```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from  sklearn.metrics import accuracy_score
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import Sequential,layers
from tensorflow.keras.layers import Dense
from keras.layers import Activation
from keras.datasets import mnist
```

+ Code + Text

```
(x_train,y_train),(x_test,y_test)=mnist.load_data()


print(y_test.shape)
```

```
(10000,)
```

```
x_train=x_train.reshape(60000,784)
x_test=x_test.reshape(10000,784)


print(x_test.shape)
print(y_train.shape)
```

```
(10000, 784)
(60000,)
```

```
from keras.utils import np_utils
n_classes=10
y_train=np_utils.to_categorical(y_train,n_classes)
ytest_cat=y_test
y_test=np_utils.to_categorical(y_test,n_classes)
print(y_train.shape)
```

```
(60000, 10)
```

```
model=Sequential()
model.add(Dense(64,activation='relu',input_shape=(784,)))
model.add(Dense(10,activation='relu'))
model.add(Dense(15,activation='relu'))
model.add(Dense(10,activation='softmax'))
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 64)                50240
```

```
 dense_5 (Dense)             (None, 10)                650

 dense_6 (Dense)             (None, 15)                165

 dense_7 (Dense)             (None, 10)                160

=================================================================
Total params: 51,215
Trainable params: 51,215
Non-trainable params: 0
_________________________________________________________________
```

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

```
model.fit(x_train,y_train,epochs=5,batch_size=10)
```

```
Epoch 1/5
6000/6000 [==============================] - 24s 4ms/step - loss: 2.1043 - accuracy:
Epoch 2/5
6000/6000 [==============================] - 23s 4ms/step - loss: 1.9824 - accuracy:
Epoch 3/5
6000/6000 [==============================] - 22s 4ms/step - loss: 1.5161 - accuracy:
Epoch 4/5
6000/6000 [==============================] - 23s 4ms/step - loss: 0.9206 - accuracy:
Epoch 5/5
6000/6000 [==============================] - 22s 4ms/step - loss: 0.5125 - accuracy:
<keras.callbacks.History at 0x1cfd0301c10>
```