Program 5: Build a network with at least 3 hidden layers that achieves better than 92% accuracy on validation and test data. You may need to train for more than 10 epochs to achieve this result.

Aim:- The aim of this program to build a network with at least 3 hidden layers that achieves better than 92% accuracy on validation and test data. You may need to train for more than 10 epochs to achieve this result.

Procedure:-

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical

                                                    # Load and preprocess data
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
y_train, y_test = to_categorical(y_train), to_categorical(y_test)

                                                        # Build the model
model = Sequential([
    Flatten(input_shape=(28, 28)),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(10, activation='softmax')
])

                                                         # Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

                                                          # Train the model
model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))

                                                        # Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {test_acc:.2%}")
```

Output:-

```
Epoch 10/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0621 - accuracy:
0.9823 - val_loss: 0.1063 - val_accuracy: 0.9660
Test Accuracy: 96.60%
```

Program 6: Build a network for classification using the built-in MNIST dataset.

Aim :- The aim of this program to build a network for classification using the built-in MNIST dataset.

Procedure:-

```
 import tensorflow as tf

                                        # Load and preprocess data
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0  # Normalize
y_train, y_test = tf.keras.utils.to_categorical(y_train), tf.keras.utils.to_categorical(y_test)

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

                                        # Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {test_acc:.2%}")
```

Output:-

During training:

```
Epoch 5/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0657 - accuracy:
0.9811 - val_loss: 0.0861 - val_accuracy: 0.9734
```

After evaluation:

```
Test Accuracy: 97.34%
```