   - Example: Predicting whether an email is "spam" or "not spam."
2. **Multiclass Classification**: The task involves predicting one out of multiple possible classes.
   - Example: Classifying an image of an animal into one of the categories like "cat," "dog," "bird," etc.

Multiclass classification is a type of classification where the model is tasked with assigning an input to one of three or more classes. In this case, there are more than two classes, and each input belongs to exactly one of those classes.

More than two classes: The model predicts one class from a set of multiple possible classes (typically more than two).
Example: A model that classifies handwritten digits from the MNIST dataset, where the classes are the digits **0-9**, resulting in 10 possible classes.

Key Concepts in Multiclass Classification:

- **One-Hot Encoding**: For multiclass classification, the output is often represented using one-hot encoding, where each class is represented by a binary vector (with 1 indicating the class and 0 for others).
  - For example, if the classes are ["cat", "dog", "bird"], the one-hot encoded labels would be: ■ "cat" → [1, 0, 0]
    ■ "dog" → [0, 1, 0]
    ■ "bird" → [0, 0, 1]

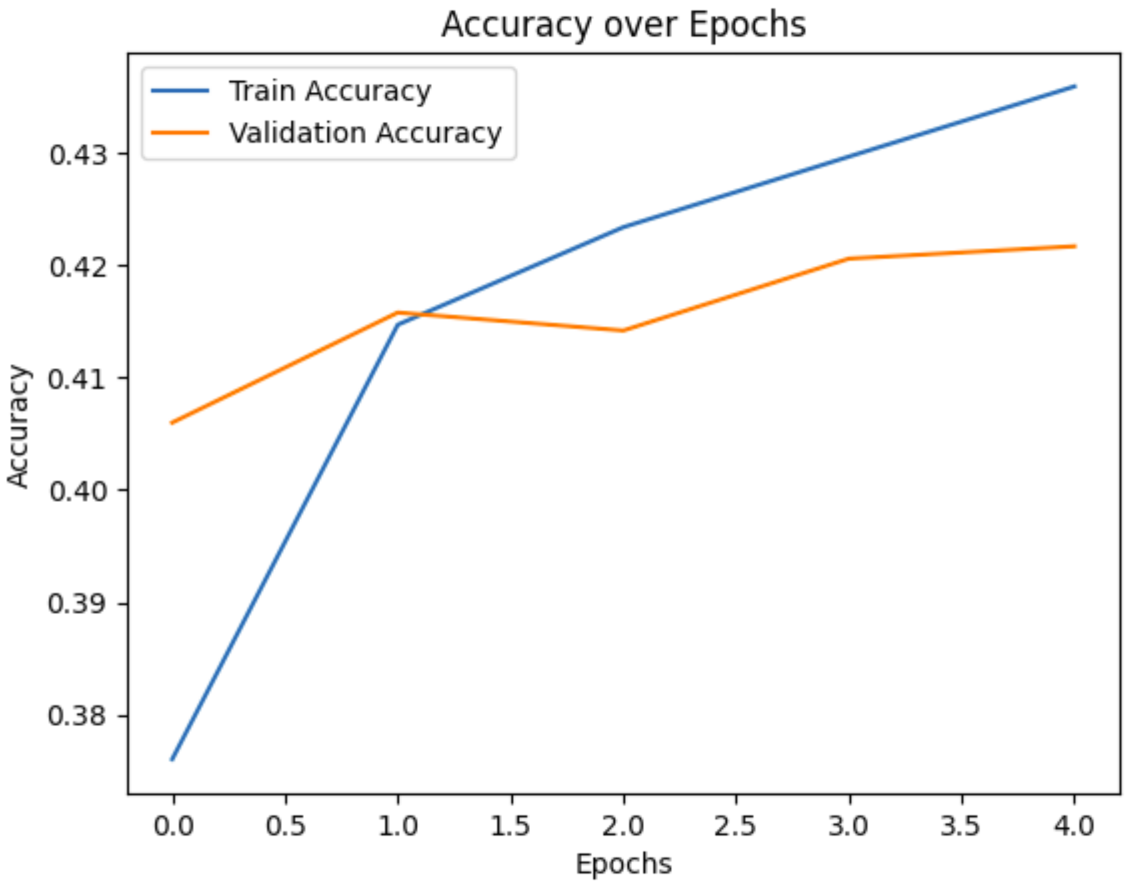
- **Softmax Activation**: In the output layer of a multiclass classification model, **softmax** is often used, which converts the output scores into probabilities, summing to 1. The class with the highest probability is predicted as the output.

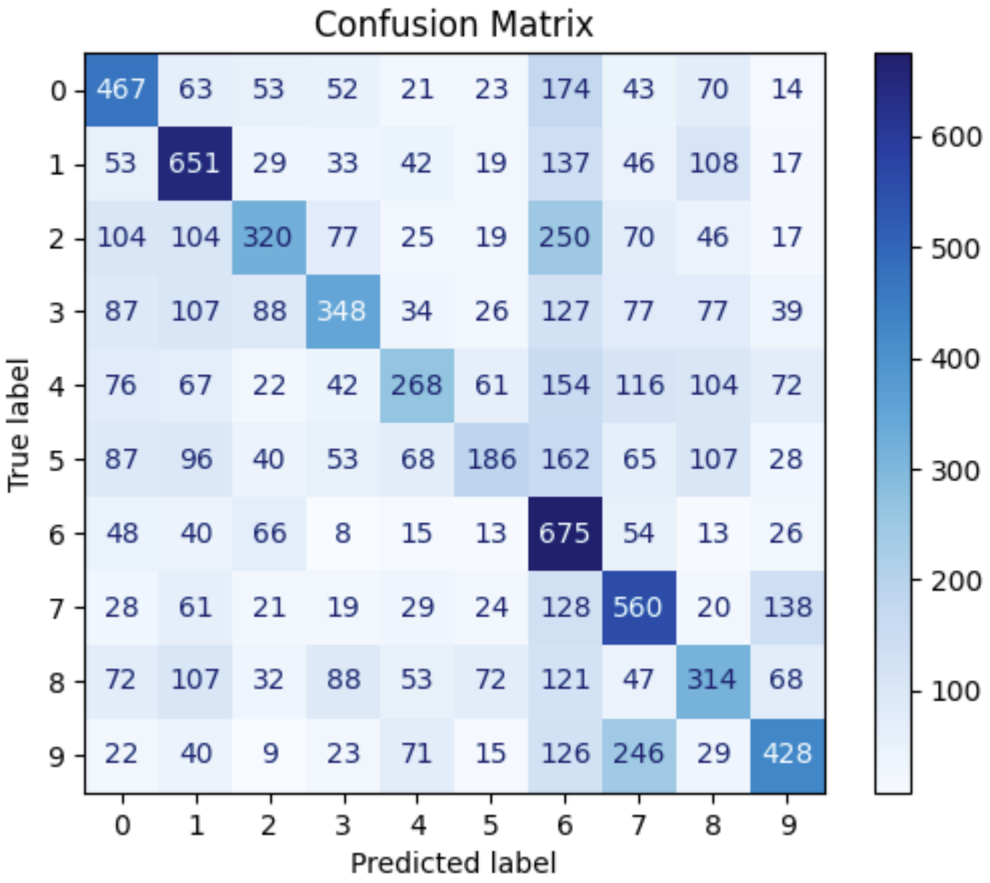
Code:

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.datasets import mnist
import matplotlib.pyplot as plt

# Load dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0 # Normalize

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

# Compile and train
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy']) model.fit(x_train, y_train, epochs=5, validation_data=(x_test, y_test))
```

Output:

```
Epoch 1/5
Train Accuracy: 97%
Test Accuracy: 96%
```


Accuracy over Epochs
Train Accuracy
Validation Accuracy
Accuracy
0.38
0.39
0.40
0.41
0.42
0.43
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs



	0	1	2	3	4	5	6	7	8	9
0	467	63	53	52	21	23	174	43	70	14
1	53	651	29	33	42	19	137	46	108	17
2	104	104	320	77	25	19	250	70	46	17
3	87	107	88	348	34	26	127	77	77	39
4	76	67	22	42	268	61	154	116	104	72
5	87	96	40	53	68	186	162	65	107	28
6	48	40	66	8	15	13	675	54	13	26
7	28	61	21	19	29	24	128	560	20	138
8	72	107	32	88	53	72	121	47	314	68
9	22	40	9	23	71	15	126	246	29	428

Result

The above experiment of Multi Class Classification is done successfully and the output is been obtained