LAB 5

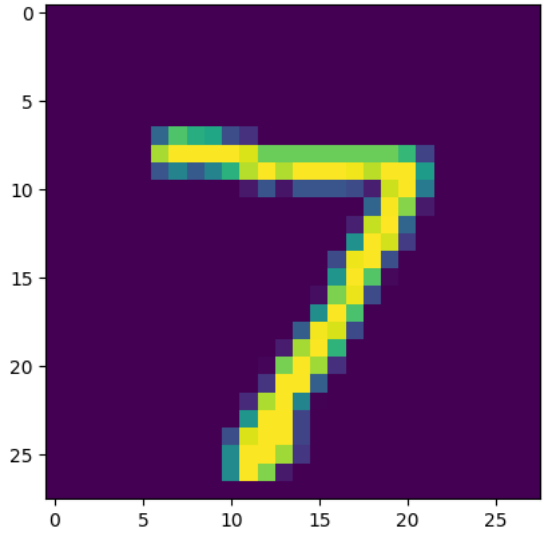
```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
```

```
(trainX,trainY),(testX,testY)=tf.keras.datasets.mnist.load_data()
```

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

```
plt.imshow(testX[0])
```

```
<matplotlib.image.AxesImage at 0x7efba3702af0>
```




```
trainX.shape
```

```
(60000, 28, 28)
```

```
trainY[0]
```

```
5
```

```
testY=tf.keras.utils.to_categorical(testY,num_classes=10) # one hot encoding
trainY=tf.keras.utils.to_categorical(trainY,num_classes=10)
```

```
trainY[0]
```

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

```
#tf.keras.backend.clear_session()
model=tf.keras.models.Sequential()
model.add(tf.keras.layers.Reshape((784,),input_shape=(28,28,)))
model.add(tf.keras.layers.BatchNormalization())
```

```
model.add(tf.keras.layers.Dense(100,activation='relu'))
model.output
```

```
<KerasTensor: shape=(None, 100) dtype=float32 (created by layer 'dense')>
```

```
model.add(tf.keras.layers.Dense(100,activation='relu'))
model.output
```

```
<KerasTensor: shape=(None, 100) dtype=float32 (created by layer 'dense_1')>
```

```
model.add(tf.keras.layers.Dense(100,activation='relu'))
model.output
```

```
<KerasTensor: shape=(None, 100) dtype=float32 (created by layer 'dense_2')>
```

```
model.add(tf.keras.layers.Dense(10,activation='softmax'))
model.output
```

```
<KerasTensor: shape=(None, 10) dtype=float32 (created by layer 'dense_3')>
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 784)               0

 batch_normalization (BatchN  (None, 784)              3136
 ormalization)

 dense (Dense)               (None, 100)               78500

 dense_1 (Dense)             (None, 100)               10100

 dense_2 (Dense)             (None, 100)               10100

 dense_3 (Dense)             (None, 10)                1010

=================================================================
Total params: 102,846
Trainable params: 101,278
Non-trainable params: 1,568
_________________________________________________________________
```

```
model.compile(optimizer='sgd',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
model.fit(trainX,trainY,validation_data=(testX,testY),epochs=3)
```

```
Epoch 1/3
1875/1875 [==============================] - 8s 4ms/step - loss: 0.5238 - accuracy: 0.8466 - val_loss: 0.3235 - val_accuracy: 0.928
Epoch 2/3
1875/1875 [==============================] - 8s 4ms/step - loss: 0.2153 - accuracy: 0.9339 - val_loss: 0.2977 - val_accuracy: 0.943
Epoch 3/3
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1684 - accuracy: 0.9482 - val_loss: 0.3661 - val_accuracy: 0.952
<keras.callbacks.History at 0x7efb966d8d00>
```

```
model.predict(testX[:1])
```

```
1/1 [==============================] - 0s 128ms/step
array([[2.5126926e-06, 5.5060673e-07, 7.9210738e-05, 1.4472604e-04,
        3.3406483e-08, 5.9053008e-07, 5.4310034e-10, 9.9976140e-01,
        1.9221243e-07, 1.0718433e-05]], dtype=float32)
```

```
testY[:1]
```

```
array([[0., 0., 0., 0., 0., 0., 0., 1., 0., 0.]], dtype=float32)
```

```
plt.imshow(testX[0])
```

```
<matplotlib.image.AxesImage at 0x7efb96fc18e0>
```