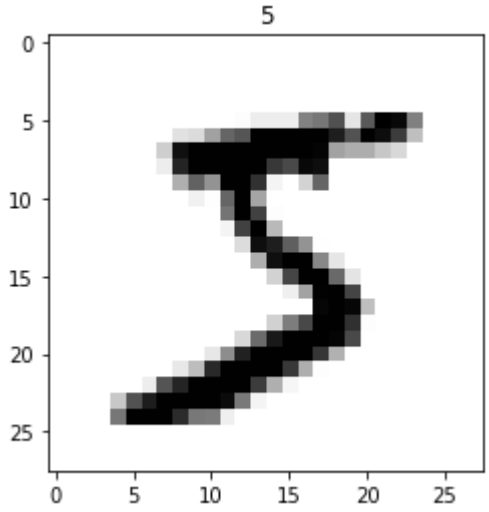
```
%matplotlib inline
import numpy as np

import matplotlib.pyplot as plt

import keras
from keras.datasets import mnist
```

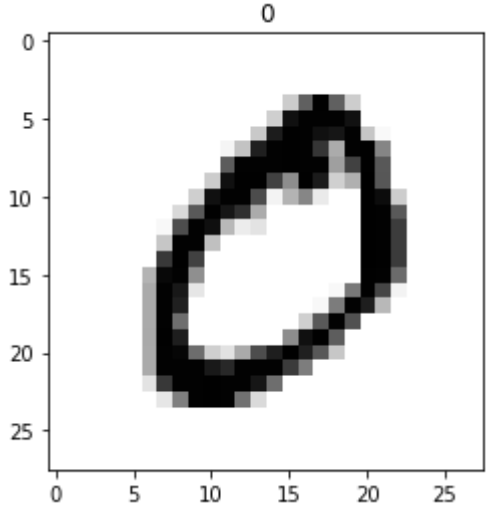
GET DATA

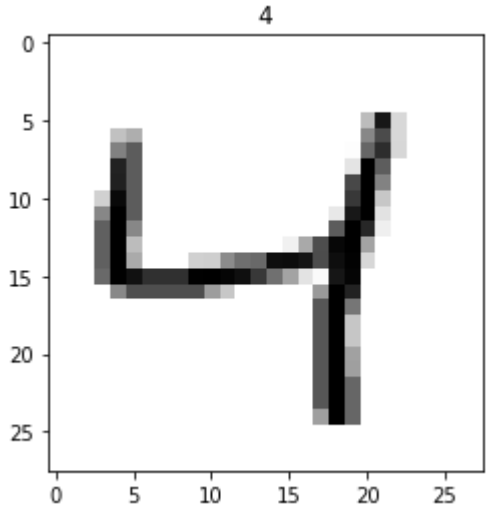
```
(X_train, y_train), (X_test, y_test)=mnist.load_data()

X_train.shape,y_train.shape,X_test.shape,y_test.shape
```

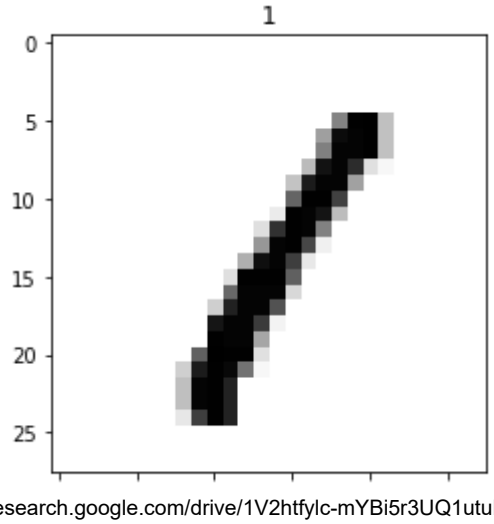
```
((60000, 28, 28), (60000,), (10000, 28, 28), (10000,))
```

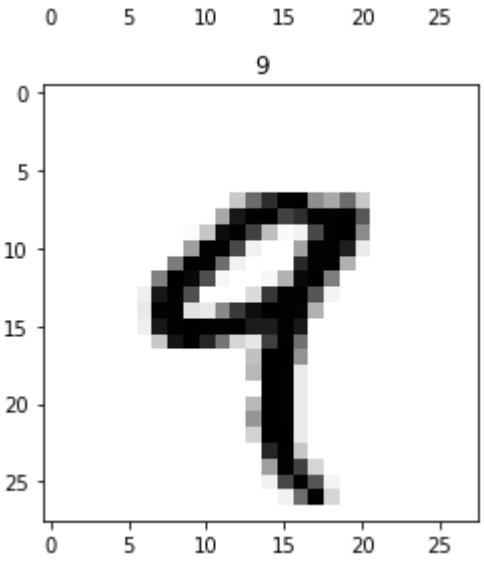
```
def plot_input_img(i):
  plt.imshow(X_train[i],cmap='binary')
  plt.title(y_train[i])
  plt.show()


for i in range(10):
  plot_input_img(i)
```

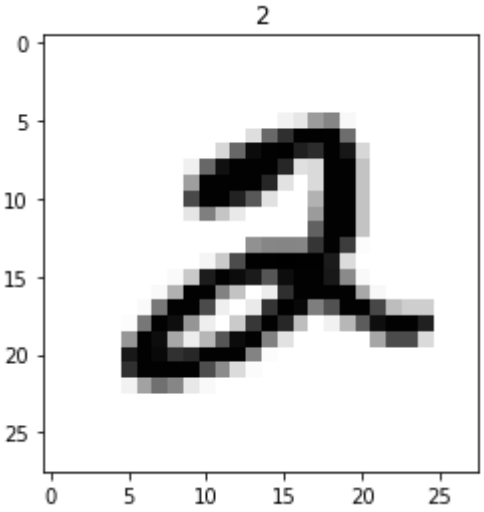

5

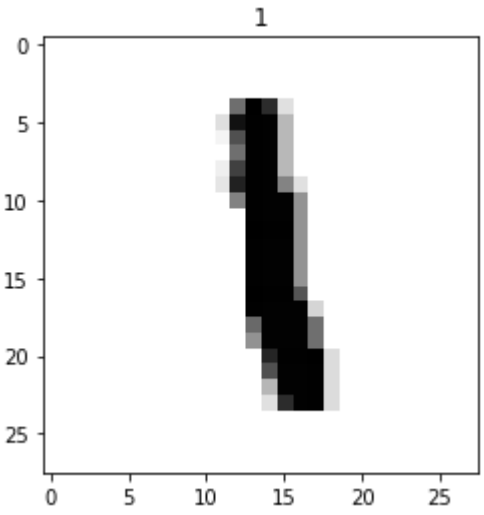

0

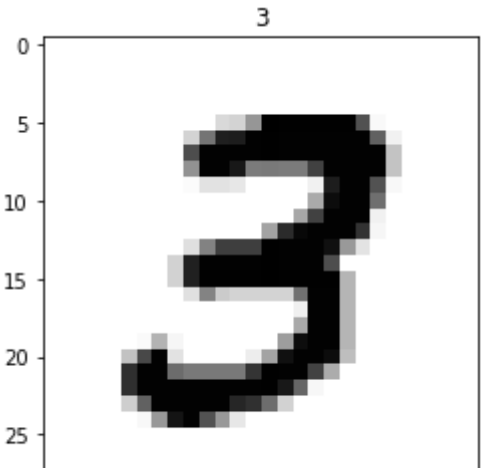

4

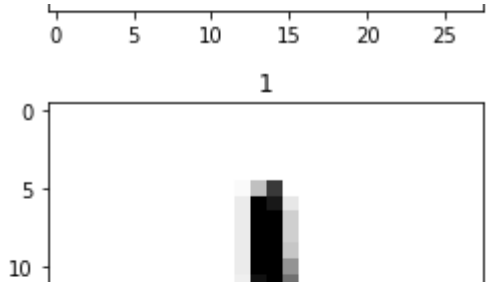

1


9


2


1


3



Pre Process

```
X_train=X_train.astype(np.float32)/255
X_test=X_test.astype(np.float32)/255

X_train=np.expand_dims(X_train,-1)
X_test=np.expand_dims(X_test,-1)
```

```
y_train.shape
```

```
(60000,)
```

```
# Classes to one hot vector

keras.utils.to_categorical(y_train)
keras.utils.to_categorical(y_test)
```

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

```
from keras.models import Sequential
from keras.layers import Dense,Conv2D,MaxPool2D,Dropout,Flatten


model=Sequential()
model.add(Conv2D(32,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(MaxPool2D(2,2))

model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(MaxPool2D(2,2))

model.add(Flatten())

model.add(Dropout(0.5))

model.add(Dense(10,activation="softmax"))
```

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_4 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 flatten_2 (Flatten)         (None, 1600)              0

 dropout_2 (Dropout)         (None, 1600)              0

 dense_2 (Dense)             (None, 10)                16010

=================================================================
Total params: 34,826
Trainable params: 34,826
Non-trainable params: 0
_________________________________________________________________
```

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


#Callbacks

from keras.callbacks import EarlyStopping,ModelCheckpoint

# Earlystopping
es=EarlyStopping(monitor='val_acc',min_delta=0.01,patience=4,verbose=1)

# Model Check Point
mc=ModelCheckpoint("/content/bestmodel.h5",monitor="val_acc",verbose=1,save_best_only=True

cb=[es,mc]


#his = model.fit(X_train,y_train,epochs=50,validation_split=0.3,callbacks=cb)


#model_S=keras.models.load_model("path")
#score=model_S.evaluate(X_test,y_test)

#print(f" the model accuracy is {score[1]} ")
```

Colab paid products - Cancel contracts here

✓ 0s completed at 00:16