Summer 2024: CS5720 NNDL - ICP-3

Lasya Vanga **(700762893)**

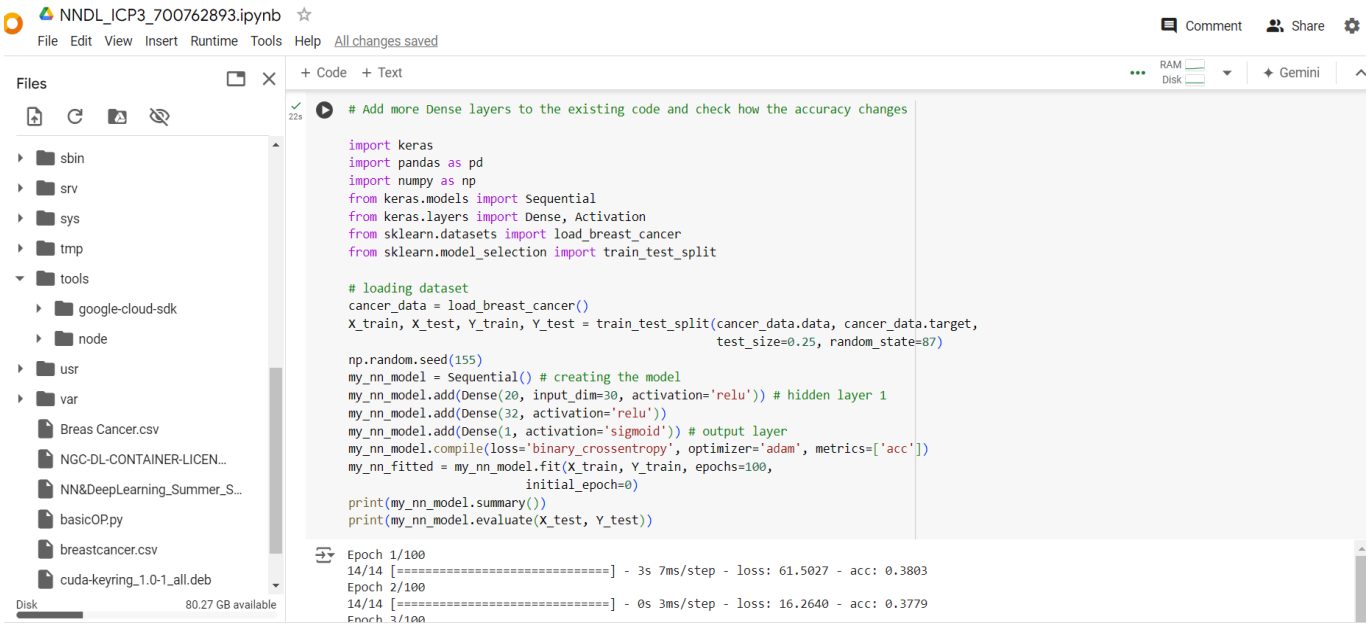
GitHub Link: https://github.com/Lasya-vanga/NNDL-ICP3

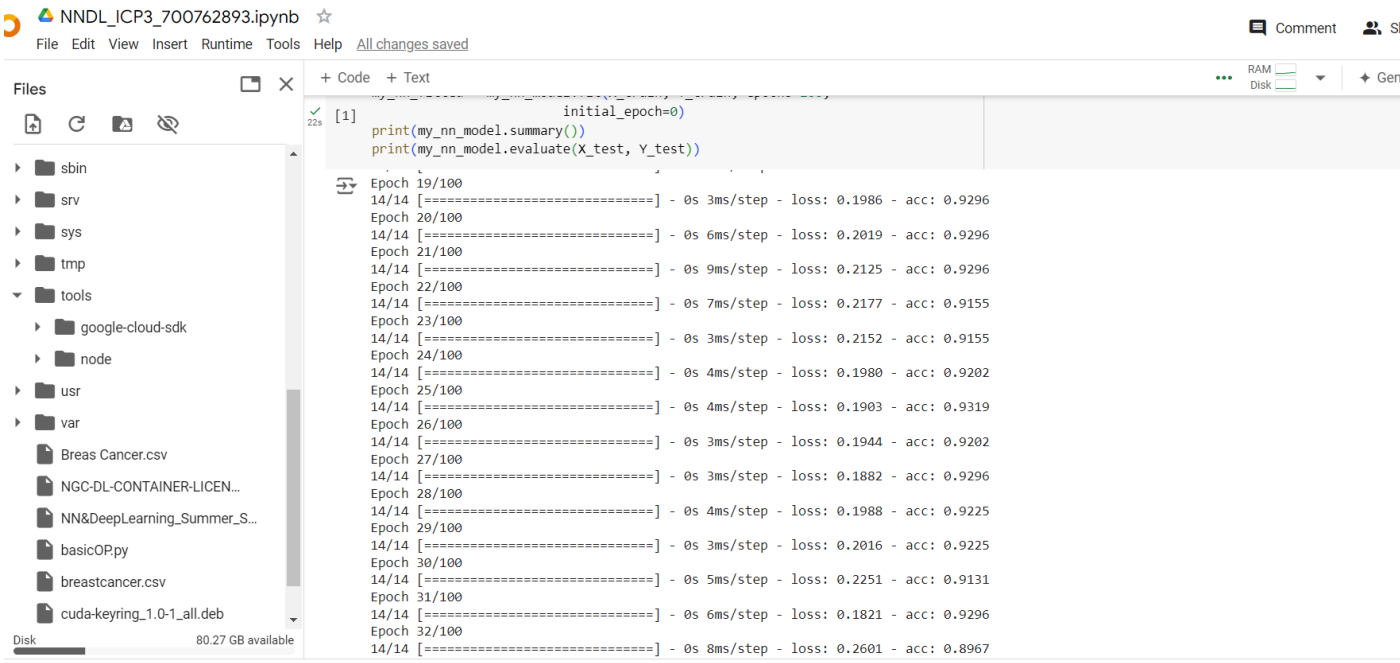
Use Case Description: Predicting the diabetes disease

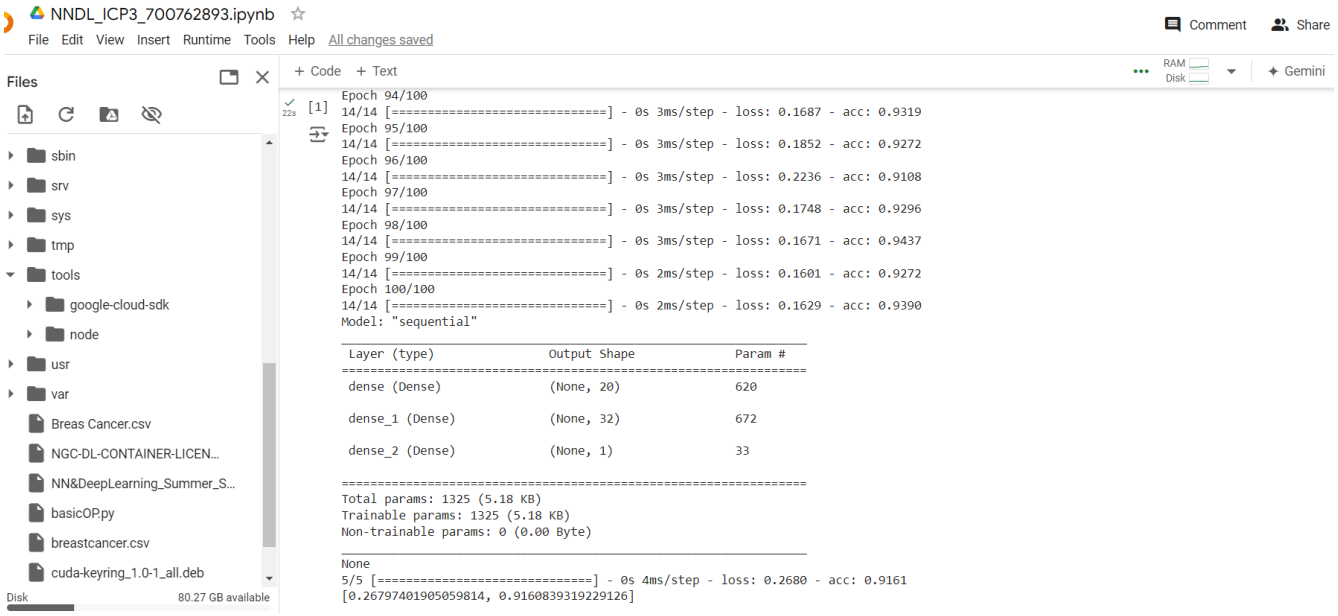
Programming elements: Keras Basics

1) Use the use case in the class:

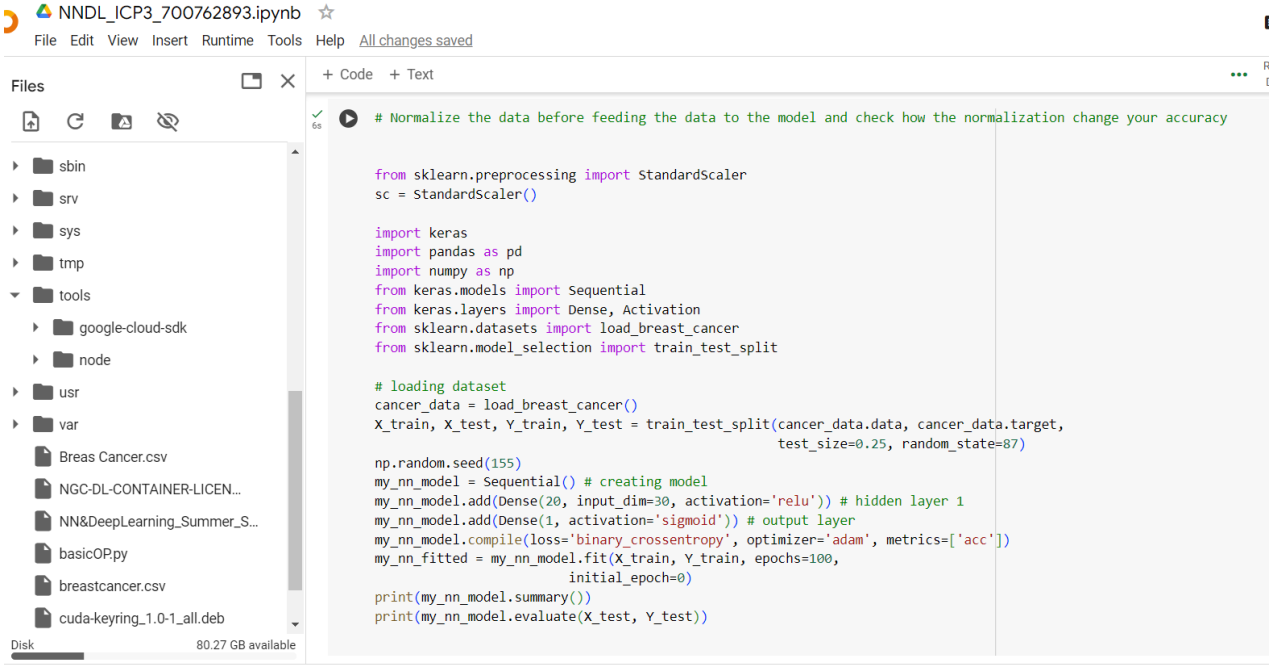
a. Add more Dense layers to the existing code and check how the accuracy changes

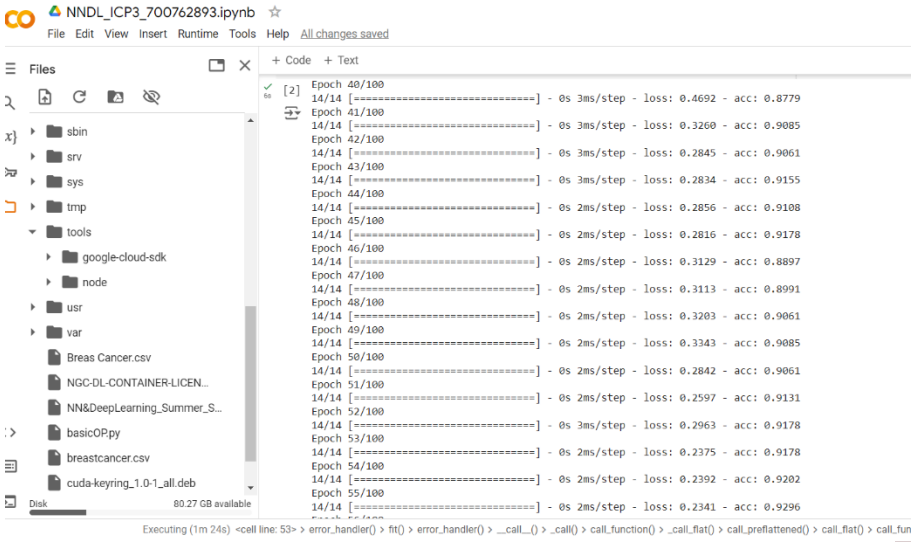


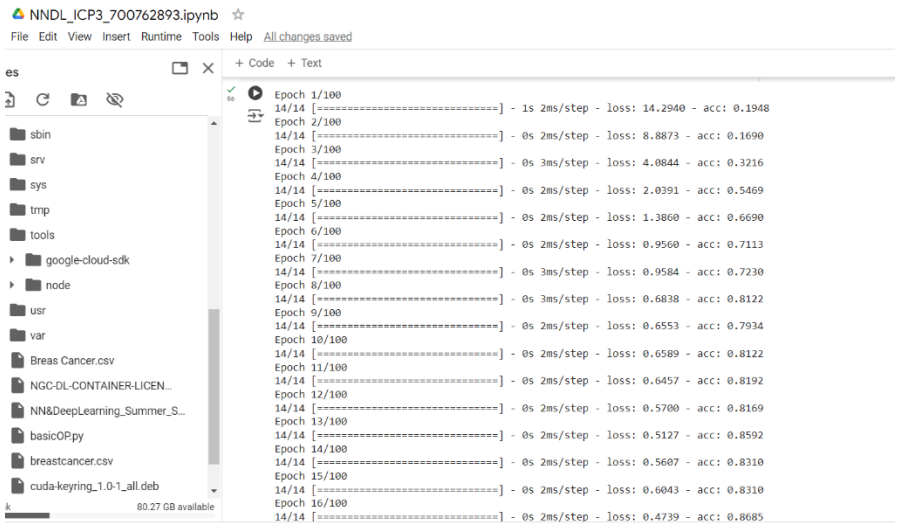


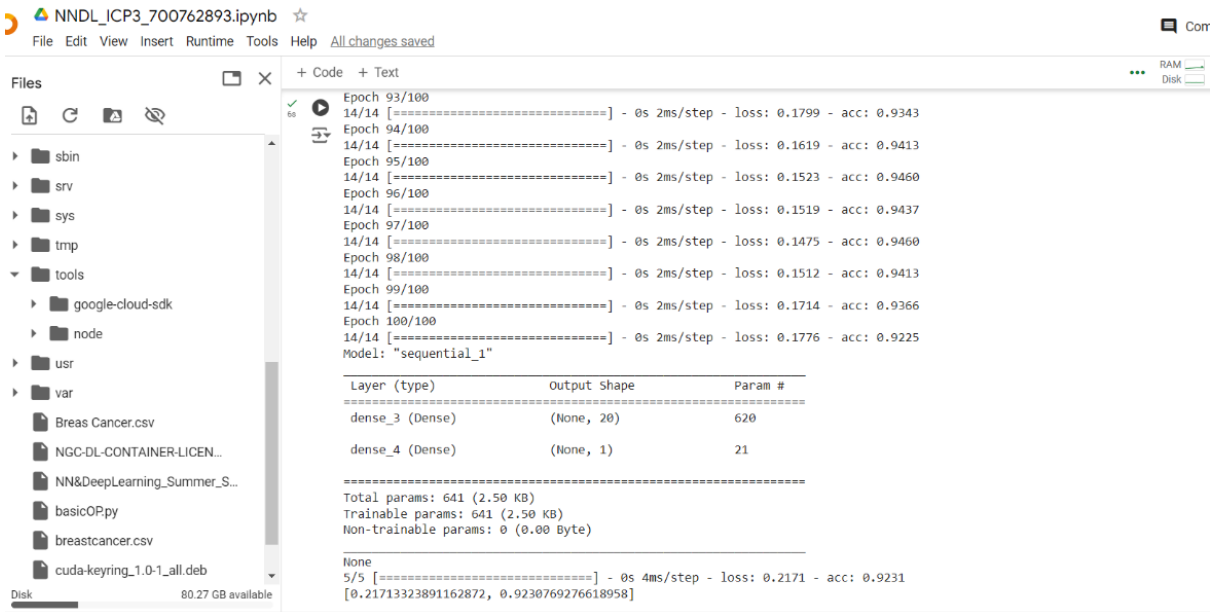


2. Normalize the data before feeding the data to the model and check how the normalization change your accuracy (code given below). from sklearn.preprocessing import StandardScaler sc = StandardScaler()



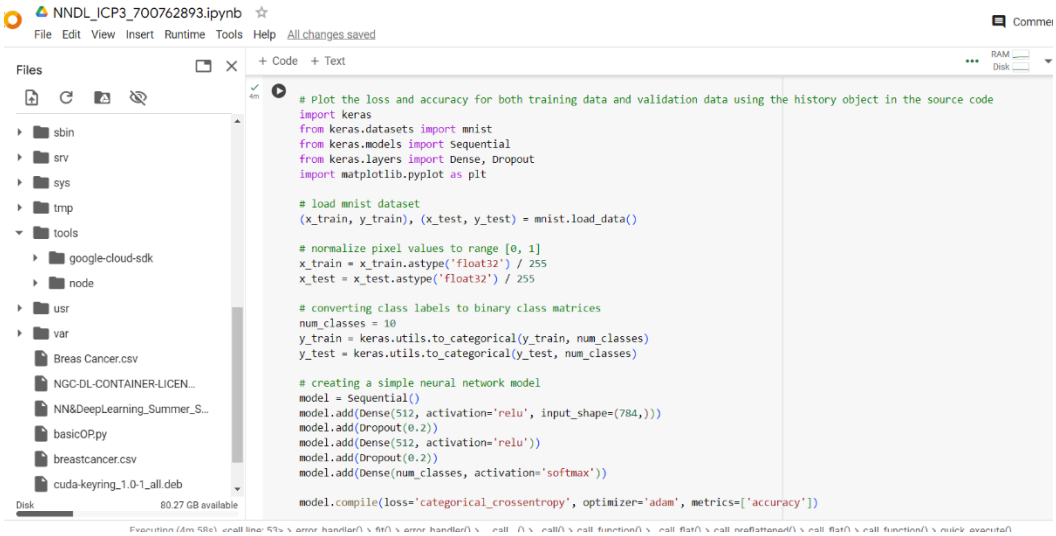

NNDL_ICP3_700762893.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
+ Code + Text
[2]
Epoch 40/100
14/14 [==========] - 0s 3ms/step - loss: 0.4692 - acc: 0.8779
Epoch 41/100
14/14 [==========] - 0s 3ms/step - loss: 0.3260 - acc: 0.9085
Epoch 42/100
14/14 [==========] - 0s 3ms/step - loss: 0.2845 - acc: 0.9061
Epoch 43/100
14/14 [==========] - 0s 3ms/step - loss: 0.2834 - acc: 0.9155
Epoch 44/100
14/14 [==========] - 0s 2ms/step - loss: 0.2856 - acc: 0.9108
Epoch 45/100
14/14 [==========] - 0s 2ms/step - loss: 0.2816 - acc: 0.9178
Epoch 46/100
14/14 [==========] - 0s 2ms/step - loss: 0.3129 - acc: 0.8897
Epoch 47/100
14/14 [==========] - 0s 2ms/step - loss: 0.3113 - acc: 0.8991
Epoch 48/100
14/14 [==========] - 0s 2ms/step - loss: 0.3203 - acc: 0.9061
Epoch 49/100
14/14 [==========] - 0s 2ms/step - loss: 0.3343 - acc: 0.9085
Epoch 50/100
14/14 [==========] - 0s 2ms/step - loss: 0.2842 - acc: 0.9061
Epoch 51/100
14/14 [==========] - 0s 2ms/step - loss: 0.2597 - acc: 0.9131
Epoch 52/100
14/14 [==========] - 0s 3ms/step - loss: 0.2963 - acc: 0.9178
Epoch 53/100
14/14 [==========] - 0s 2ms/step - loss: 0.2375 - acc: 0.9178
Epoch 54/100
14/14 [==========] - 0s 2ms/step - loss: 0.2392 - acc: 0.9202
Epoch 55/100
14/14 [==========] - 0s 2ms/step - loss: 0.2341 - acc: 0.9296
sbin
srv
sys
tmp
tools
google-cloud-sdk
node
usr
var
Breas Cancer.csv
NGC-DL-CONTAINER-LICEN...
NN&DeepLearning_Summer_S...
basicOP.py
breastcancer.csv
cuda-keyring_1.0-1_all.deb
Disk 80.27 GB available
Executing (1m 24s) <cell line: 53> > error_handler() > fit() > error_handler() > __call__() > _call() > call_function() > _call_flat() > call_preflattened() > call_flat() > call_fun

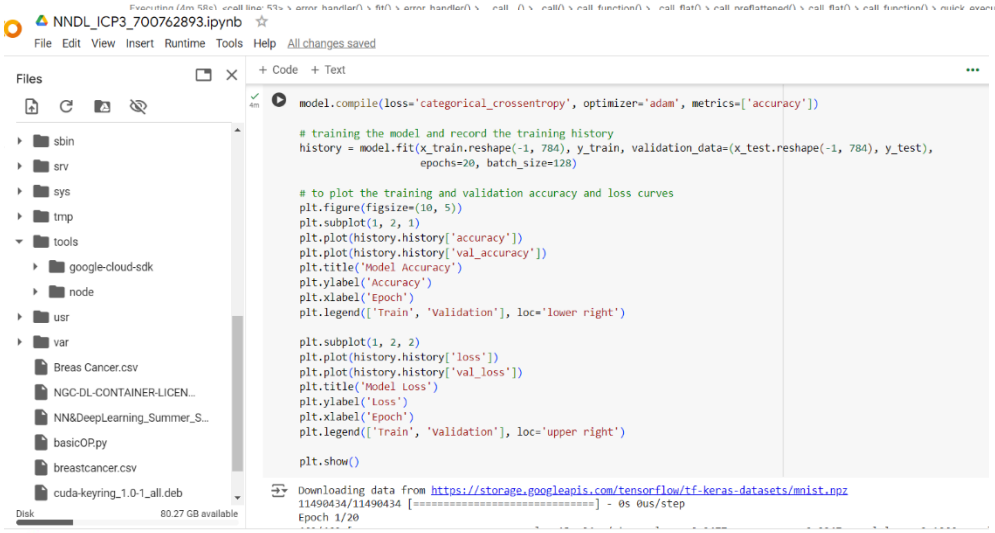

NNDL_ICP3_700762893.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Epoch 1/100
14/14 [==========] - 1s 2ms/step - loss: 14.2940 - acc: 0.1948
Epoch 2/100
14/14 [==========] - 0s 2ms/step - loss: 8.8873 - acc: 0.1690
Epoch 3/100
14/14 [==========] - 0s 3ms/step - loss: 4.0844 - acc: 0.3216
Epoch 4/100
14/14 [==========] - 0s 2ms/step - loss: 2.0391 - acc: 0.5469
Epoch 5/100
14/14 [==========] - 0s 2ms/step - loss: 1.3860 - acc: 0.6690
Epoch 6/100
14/14 [==========] - 0s 2ms/step - loss: 0.9560 - acc: 0.7113
Epoch 7/100
14/14 [==========] - 0s 3ms/step - loss: 0.9584 - acc: 0.7230
Epoch 8/100
14/14 [==========] - 0s 3ms/step - loss: 0.6838 - acc: 0.8122
Epoch 9/100
14/14 [==========] - 0s 2ms/step - loss: 0.6553 - acc: 0.7934
Epoch 10/100
14/14 [==========] - 0s 2ms/step - loss: 0.6589 - acc: 0.8122
Epoch 11/100
14/14 [==========] - 0s 2ms/step - loss: 0.6457 - acc: 0.8192
Epoch 12/100
14/14 [==========] - 0s 2ms/step - loss: 0.5700 - acc: 0.8169
Epoch 13/100
14/14 [==========] - 0s 2ms/step - loss: 0.5127 - acc: 0.8592
Epoch 14/100
14/14 [==========] - 0s 2ms/step - loss: 0.5607 - acc: 0.8310
Epoch 15/100
14/14 [==========] - 0s 2ms/step - loss: 0.6043 - acc: 0.8310
Epoch 16/100
14/14 [==========] - 0s 2ms/step - loss: 0.4739 - acc: 0.8685
sbin
srv
sys
tmp
tools
google-cloud-sdk
node
usr
var
Breas Cancer.csv
NGC-DL-CONTAINER-LICEN...
NN&DeepLearning_Summer_S...
basicOP.py
breastcancer.csv
cuda-keyring_1.0-1_all.deb
80.27 GB available

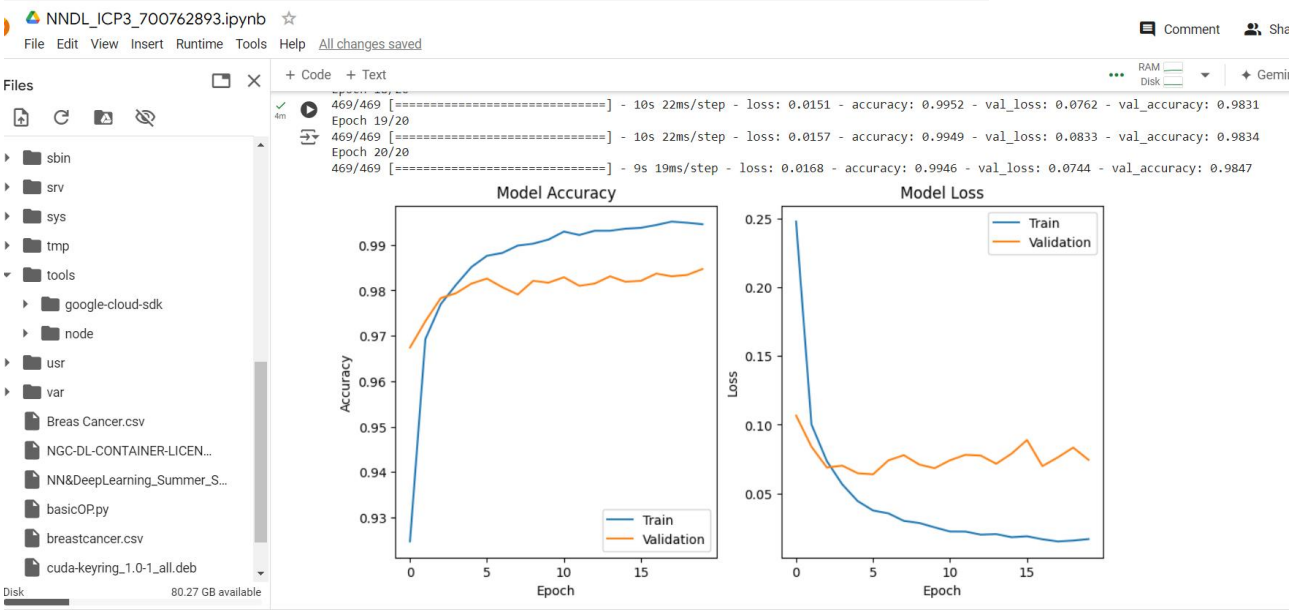

NNDL_ICP3_700762893.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
+ Code + Text
RAM
Disk
Epoch 93/100
14/14 [==========] - 0s 2ms/step - loss: 0.1799 - acc: 0.9343
Epoch 94/100
14/14 [==========] - 0s 2ms/step - loss: 0.1619 - acc: 0.9413
Epoch 95/100
14/14 [==========] - 0s 2ms/step - loss: 0.1523 - acc: 0.9460
Epoch 96/100
14/14 [==========] - 0s 2ms/step - loss: 0.1519 - acc: 0.9437
Epoch 97/100
14/14 [==========] - 0s 2ms/step - loss: 0.1475 - acc: 0.9460
Epoch 98/100
14/14 [==========] - 0s 2ms/step - loss: 0.1512 - acc: 0.9413
Epoch 99/100
14/14 [==========] - 0s 2ms/step - loss: 0.1714 - acc: 0.9366
Epoch 100/100
14/14 [==========] - 0s 2ms/step - loss: 0.1776 - acc: 0.9225
Model: "sequential_1"
Layer (type) Output Shape Param #
dense_3 (Dense) (None, 20) 620
dense_4 (Dense) (None, 1) 21
Total params: 641 (2.50 KB)
Trainable params: 641 (2.50 KB)
Non-trainable params: 0 (0.00 Byte)
None
5/5 [==========] - 0s 4ms/step - loss: 0.2171 - acc: 0.9231
[0.2171332389162872, 0.9230769276618958]
sbin
srv
sys
tmp
tools
google-cloud-sdk
node
usr
var
Breas Cancer.csv
NGC-DL-CONTAINER-LICEN...
NN&DeepLearning_Summer_S...
basicOP.py
breastcancer.csv
cuda-keyring_1.0-1_all.deb
Disk 80.27 GB available

Use Image Classification on the hand written digits data set (mnist)

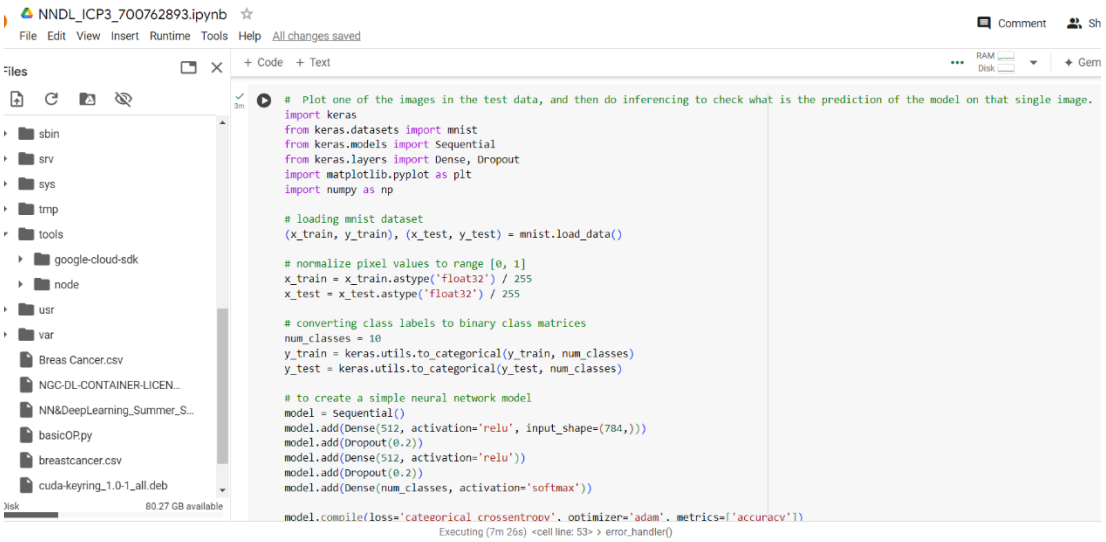
1. Plot the loss and accuracy for both training data and validation data using the history object in the source code.

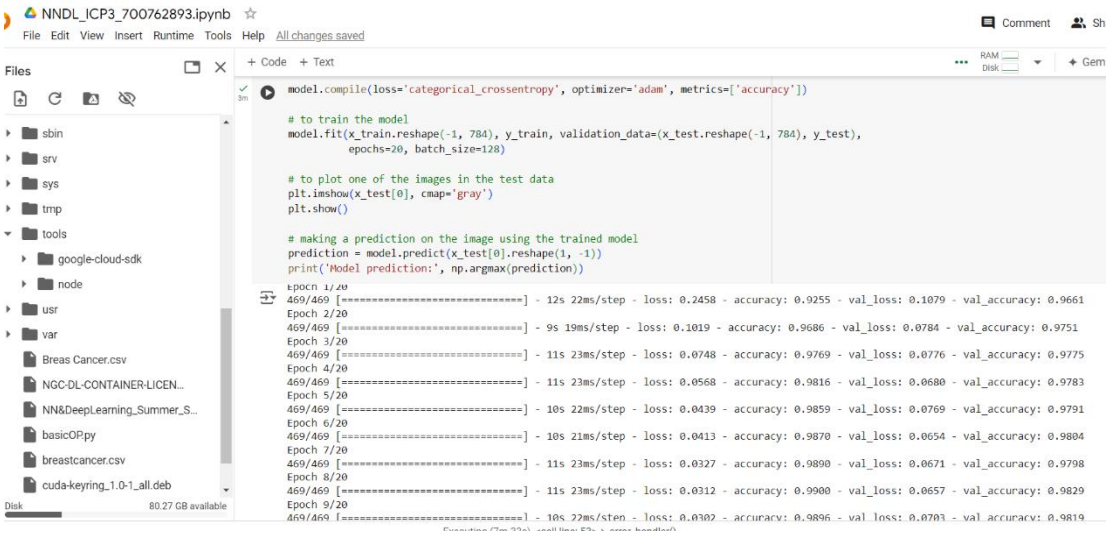


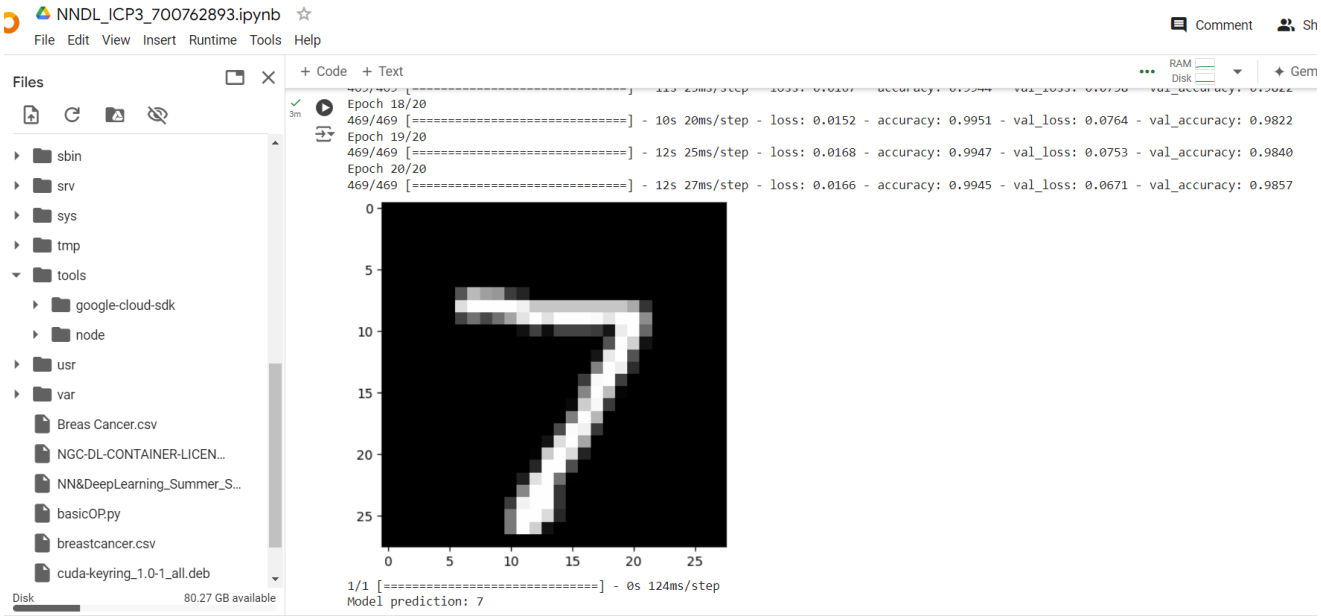




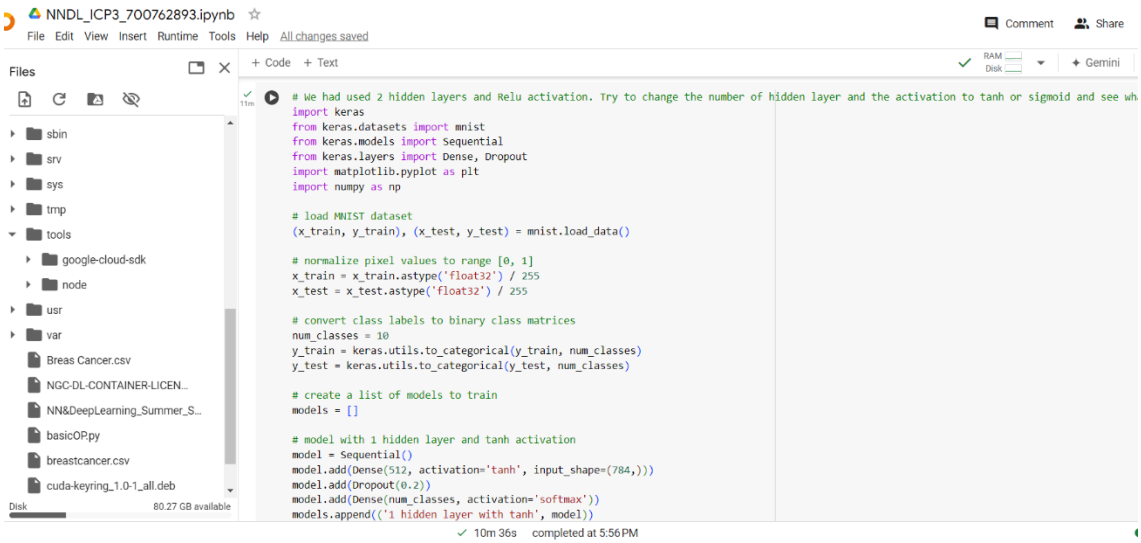
2. Plot one of the images in the test data, and then do inferencing to check what is the prediction of the model on that single image.

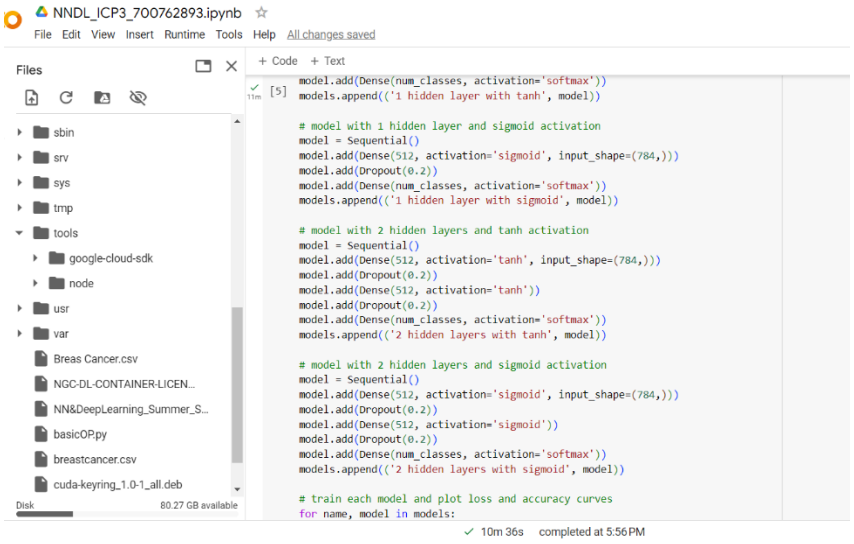


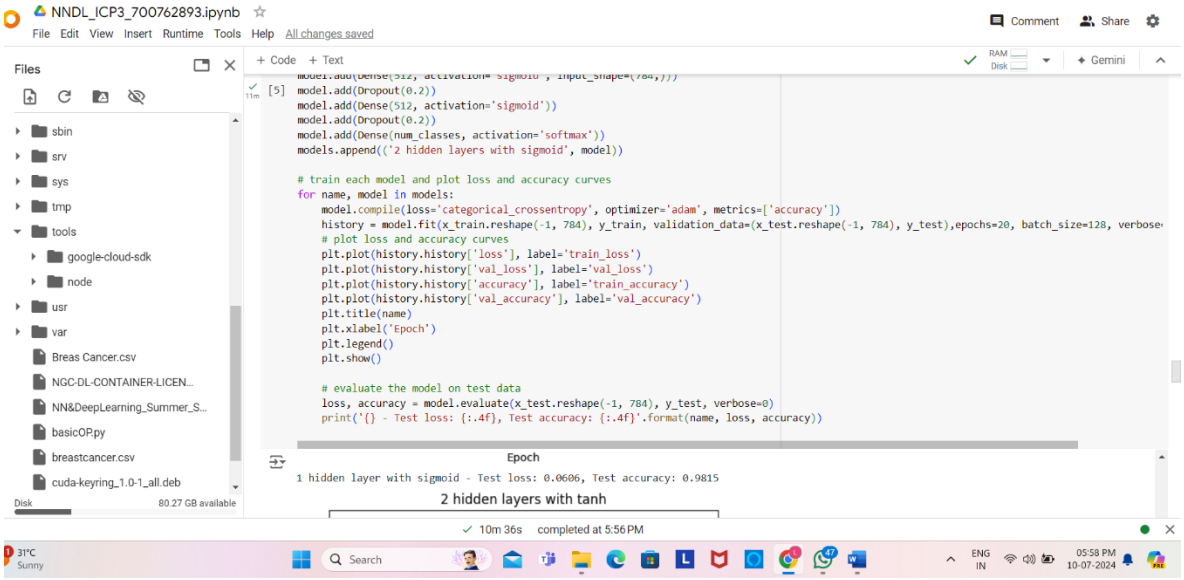




3. We had used 2 hidden layers and Relu activation. Try to change the number of hidden layer and the activation to tanh or sigmoid and see what happens.







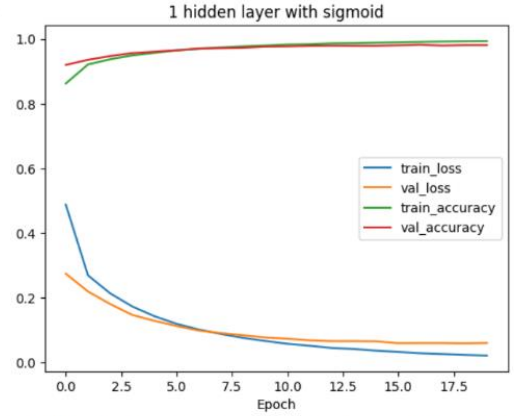
NNDL_ICP3_700762893.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

+ Code + Text

```
loss, accuracy = model.evaluate(x_test.reshape(-1, 784), y_test, verbose=0)
print('{} - Test loss: {:.4f}, Test accuracy: {:.4f}'.format(name, loss, accuracy))
```




```
1 hidden layer with sigmoid - Test loss: 0.0606, Test accuracy: 0.9815
```

2 hidden layers with tanh

✓ 10m 36s completed at 5:56 PM

NNDL_ICP3_700762893.ipynb

File Edit View Insert Runtime Tools Help All changes saved

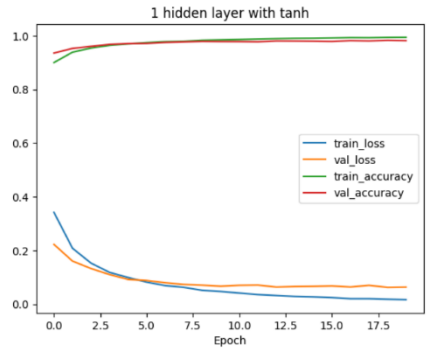
Comment Share

Files

+ Code + Text

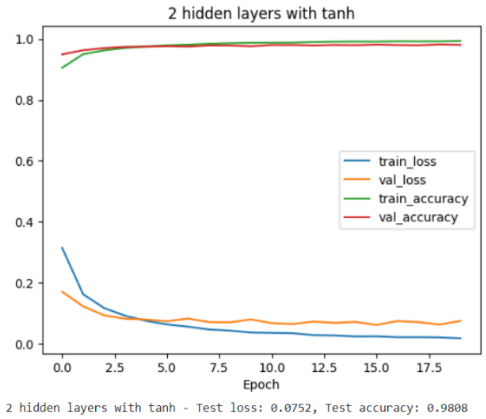
RAM Disk Gemini

```
loss, accuracy = model.evaluate(x_test.reshape(-1, 784), y_test, verbose=0)
print('{} - Test loss: {:.4f}, Test accuracy: {:.4f}'.format(name, loss, accuracy))
```

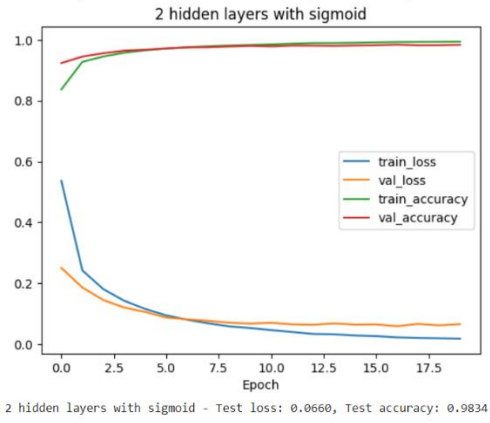



```
1 hidden layer with tanh - Test loss: 0.0634, Test accuracy: 0.9820
```

✓ 10m 36s completed at 5:56 PM

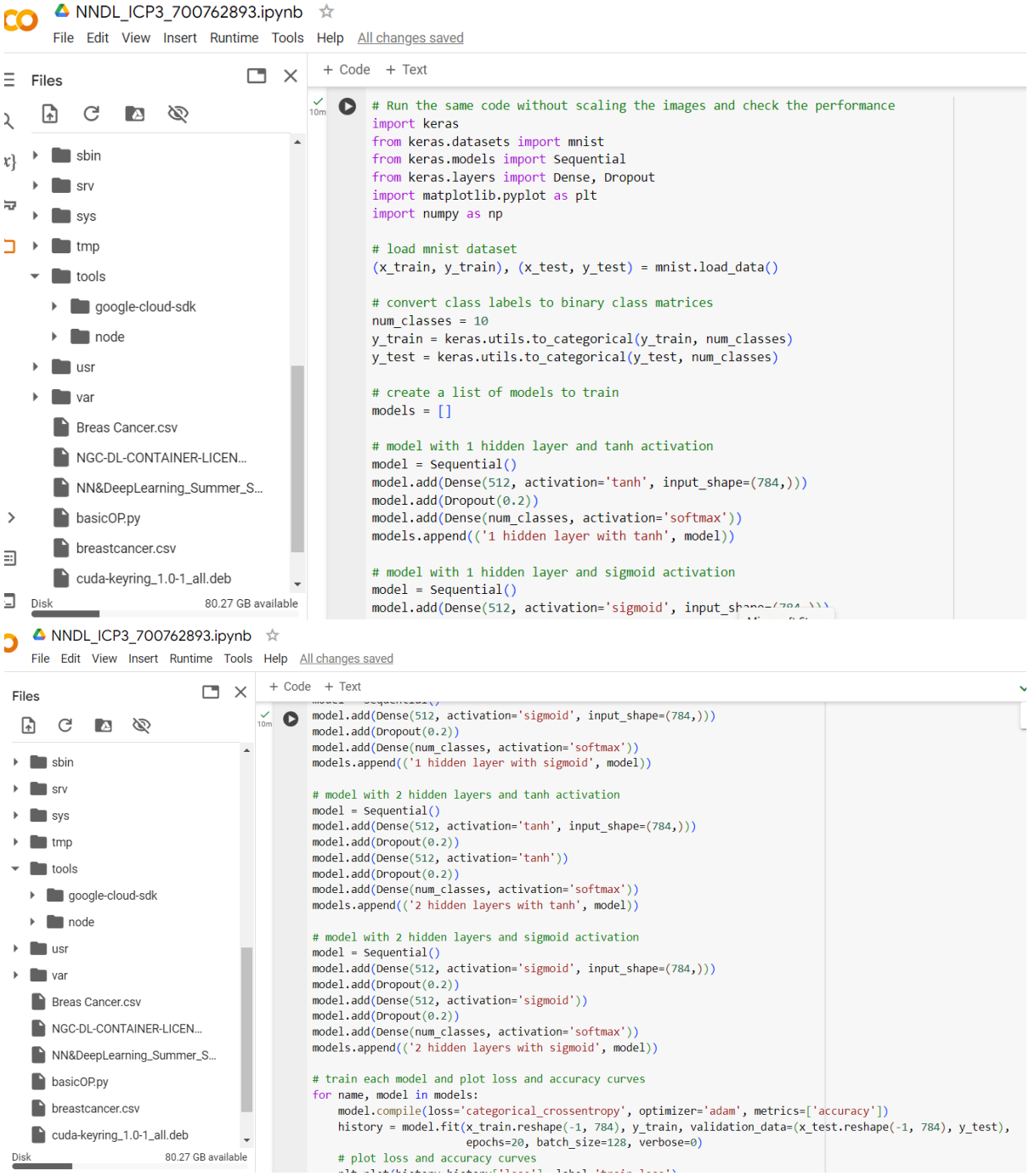



```
2 hidden layers with tanh - Test loss: 0.0752, Test accuracy: 0.9808
```




```
2 hidden layers with sigmoid - Test loss: 0.0660, Test accuracy: 0.9834
```

4. Run the same code without scaling the images and check the performance?



NNDL_ICP3_700762893.ipynb

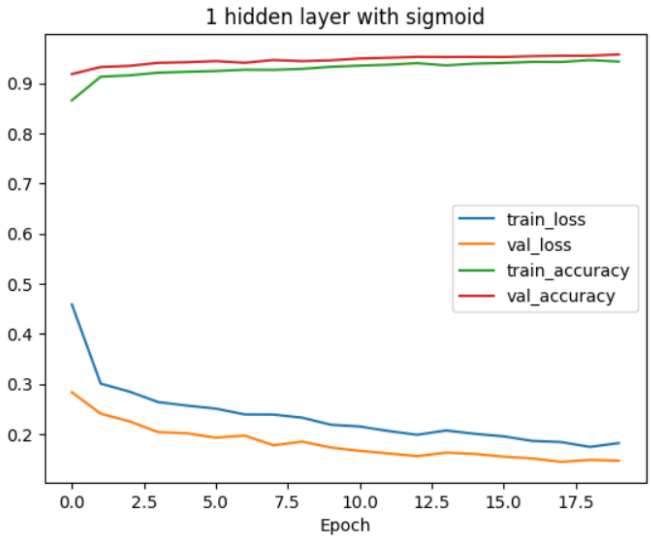
File Edit View Insert Runtime Tools Help All changes saved

Files

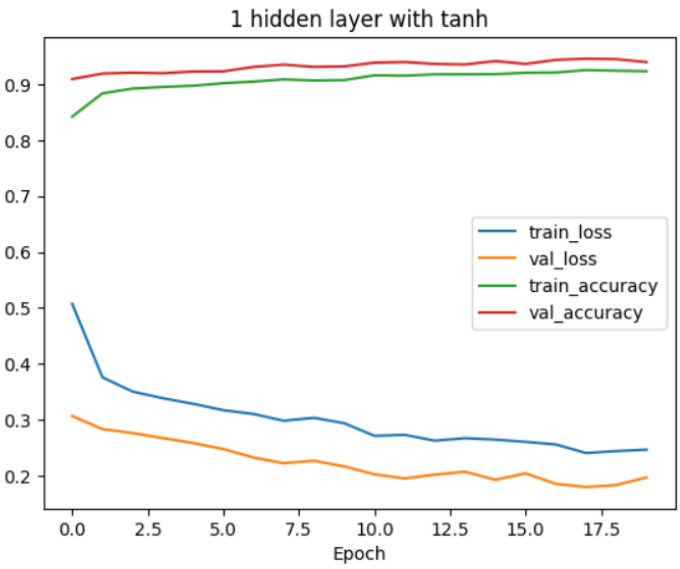
sbin
srv
sys
tmp
tools
google-cloud-sdk
node
usr
var
Breas Cancer.csv
NGC-DL-CONTAINER-LICEN...
NN&DeepLearning_Summer_S...
basicOP.py

+ Code + Text

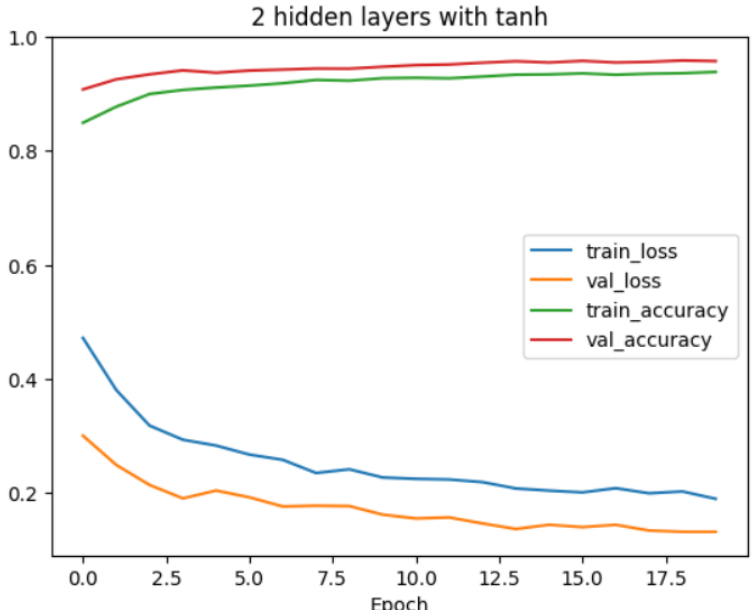
```
model.add(Dense(512, activation='sigmoid'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('2 hidden layers with sigmoid', model))

# train each model and plot loss and accuracy curves
for name, model in models:
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    history = model.fit(x_train.reshape(-1, 784), y_train, validation_data=(x_test.reshape(-1, 784), y_test),
                        epochs=20, batch_size=128, verbose=0)
    # plot loss and accuracy curves
    plt.plot(history.history['loss'], label='train_loss')
    plt.plot(history.history['val_loss'], label='val_loss')
    plt.plot(history.history['accuracy'], label='train_accuracy')
    plt.plot(history.history['val_accuracy'], label='val_accuracy')
    plt.title(name)
    plt.xlabel('Epoch')
    plt.legend()
    plt.show()

    # evaluate the model on test data
    loss, accuracy = model.evaluate(x_test.reshape(-1, 784), y_test, verbose=0)
    print('{} - Test loss: {:.4f}, Test accuracy: {:.4f}'.format(name, loss, accuracy))
```



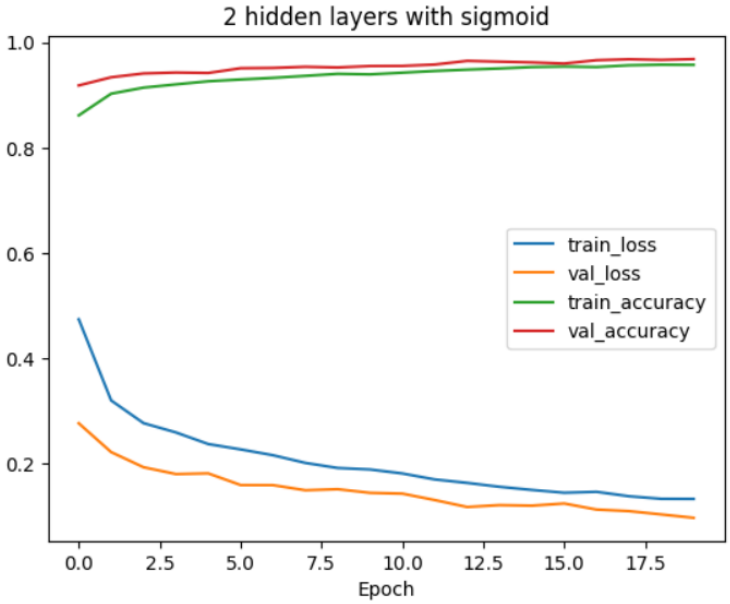

```
1 hidden layer with sigmoid - Test loss: 0.1469, Test accuracy: 0.9573
```




```
1 hidden layer with tanh - Test loss: 0.1967, Test accuracy: 0.9399
```



2 hidden layers with tanh - Test loss: 0.1324, Test accuracy: 0.9577



2 hidden layers with sigmoid - Test loss: 0.0966, Test accuracy: 0.9684