```
# importing necessary documents

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os
import tensorflow as tf
import seaborn as sns
```

```
!pip install tensorflow-gpu
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting tensorflow-gpu
  Using cached tensorflow-gpu-2.12.0.tar.gz (2.6 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: python_version>"3.7" in /usr/local/lib/python3.8/dist-packages (from tensorflow-gpu) (0.0.2)
Building wheels for collected packages: tensorflow-gpu
  error: subprocess-exited-with-error

  × python setup.py bdist_wheel did not run successfully.
  │ exit code: 1
  ╰─> See above for output.

  note: This error originates from a subprocess, and is likely not a problem with pip.
  Building wheel for tensorflow-gpu (setup.py) ... error
  ERROR: Failed building wheel for tensorflow-gpu
  Running setup.py clean for tensorflow-gpu
Failed to build tensorflow-gpu
Installing collected packages: tensorflow-gpu
  error: subprocess-exited-with-error
```



```
  note: This error originates from a subprocess, and is likely not a problem with pip.
  Running setup.py install for tensorflow-gpu ... error
error: legacy-install-failure

× Encountered error while trying to install package.
╰─> tensorflow-gpu

note: This is an issue with the package mentioned above, not pip.
hint: See above for output from the failure.
```

```
#loading the mnist dataset fron tf.keras

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

```
mnist
```

```
<module 'keras.api._v2.keras.datasets.mnist' from '/usr/local/lib/python3.8/dist-
packages/keras/api/_v2/keras/datasets/mnist/__init__.py'>
```

```
(x_train_full,y_train_full),(x_test,y_test)=mnist.load_data()
```

```
x_train_full
```

```
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]]], dtype=uint8)
```

```
x_train_full.shape
```

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

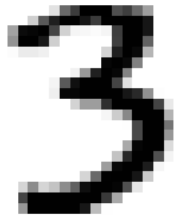
Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
x_train_full[5000]
```

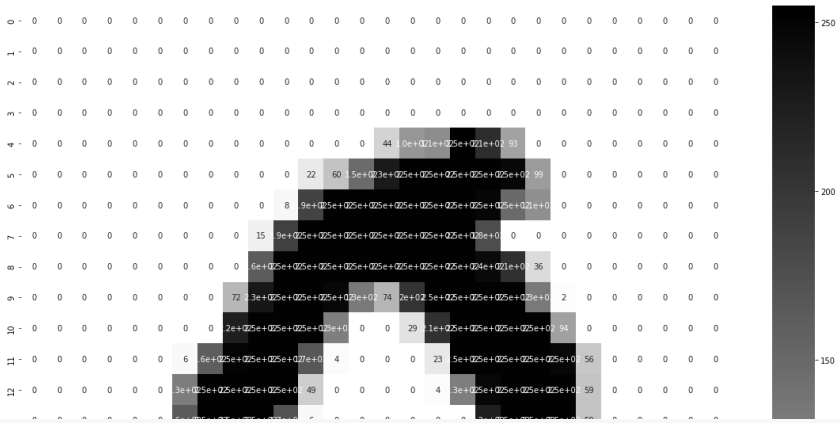
```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0, 140, 251, 254, 254, 254, 254, 254, 254, 254, 254, 254,
        254, 254, 254, 254, 254, 189,  23,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0, 226, 254, 208, 199, 199, 199, 199, 139,  61,  61,  61,
         61,  61, 128, 222, 254, 254, 189,  21,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,  38,  82,  13,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,  34, 213, 254, 254, 115,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,  84, 254, 254, 234,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,  84, 254, 254, 234,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0, 106, 157, 254, 254, 243,  51,   0,   0,   0,
          0,   0],
```

```
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,  89, 251, 241,  86,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   5, 206, 246, 157,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   4, 117,  69,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```

```
plt.imshow(x_train_full[10000],cmap='binary')
plt.axis("off")
plt.show()
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3e50bed5b0>
```



```
# Scale the data between 0 to 1 by dividing it by 255. as its an unsigned data between 0-255 range
x_valid, x_train = x_train_full[:5000] / 255., x_train_full[5000:] / 255.
y_valid, y_train = y_train_full[:5000] / 255., y_train_full[5000:] / 255.
```

```
# scale the test set as well
x_test = x_test / 255.
```

```
print(x_valid.shape)
print(x_train.shape)
print(y_valid.shape)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
(5000, 28, 28)
(55000, 28, 28)
(5000,)
(55000,)
```

```
layers = [tf.keras.layers.Flatten(input_shape=[28,28],name='inputlayer'),
          tf.keras.layers.Dense(300,activation='relu', name='hiddenlayer1'),
          tf.keras.layers.Dense(100,activation='relu', name='hiddenlayer2'),
          tf.keras.layers.Dense(10, activation = 'softmax', name = 'outputlayer')]
```

```
model = tf.keras.models.Sequential(layers)
```

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inputlayer (Flatten)        (None, 784)               0

 hiddenlayer1 (Dense)        (None, 300)               235500

 hiddenlayer2 (Dense)        (None, 100)               30100

 outputlayer (Dense)         (None, 10)                1010

=================================================================
Total params: 266,610
Trainable params: 266,610
Non-trainable params: 0
_________________________________________________________________
```

```
loss_function = 'sparse_categorical_crossentropy'
Optimizer = 'adam'
metrics = ['accuracy']
```

```
model.layers[1].name
```

```
'hiddenlayer1'
```

```
model.compile(loss = loss_function,optimizer=Optimizer, metrics = metrics)
```

```
model.compile(loss = loss_function,optimizer=optimizer, metrics = metrics)
```

```
# getting the weights

hidden1 = model.layers[1]
```

```
weights_biases = hidden1.get_weights()
```

```
#Training the model

epochs = 5
validation_set =(x_valid,y_valid)
history = model.fit(x_train,y_train,epochs=epochs,validation_data=validation_set)
```

```
Epoch 1/5
1719/1719 [==============================] - 10s 6ms/step - loss: 2.7960e-09 - accuracy: 0.0990 - val_loss: 5.7220e-10 - val_accuracy: 0
Epoch 2/5
1719/1719 [==============================] - 10s 6ms/step - loss: 1.4023e-09 - accuracy: 0.0990 - val_loss: 3.0994e-10 - val_accuracy: 0
Epoch 3/5
1719/1719 [==============================] - 9s 6ms/step - loss: 7.3043e-10 - accuracy: 0.0990 - val_loss: 7.1526e-11 - val_accuracy: 0.
Epoch 4/5
1719/1719 [==============================] - 10s 6ms/step - loss: 4.0531e-10 - accuracy: 0.0990 - val_loss: 0.0000e+00 - val_accuracy: 0
Epoch 5/5
1719/1719 [==============================] - 11s 6ms/step - loss: 2.3408e-10 - accuracy: 0.0990 - val_loss: 0.0000e+00 - val_accuracy: 0
```

```
# model saving

import time
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
  os.makedirs(model_dir, exist_ok=True)
  filename=time.strftime("model_%Y_%m_%d_%H_%M_%S_.h5")
  model_path = os.path.join(model_dir,filename)
  print(f"your model will be saved at the following location\n{model_path}")
  return model_path
```

```
model.save(savemodel_path())
```

```
your model will be saved at the following location
/saved models/model_2023_02_06_09_50_37_.h5
```

```
history.params
```

```
{'verbose': 1, 'epochs': 5, 'steps': 1719}
```

```
#checking the history

pd.DataFrame(history.history)
```

	loss	accuracy	val_loss	val_accuracy
0	2.795990e-09	0.098982	5.722045e-10	0.0958
1	1.402332e-09	0.098982	3.099441e-10	0.0958
2	7.304272e-10	0.098982	7.152557e-11	0.0958
3	4.053114e-10	0.098982	0.000000e+00	0.0958
4	2.340836e-10	0.098982	0.000000e+00	0.0958

```
pd.DataFrame(history.history).plot(figsize=(8,5))
plt.grid=True
plt.gca().set_ylim(0,1)
plt.show()
```


```
#predicting the model

model.evaluate(x_test,y_test)
```

```
313/313 [==============================] - 2s 5ms/step - loss: 2.3406 - accuracy: 0.0980
[2.3405630588531494, 0.09799999743700027]
```

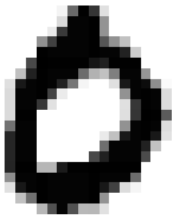
```
x_test[3]
```

```
array([[[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
```



```
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         1.02021518e-11, 1.39120252e-10, 2.34649492e-10, 1.87348607e-10,
         2.87515188e-11, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         3.43163289e-11, 2.32794556e-10, 2.32794556e-10, 2.34649492e-10,
         9.92391134e-11, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 1.94768353e-11,
         1.82711265e-10, 2.32794556e-10, 2.32794556e-10, 2.34649492e-10,
         9.92391134e-11, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 1.02021518e-10, 1.76218986e-10,
         2.32794556e-10, 2.32794556e-10, 2.32794556e-10, 2.34649492e-10,
         1.56742151e-10, 1.01094050e-10, 5.75030377e-11, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
        [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
         0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
```

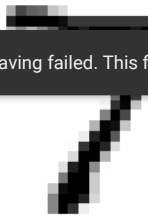
```
plt.imshow(x_test[3],cmap='binary')
plt.axis("off")
plt.show()
```

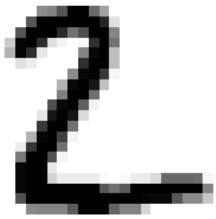


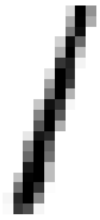
```
prediction = model.predict(x_test[:3])
```

```
1/1 [==============================] - 0s 49ms/step
```

```
for i in range(0,3):
  plt.imshow(x_test[i],cmap='binary')
  plt.axis("off")
  plt.show()
```





```
y_prob = prediction[0]
```

```
#finding the max value
np.argmax(y_prob.round(2))
```

```
0
```

```
#loading the model
```

```
from tf.keras.models import load_model
model=load_model()
```

✓ 0s completed at 4:02 PM

Automatic saving failed. This file was updated remotely or in another tab. Show diff