In [ ]:

```
import tensorflow as tf
import matplotlib as plt
import numpy as np
import keras
```

In [2]:

```
mnistDB=tf.keras.datasets.mnist
```

In [3]:

```
#Splitting The Data
(X_train,Y_train),(X_test,Y_test)=mnistDB.load_data()
X_train=X_train.reshape(60000,28,28,1)
X_test=X_test.reshape(10000,28,28,1)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.
npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```

In [4]:

```
#Data Normalisation
X_train=X_train.astype('float32')/255
X_test=X_test.astype('float32')/255
```

In [5]:

```
#Defining the model

ML=keras.models.Sequential()
ML.add(keras.layers.Conv2D(32,(3,3),activation="relu",input_shape=X_train.shape[1:]))
ML.add(keras.layers.Conv2D(64,(3,3),activation="relu"))
ML.add((keras.layers.BatchNormalization()))
ML.add(keras.layers.MaxPooling2D((2,2)))

ML.add(keras.layers.Dropout(0.25))
ML.add(keras.layers.Flatten())
ML.add(keras.layers.Dense(128,activation='relu'))

ML.add(keras.layers.Dropout(0.25))
ML.add(keras.layers.Dense(units=10,activation="softmax"))
ML.compile(loss="sparse_categorical_crossentropy",optimizer="adam",metrics=['Accuracy'])
es=keras.callbacks.EarlyStopping(monitor='loss',patience=3,restore_best_weights=True)
cp=keras.callbacks.ModelCheckpoint("modelname1.h5",monitor="val_loss")
```

In [6]:

```
ML.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
batch_normalization (BatchNo (None, 24, 24, 64)        256
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 12, 12, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
```

```
_________________________________________________________________
dense (Dense)                (None, 128)               1179776
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 1,200,138
Trainable params: 1,200,010
Non-trainable params: 128
_________________________________________________________________
```

In [7]:

```
#Training the data
history=ML.fit(X_train,Y_train,epochs=25,batch_size=16,callbacks=[es,cp])
```

```
Epoch 1/25
3750/3750 [==============================] - 47s 5ms/step - loss: 0.1481 - Accuracy: 0.9
576
Epoch 2/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0749 - Accuracy: 0.9
782
Epoch 3/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0559 - Accuracy: 0.9
834
Epoch 4/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0491 - Accuracy: 0.9
865
Epoch 5/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0427 - Accuracy: 0.9
879
Epoch 6/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0393 - Accuracy: 0.9
887
Epoch 7/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0350 - Accuracy: 0.9
898
Epoch 8/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0310 - Accuracy: 0.9
915
Epoch 9/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0306 - Accuracy: 0.9
915
Epoch 10/25
3750/3750 [==============================] - 20s 5ms/step - loss: 0.0284 - Accuracy: 0.9
923
Epoch 11/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0245 - Accuracy: 0.9
932
Epoch 12/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0235 - Accuracy: 0.9
935
Epoch 13/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0261 - Accuracy: 0.9
932
Epoch 14/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0210 - Accuracy: 0.9
941
Epoch 15/25
3750/3750 [==============================] - 20s 5ms/step - loss: 0.0227 - Accuracy: 0.9
937
Epoch 16/25
3750/3750 [==============================] - 20s 5ms/step - loss: 0.0211 - Accuracy: 0.9
943
Epoch 17/25
3750/3750 [==============================] - 19s 5ms/step - loss: 0.0211 - Accuracy: 0.9
943
```

In [8]:

```
testloss,testaccuracy=ML.evaluate(X_test,Y_test)
print("Test loss:", testloss)
print("Test accuracy:",testaccuracy)
```

```
313/313 [==============================] - 1s 4ms/step - loss: 0.0524 - Accuracy: 0.9913
Test loss: 0.052437249571084976
Test accuracy: 0.9912999868392944
```