

```
In[ ]: ##Name: Tanmay Gujar
##Rollno:14159
##Subject:ISF-IV (DL)
```

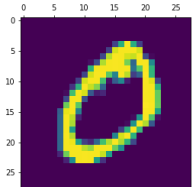
```
In[ ]: !pipinstalltensorflow
```

```
In[1]: #importingnecessarylibraries
importtensorflowastf
fromtensorflowimportkeras
importpandasaspd
importnumpyasnp
importmatplotlib.pyplotasplt
importrandom
importmatplotlibinline
```

```
In[2]: #importdataset andsplit intotrain andtest data
mnist=tf.keras.datasets.mnist
(x_train,y_train),(x_test,y_test)=mnist.load_data()
```

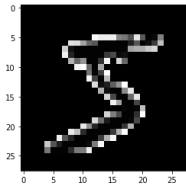
```
In[3]: plt.matshow(x_train[1])
```

Out[3]:<matplotlib.image.AxesImage at 0x216d2a9c9d0>



```
In[4]: plt.imshow(-x_train[0],cmap="gray")
```

Out[4]:<matplotlib.image.AxesImage at 0x216d321cac0>



```
In[5]: x_train=x_train/255x_te
stx_test/255
```

```
In[6]: model =
keras.Sequential([keras.layers.Flatten(input
_shape=(28, 28)),keras.layers.Dense(128,
activation="relu"),keras.layers.Dense(10,acti
vation="softmax")
])
model.summary()
Model:"sequential"
Layer (type) OutputShape Param #
-----
flatten(Flatten) (None, 784) 0
dense (Dense) (None, 128) 100480
dense_1 (Dense) (None, 10) 1290
-----
Total params: 101,770
Trainable params: 101,770
Non-trainable params: 0
```

```
In[7]: model.compile(optimizer="sgd",loss="sparse_c
ategorical_crossentropy",metrics=['accuracy']
)
```

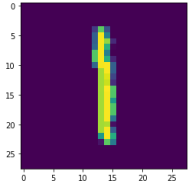
```
In[8]: history=model.fit(x_train,y_train,validation_data=(x_t
est,y_test),epochs=10)

Epoch 1/10
1875/1875[=====] - 3s2ms/step-loss:0.6614-accuracy:0.8326-val_loss:0.3559-val_accuracy:0.9034Epoch2/10
1875/1875[=====] - 3s2ms/step-loss:0.3358-accuracy:0.9069-val_loss:0.2906-val_accuracy:0.9198Epoch3/10
1875/1875[=====] - 3s2ms/step-loss:0.2860-accuracy:0.9197-val_loss:0.2582-val_accuracy:0.9290Epoch4/10
1875/1875[=====] - 3s2ms/step-loss:0.2551-accuracy:0.9286-val_loss:0.2334-val_accuracy:0.9348Epoch5/10
1875/1875[=====] - 3s2ms/step-loss:0.2325-accuracy:0.9351-val_loss:0.2150-val_accuracy:0.9390Epoch6/10
1875/1875[=====] - 3s2ms/step-loss:0.2140-accuracy:0.9397-val_loss:0.2011-val_accuracy:0.9437Epoch7/10
1875/1875[=====] - 3s2ms/step-loss:0.1991-accuracy:0.9441-val_loss:0.1890-val_accuracy:0.9472Epoch8/10
1875/1875[=====] - 3s2ms/step-loss:0.1861-accuracy:0.9479-val_loss:0.1787-val_accuracy:0.9484Epoch9/10
1875/1875[=====] - 3s2ms/step-loss:0.1747-accuracy:0.9509-val_loss:0.1690-val_accuracy:0.9505Epoch10/10
1875/1875[=====] - 3s2ms/step-loss:0.1649- accuracy:0.9537 -val_loss:0.1604 -val_accuracy:0.9537

test_loss,test_acc=model.evaluate(x_test,y_test)pri
nt("Loss=%.3f"
%test_loss)print("Accuracy%.3f"%test_acc)
```

```
In[9]: 313/313[=====] - 0s1ms/step-loss:0.1604-accuracy:0.9537Loss=0.160
Accuracy=0.954
```

```
In[10]: n=random.randint(0,9999)plt.ims
how(x_test[n])plt.show()
```



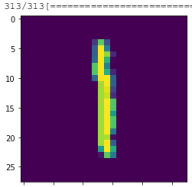
```
In[11]: x_train
```

Out[11]:array([[0.,0., 0.,..., 0.,0., 0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]]])

```
In[12]: x_test
```

Out[12]:array([[0.,0., 0.,..., 0.,0., 0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0.,0., 0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0.,0., 0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0.,0., 0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]],
[[0.,0., 0.,..., 0.,0., 0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
...,
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.],
[0.,0., 0.,..., 0., 0.,0.]]])

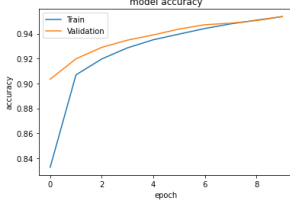
```
In[13]: predicted_value=model.predict(x_test)plt.ims
how(x_test[n])
plt.show()print(predicted_value
[n])
```



[5.5406590e-06 9.9045217e-01 9.6445804e-04 1.1881288e-03 5.5083026e-051.0767796e-035.1782565e-045.0224096e-044.8863539e-033.5130000e-04]

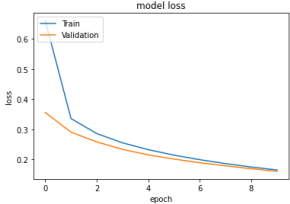
```
In[14]: #history.history()
history.history.keys()
#dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])

plt.plot(history.history['accuracy'])plt.
plot(history.history['val_accuracy'])plt.t
itle('model
accuracy')plt.ylabel('accuracy')plt.xlabe
l('epoch')
plt.legend(['Train','Validation'],loc='upperleft')plt
.show()
```



```
In[15]: #history.history()
history.history.keys()
#dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])

plt.plot(history.history['loss'])plt.
plot(history.history['val_loss'])plt.t
itle('model
loss')plt.ylabel('loss')plt.xlabel('e
poch')
plt.legend(['Train','Validation'],loc='upperleft')plt
.show()
```



```
In[ ]:
```