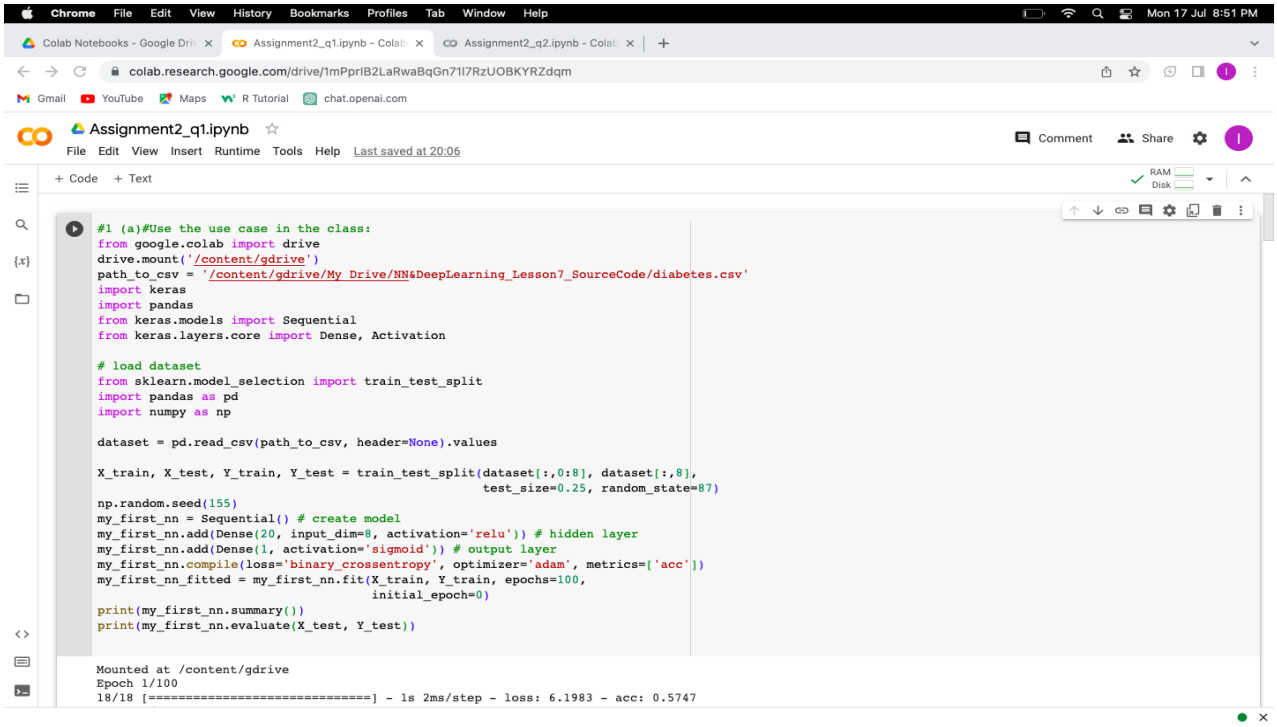
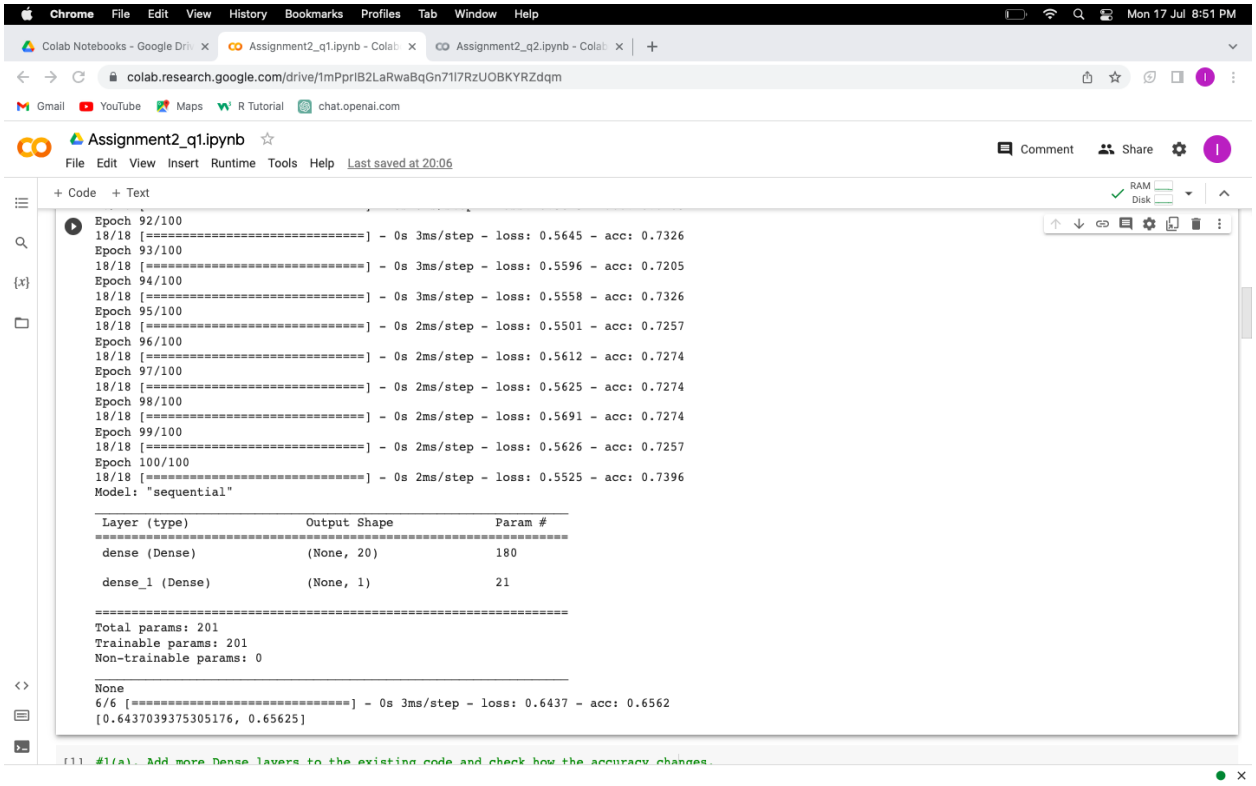
NNDL - Assignment - 6 ICP_Basics in Keras

Use Case Description: Predicting the diabetes disease
Programming elements: Keras Basics
In class programming:
1) Use the use case in the class:
a) Add more Dense layers to the existing code and check how the accuracy changes
Ans
Code before adding the dense layers is -



Accuracy of the model before adding the code is -

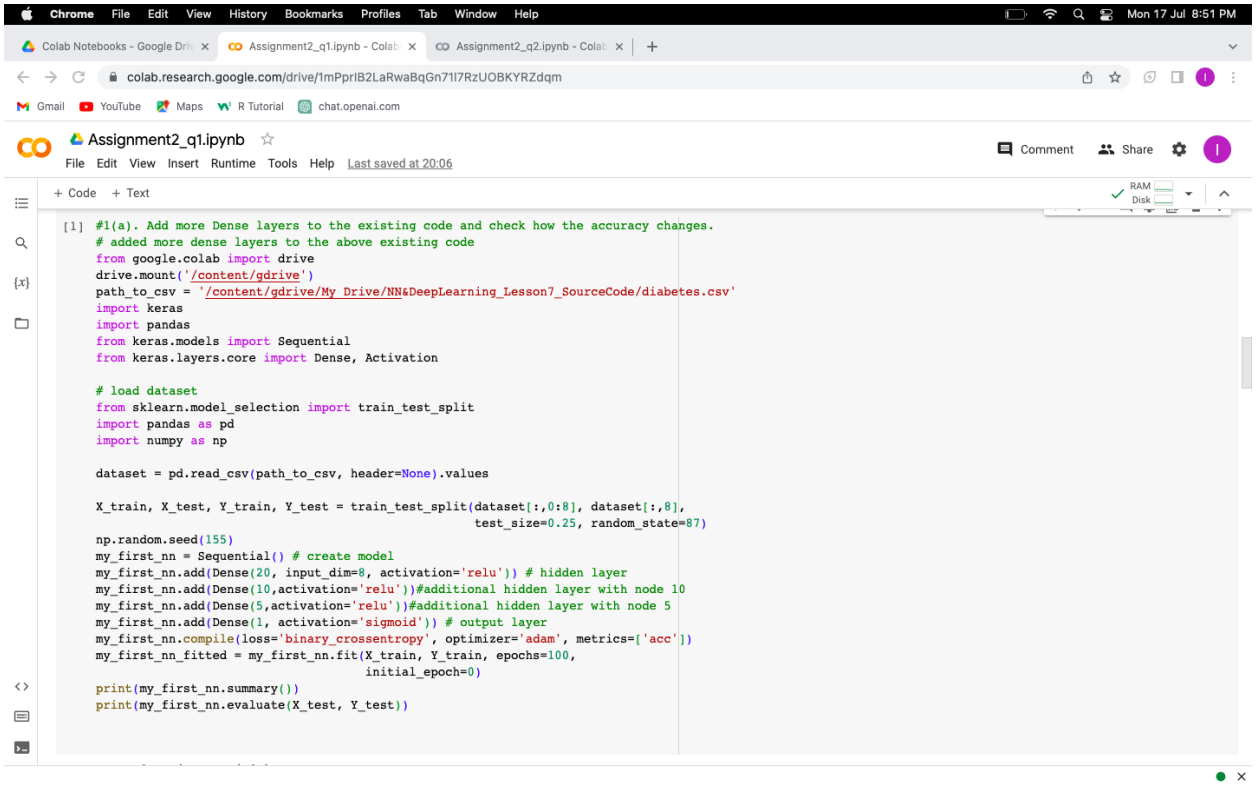


Adding more dense layers to the code -

1) We can easily add more dense layers to the existing code , we can simply add more dense layers to the sequential model.

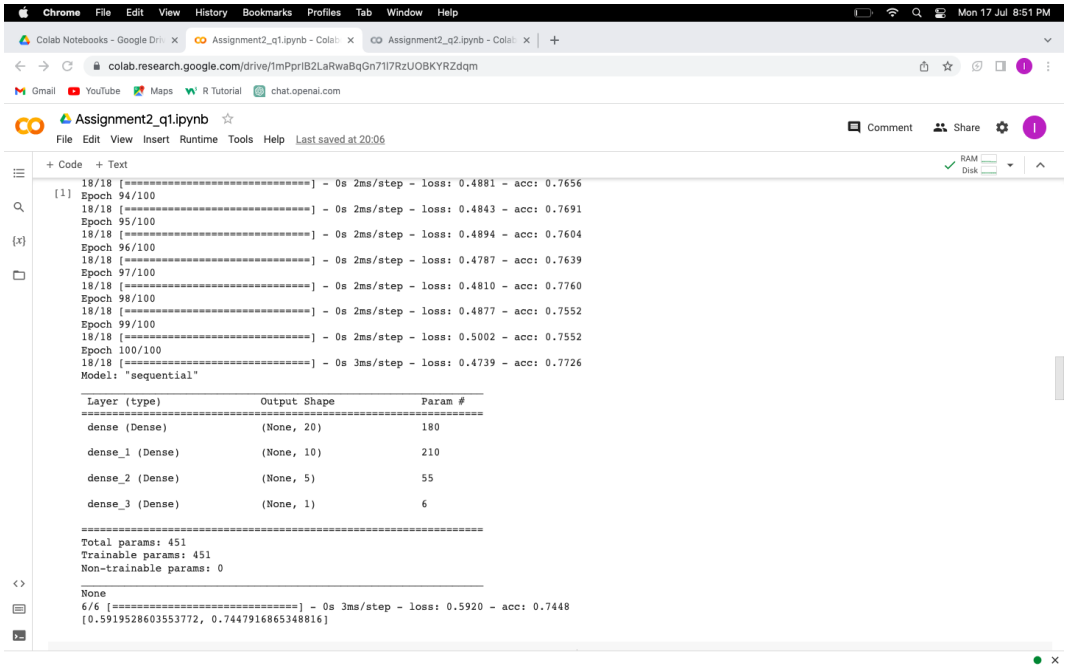
2) Here I have added two additional dense layers with 10 and 5 nodes.

Code after adding the dense layers is -



```
[1] #1(a). Add more Dense layers to the existing code and check how the accuracy changes.
    # added more dense layers to the above existing code
    from google.colab import drive
    drive.mount('/content/gdrive')
    path_to_csv = '/content/gdrive/My Drive/NN&DeepLearning_Lesson7_SourceCode/diabetes.csv'
    import keras
    import pandas
    from keras.models import Sequential
    from keras.layers.core import Dense, Activation

    # load dataset
    from sklearn.model_selection import train_test_split
    import pandas as pd
    import numpy as np

    dataset = pd.read_csv(path_to_csv, header=None).values

    X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,0:8], dataset[:,8],
                                                        test_size=0.25, random_state=87)
    np.random.seed(155)
    my_first_nn = Sequential() # create model
    my_first_nn.add(Dense(20, input_dim=8, activation='relu')) # hidden layer
    my_first_nn.add(Dense(10,activation='relu'))#additional hidden layer with node 10
    my_first_nn.add(Dense(5,activation='relu'))#additional hidden layer with node 5
    my_first_nn.add(Dense(1, activation='sigmoid')) # output layer
    my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
    my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=100,
                                         initial_epoch=0)
    print(my_first_nn.summary())
    print(my_first_nn.evaluate(X_test, Y_test))
```

Accuracy of the model after adding the additional dense layers is



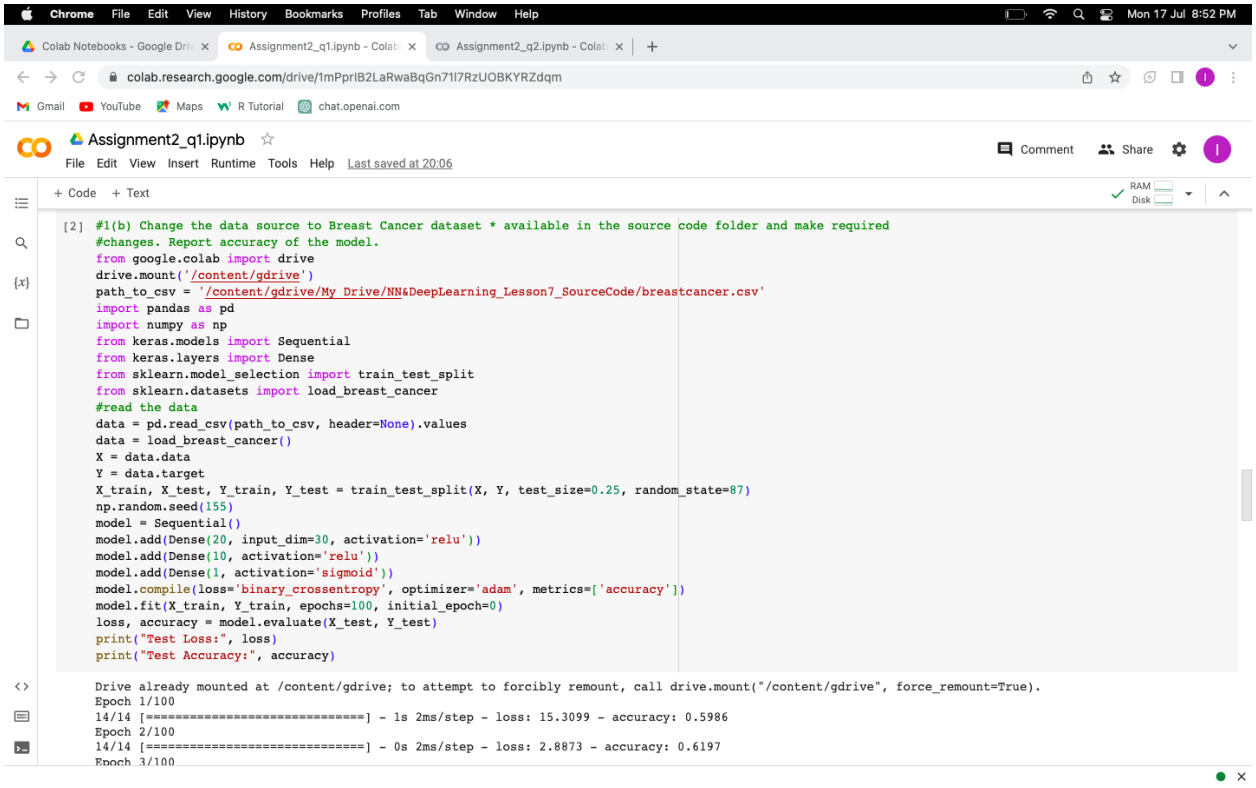
```
[1] 18/18 [==============================] - 0s 2ms/step - loss: 0.4881 - acc: 0.7656
    Epoch 94/100
    18/18 [==============================] - 0s 2ms/step - loss: 0.4843 - acc: 0.7691
    Epoch 95/100
    18/18 [==============================] - 0s 2ms/step - loss: 0.4894 - acc: 0.7604
    Epoch 96/100
    18/18 [==============================] - 0s 2ms/step - loss: 0.4787 - acc: 0.7639
    Epoch 97/100
    18/18 [==============================] - 0s 2ms/step - loss: 0.4810 - acc: 0.7760
    Epoch 98/100
    18/18 [==============================] - 0s 2ms/step - loss: 0.4877 - acc: 0.7552
    Epoch 99/100
    18/18 [==============================] - 0s 2ms/step - loss: 0.5002 - acc: 0.7552
    Epoch 100/100
    18/18 [==============================] - 0s 3ms/step - loss: 0.4739 - acc: 0.7726
    Model: "sequential"
    _________________________________________________________________
     Layer (type)                Output Shape              Param #
    =================================================================
     dense (Dense)               (None, 20)                180

     dense_1 (Dense)             (None, 10)                210

     dense_2 (Dense)             (None, 5)                 55

     dense_3 (Dense)             (None, 1)                 6

    =================================================================
    Total params: 451
    Trainable params: 451
    Non-trainable params: 0
    _________________________________________________________________
    None
    6/6 [==============================] - 0s 3ms/step - loss: 0.5920 - acc: 0.7448
    [0.5919528603553772, 0.7447916865348816]
```

Accuracy of the model increased from 0.6562 to 0.7448 after adding 2 more dense layers.Therefore, by adding additional layers the model has more parameters to learn from the training data which may lead to better accuracy.

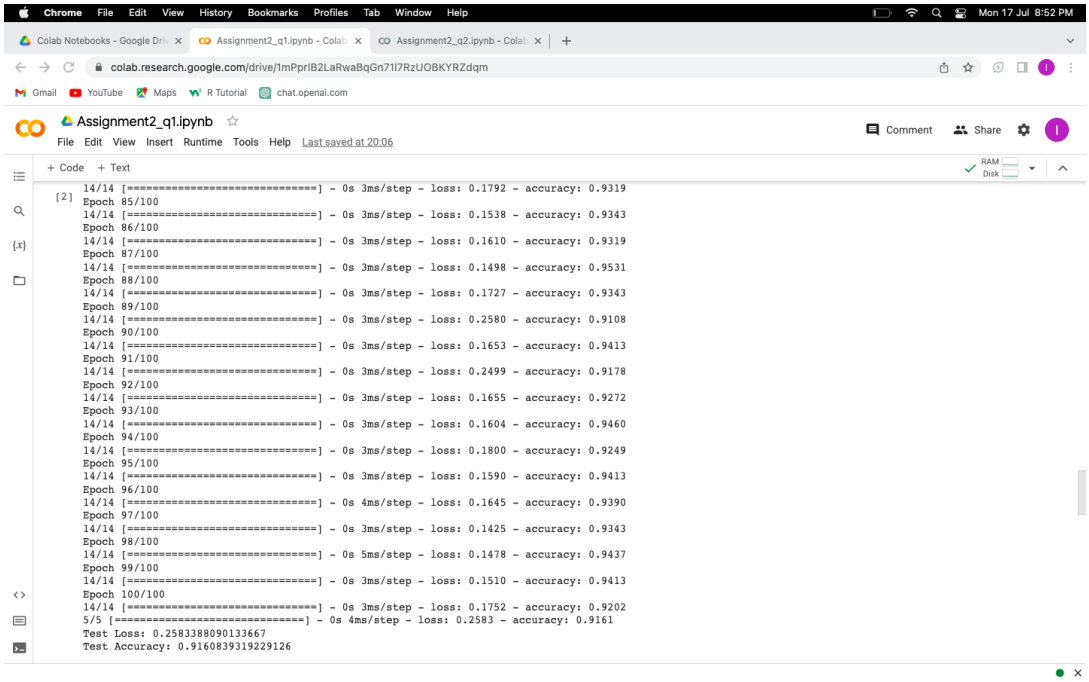
b) Change the data source to Breast Cancer dataset * available in the source code folder and make required changes. Report accuracy of the model

Ans)

Code change after changing the data source to Breast Cancer dataset



Accuracy of the model after changing the data source to the breast cancer :

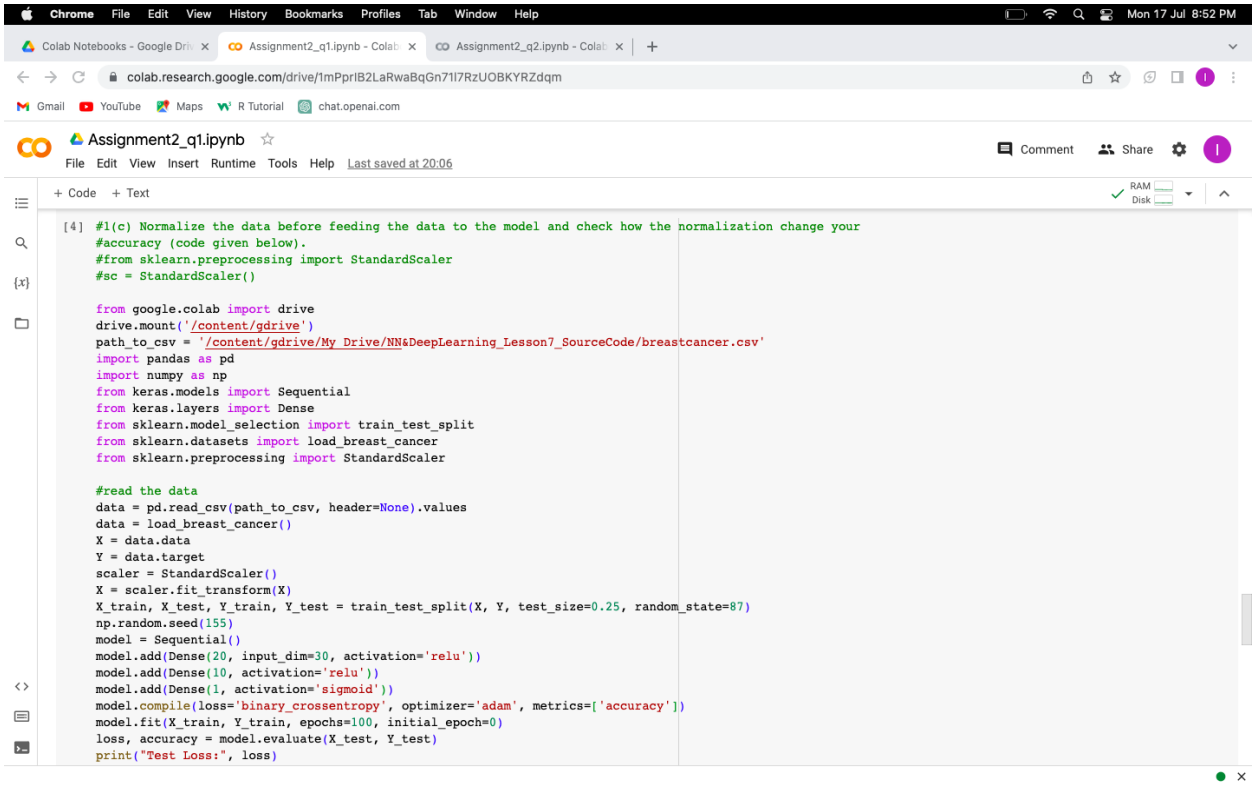


c) Normalize the data before feeding the data to the model and check how the normalization change your accuracy (code given below).

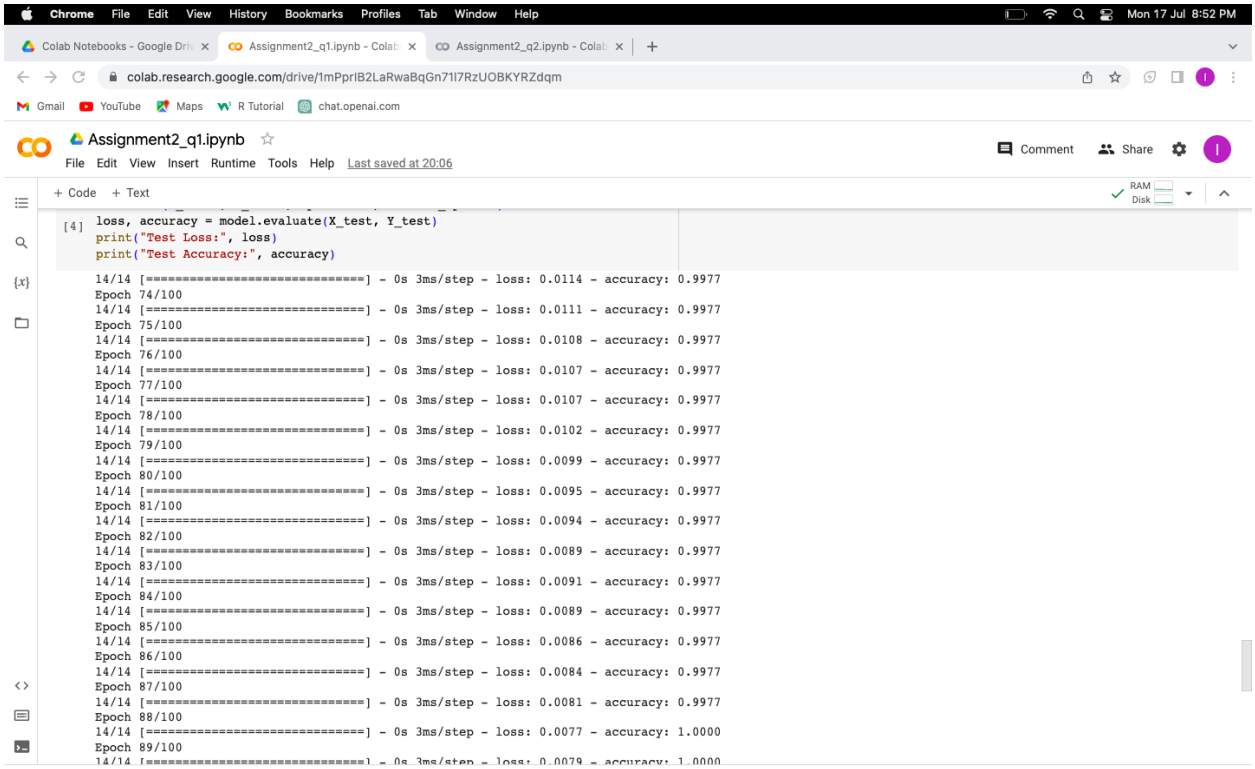
from sklearn.preprocessing

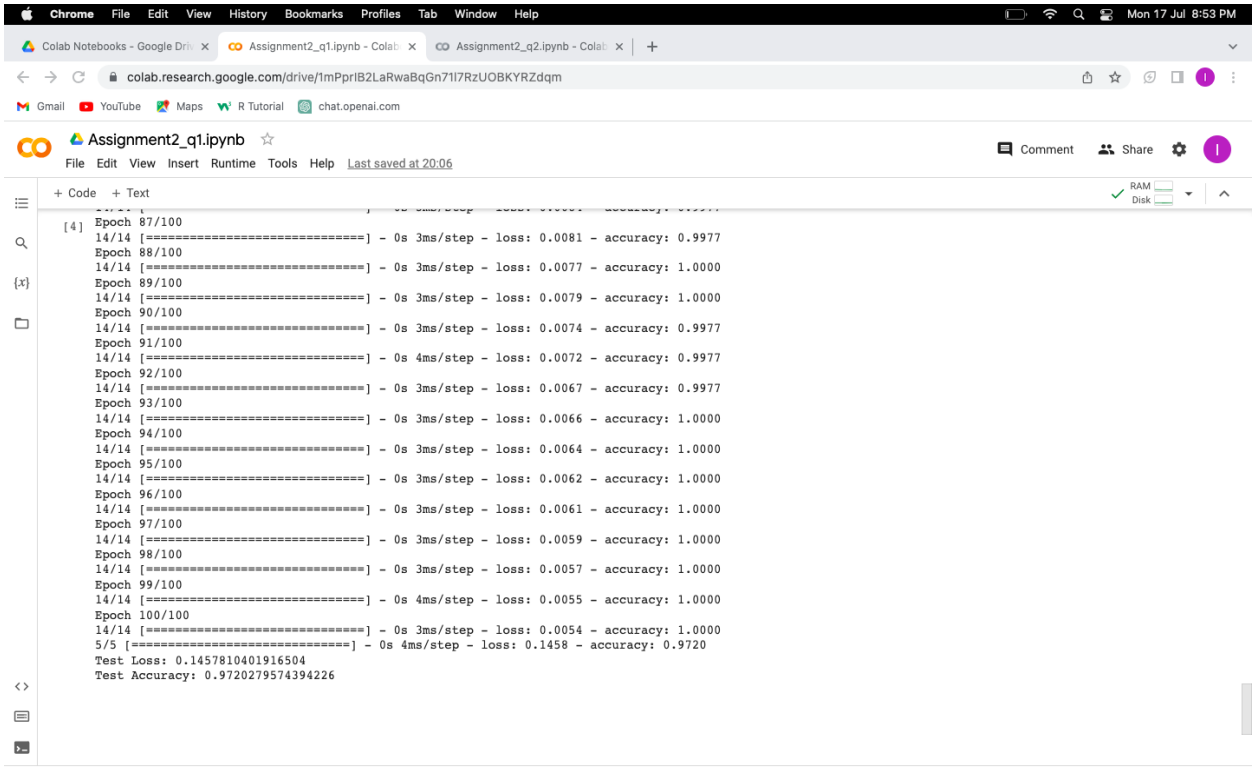
import StandardScaler sc = StandardScaler()

Ans)



Output:

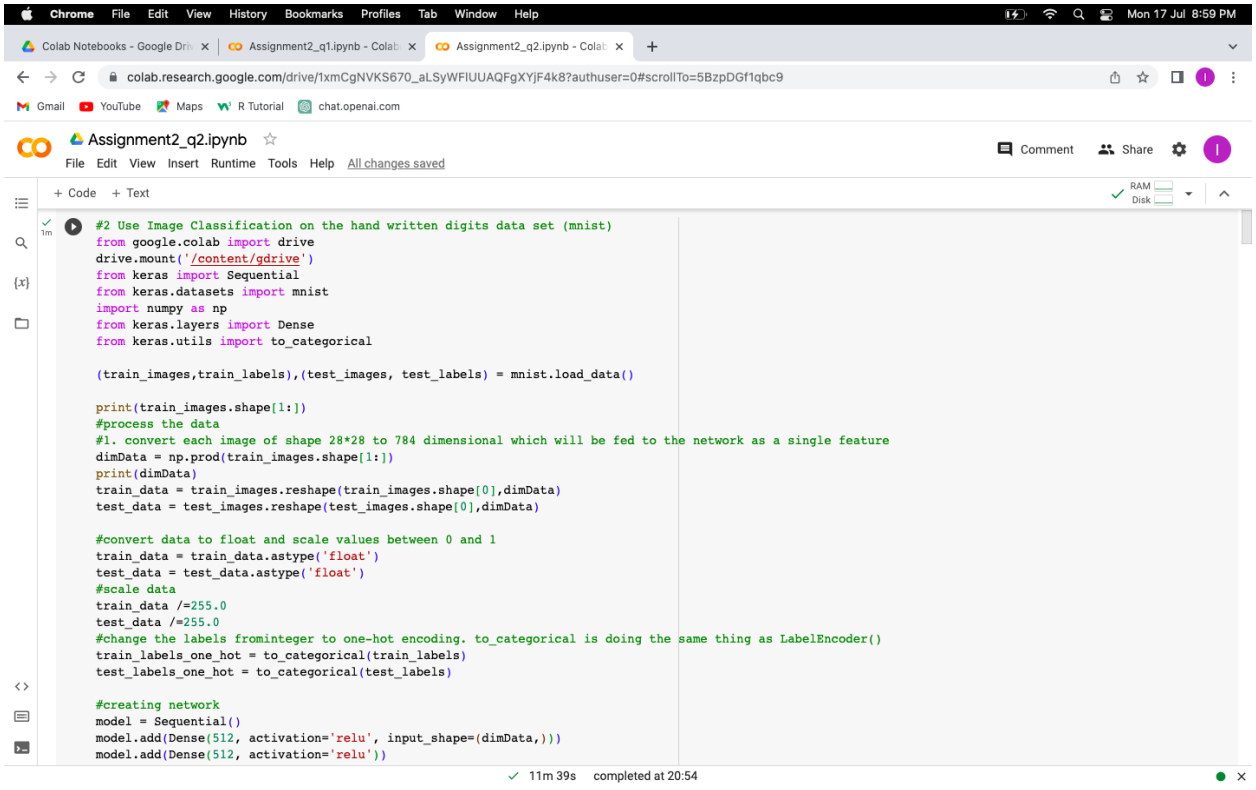

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 8:52 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1mPprIB2LaRwaBqGn71l7RzUOBKYRZdqm
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q1.ipynb
File Edit View Insert Runtime Tools Help Last saved at 20:06
Comment Share
+ Code + Text
RAM Disk
[4] loss, accuracy = model.evaluate(X_test, Y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
14/14 [==========] - 0s 3ms/step - loss: 0.0114 - accuracy: 0.9977
Epoch 74/100
14/14 [==========] - 0s 3ms/step - loss: 0.0111 - accuracy: 0.9977
Epoch 75/100
14/14 [==========] - 0s 3ms/step - loss: 0.0108 - accuracy: 0.9977
Epoch 76/100
14/14 [==========] - 0s 3ms/step - loss: 0.0107 - accuracy: 0.9977
Epoch 77/100
14/14 [==========] - 0s 3ms/step - loss: 0.0107 - accuracy: 0.9977
Epoch 78/100
14/14 [==========] - 0s 3ms/step - loss: 0.0102 - accuracy: 0.9977
Epoch 79/100
14/14 [==========] - 0s 3ms/step - loss: 0.0099 - accuracy: 0.9977
Epoch 80/100
14/14 [==========] - 0s 3ms/step - loss: 0.0095 - accuracy: 0.9977
Epoch 81/100
14/14 [==========] - 0s 3ms/step - loss: 0.0094 - accuracy: 0.9977
Epoch 82/100
14/14 [==========] - 0s 3ms/step - loss: 0.0089 - accuracy: 0.9977
Epoch 83/100
14/14 [==========] - 0s 3ms/step - loss: 0.0091 - accuracy: 0.9977
Epoch 84/100
14/14 [==========] - 0s 3ms/step - loss: 0.0089 - accuracy: 0.9977
Epoch 85/100
14/14 [==========] - 0s 3ms/step - loss: 0.0086 - accuracy: 0.9977
Epoch 86/100
14/14 [==========] - 0s 3ms/step - loss: 0.0084 - accuracy: 0.9977
Epoch 87/100
14/14 [==========] - 0s 3ms/step - loss: 0.0081 - accuracy: 0.9977
Epoch 88/100
14/14 [==========] - 0s 3ms/step - loss: 0.0077 - accuracy: 1.0000
Epoch 89/100


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 8:53 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1mPprIB2LaRwaBqGn71l7RzUOBKYRZdqm
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q1.ipynb
File Edit View Insert Runtime Tools Help Last saved at 20:06
Comment Share
+ Code + Text
RAM Disk
[4] Epoch 87/100
14/14 [==========] - 0s 3ms/step - loss: 0.0081 - accuracy: 0.9977
Epoch 88/100
14/14 [==========] - 0s 3ms/step - loss: 0.0077 - accuracy: 1.0000
Epoch 89/100
14/14 [==========] - 0s 3ms/step - loss: 0.0079 - accuracy: 1.0000
Epoch 90/100
14/14 [==========] - 0s 3ms/step - loss: 0.0074 - accuracy: 0.9977
Epoch 91/100
14/14 [==========] - 0s 4ms/step - loss: 0.0072 - accuracy: 0.9977
Epoch 92/100
14/14 [==========] - 0s 3ms/step - loss: 0.0067 - accuracy: 0.9977
Epoch 93/100
14/14 [==========] - 0s 3ms/step - loss: 0.0066 - accuracy: 1.0000
Epoch 94/100
14/14 [==========] - 0s 3ms/step - loss: 0.0064 - accuracy: 1.0000
Epoch 95/100
14/14 [==========] - 0s 3ms/step - loss: 0.0062 - accuracy: 1.0000
Epoch 96/100
14/14 [==========] - 0s 3ms/step - loss: 0.0061 - accuracy: 1.0000
Epoch 97/100
14/14 [==========] - 0s 3ms/step - loss: 0.0059 - accuracy: 1.0000
Epoch 98/100
14/14 [==========] - 0s 3ms/step - loss: 0.0057 - accuracy: 1.0000
Epoch 99/100
14/14 [==========] - 0s 4ms/step - loss: 0.0055 - accuracy: 1.0000
Epoch 100/100
14/14 [==========] - 0s 3ms/step - loss: 0.0054 - accuracy: 1.0000
5/5 [==========] - 0s 4ms/step - loss: 0.1458 - accuracy: 0.9720
Test Loss: 0.1457810401916504
Test Accuracy: 0.9720279574394226

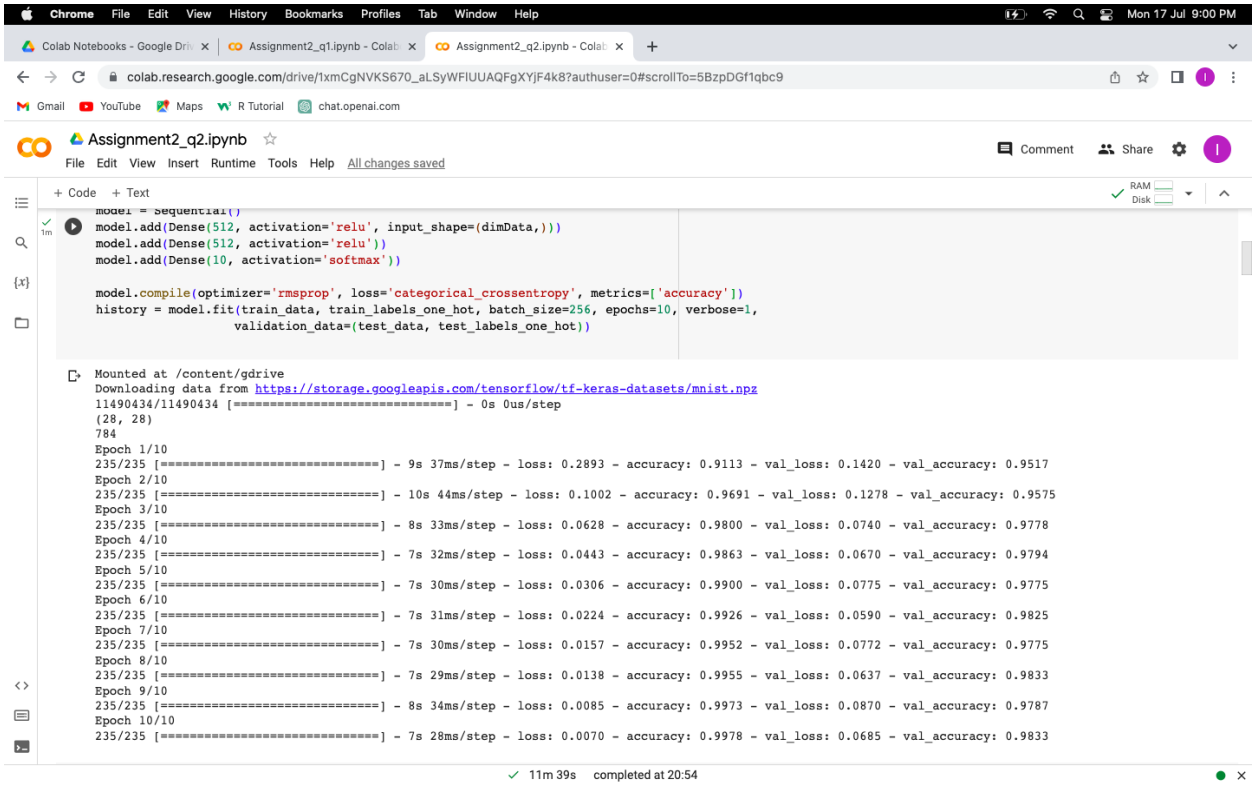
2)Use Image Classification on the handwritten digits data set (mnist)

Ans)

Code for the above is

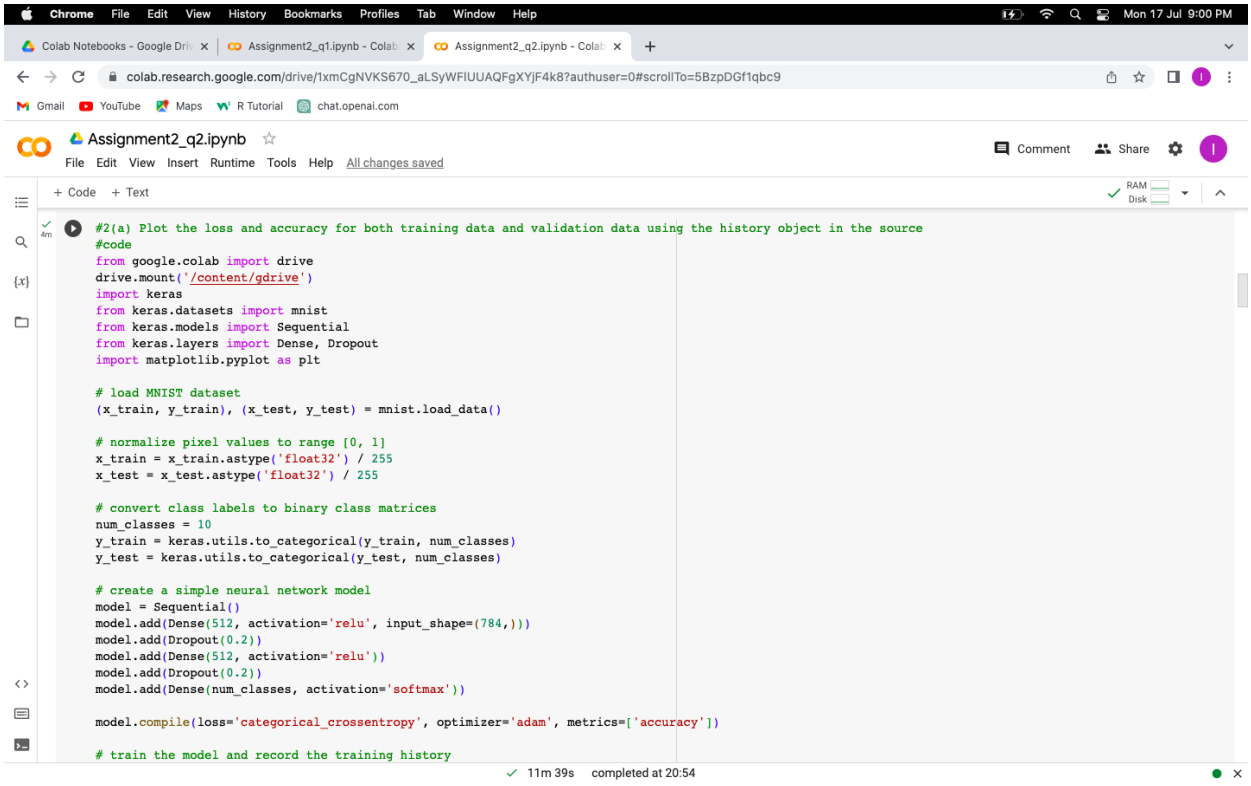


Output:

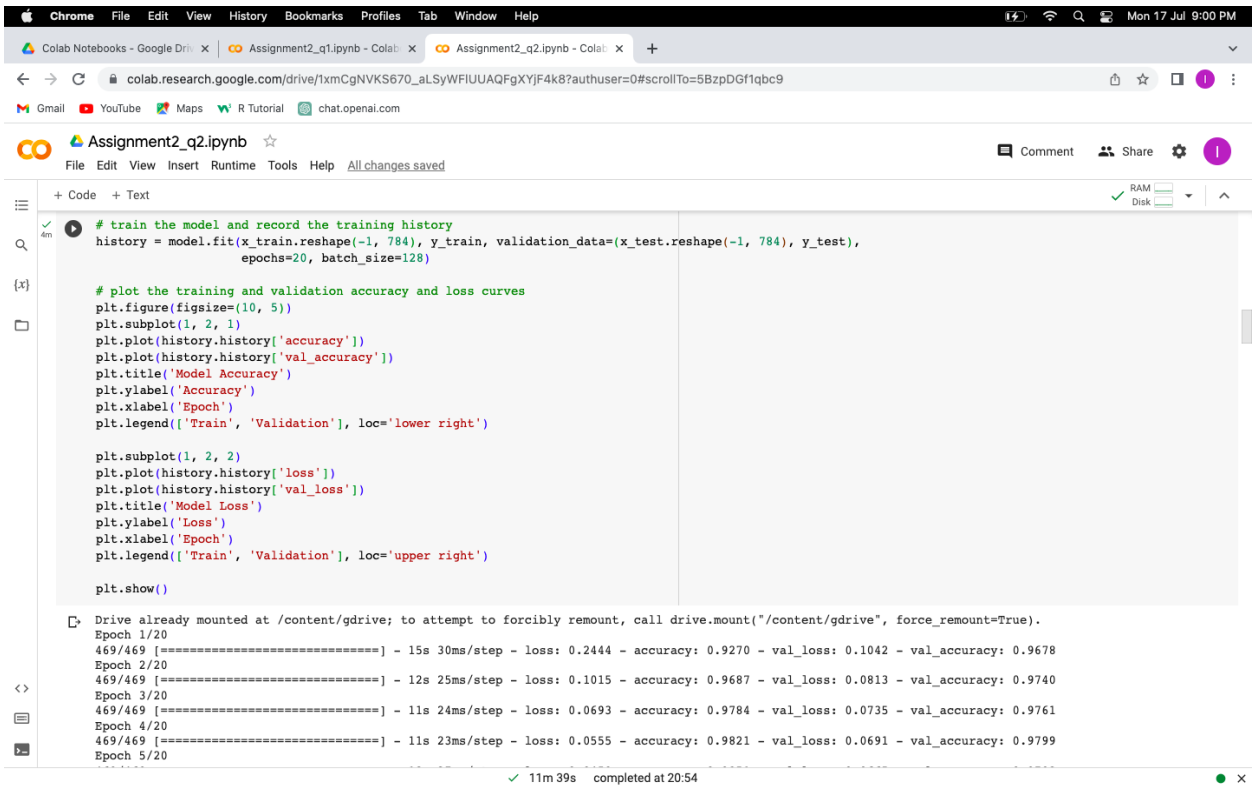


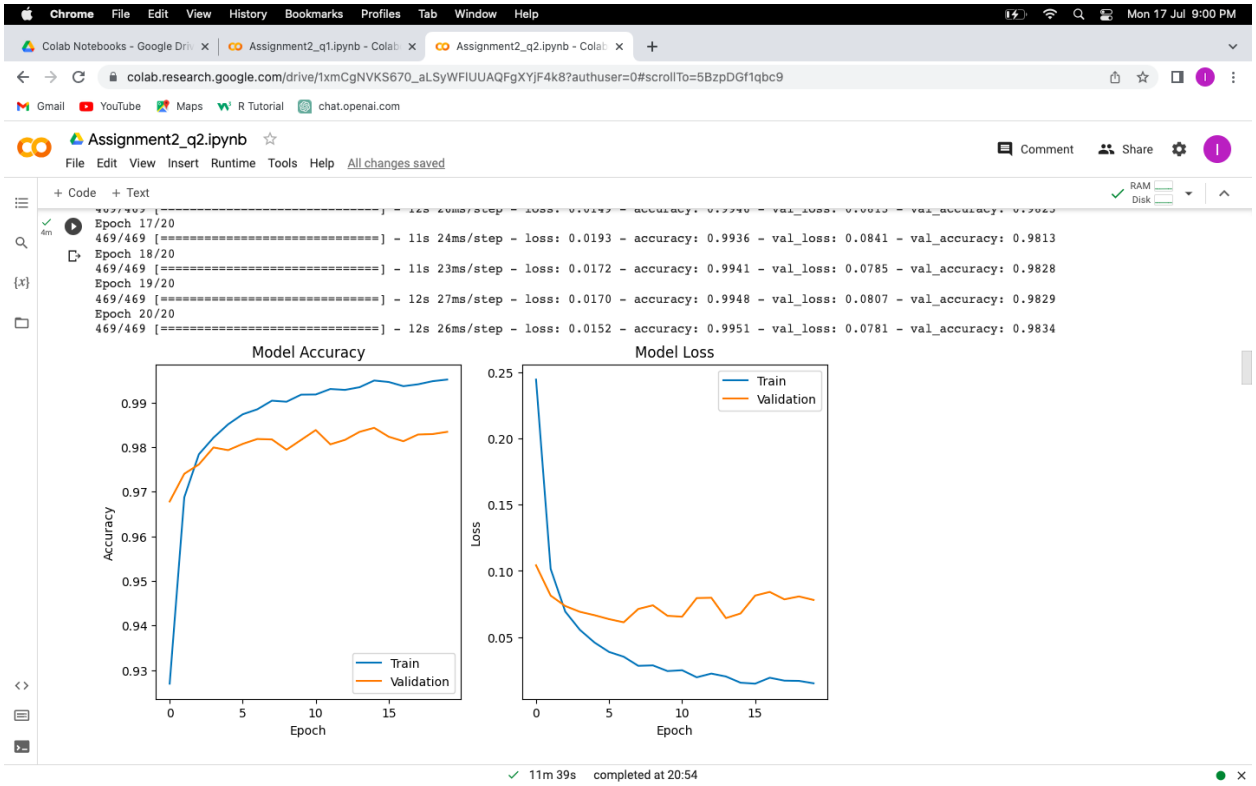
a). Plot the loss and accuracy for both training data and validation data using the history object in the source code.
Ans:

Code is

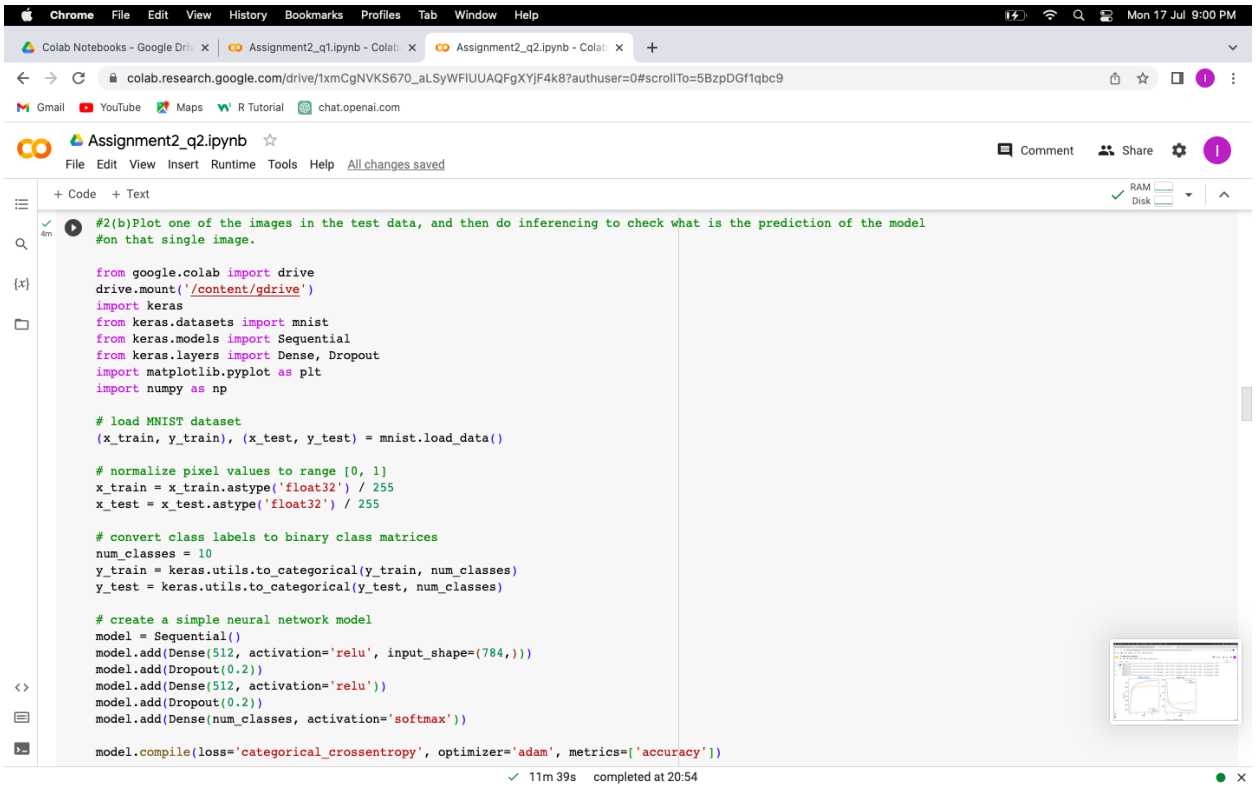

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 9:00 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
#2(a) Plot the loss and accuracy for both training data and validation data using the history object in the source
#code
from google.colab import drive
drive.mount('/content/gdrive')
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
import matplotlib.pyplot as plt
# load MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()
# normalize pixel values to range [0, 1]
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255
# convert class labels to binary class matrices
num_classes = 10
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
# create a simple neural network model
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# train the model and record the training history
11m 39s completed at 20:54

Output:

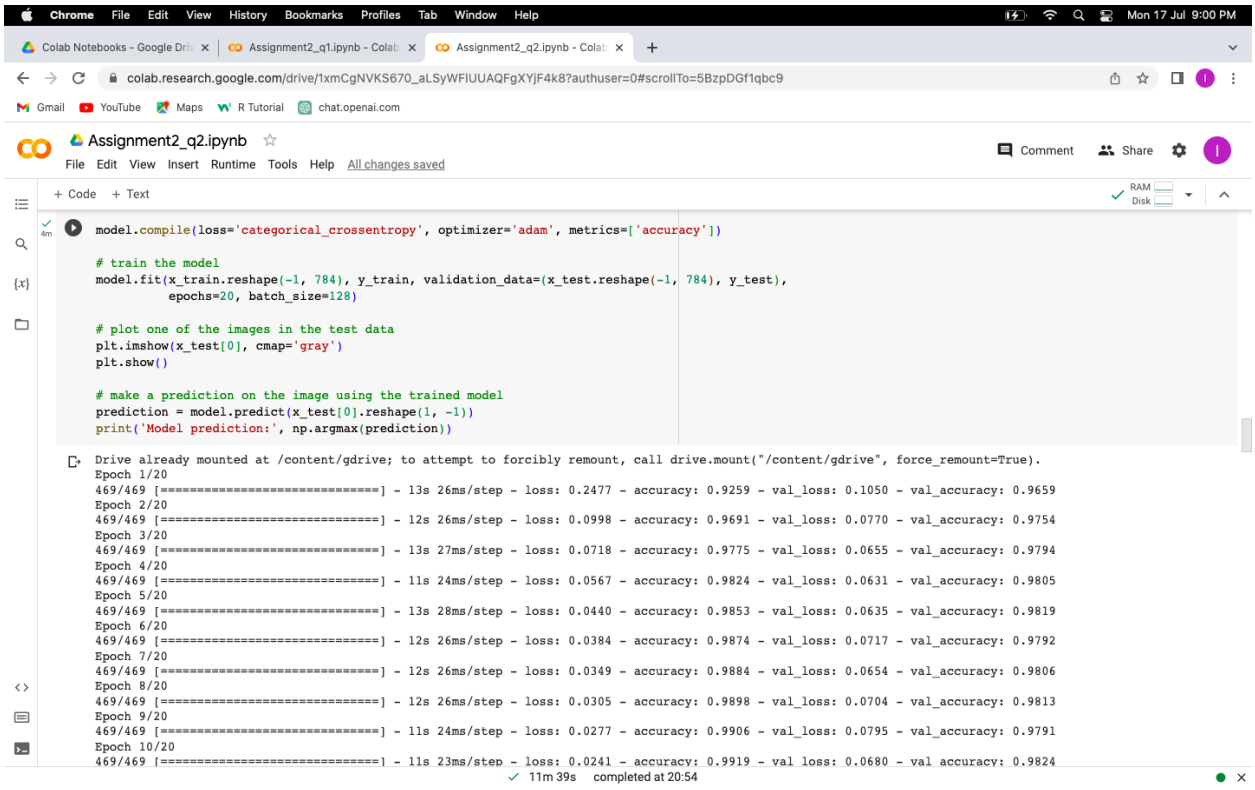

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 9:00 PM
colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9
Assignment2_q2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
# train the model and record the training history
history = model.fit(x_train.reshape(-1, 784), y_train, validation_data=(x_test.reshape(-1, 784), y_test),
epochs=20, batch_size=128)
# plot the training and validation accuracy and loss curves
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='lower right')
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
Epoch 1/20
469/469 [==========] - 15s 30ms/step - loss: 0.2444 - accuracy: 0.9270 - val_loss: 0.1042 - val_accuracy: 0.9678
Epoch 2/20
469/469 [==========] - 12s 25ms/step - loss: 0.1015 - accuracy: 0.9687 - val_loss: 0.0813 - val_accuracy: 0.9740
Epoch 3/20
469/469 [==========] - 11s 24ms/step - loss: 0.0693 - accuracy: 0.9784 - val_loss: 0.0735 - val_accuracy: 0.9761
Epoch 4/20
469/469 [==========] - 11s 23ms/step - loss: 0.0555 - accuracy: 0.9821 - val_loss: 0.0691 - val_accuracy: 0.9799
Epoch 5/20
11m 39s completed at 20:54

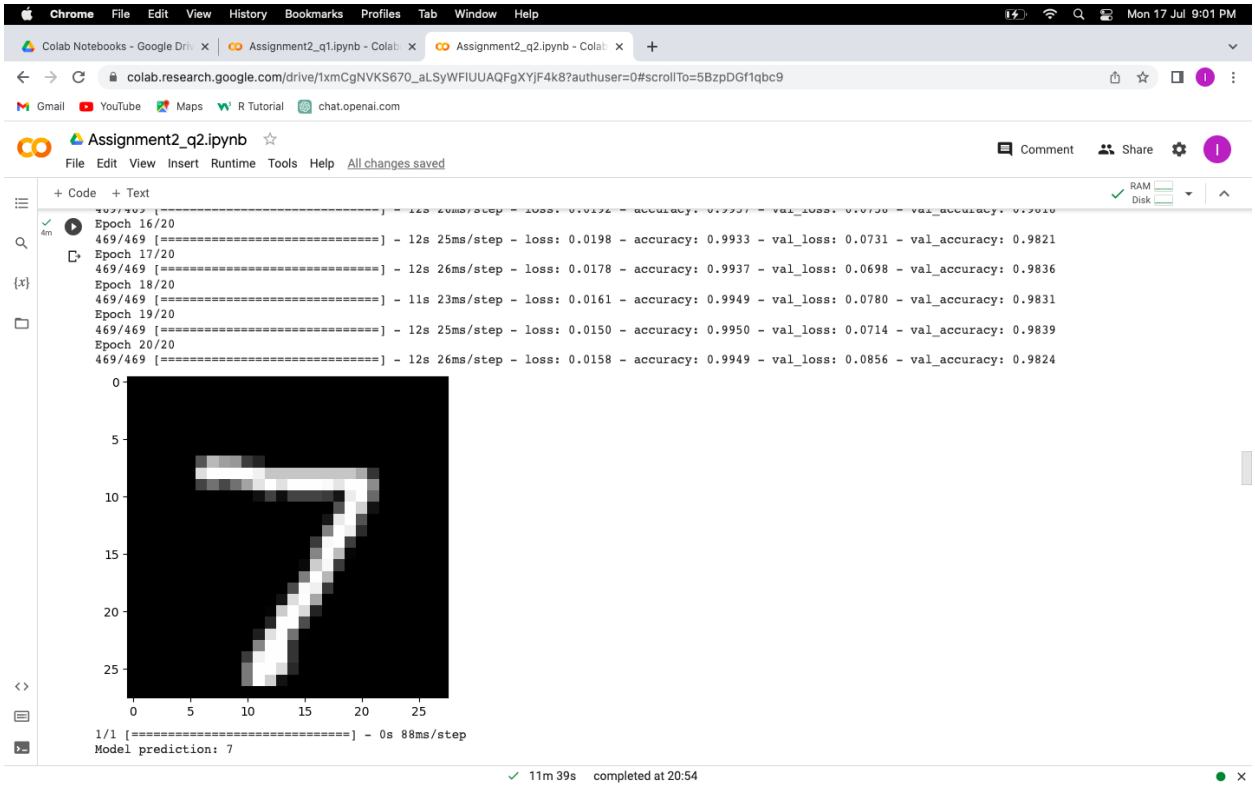


b). Plot one of the images in the test data, and then do inferencing to check what is the prediction of the model on that single image.



Output:

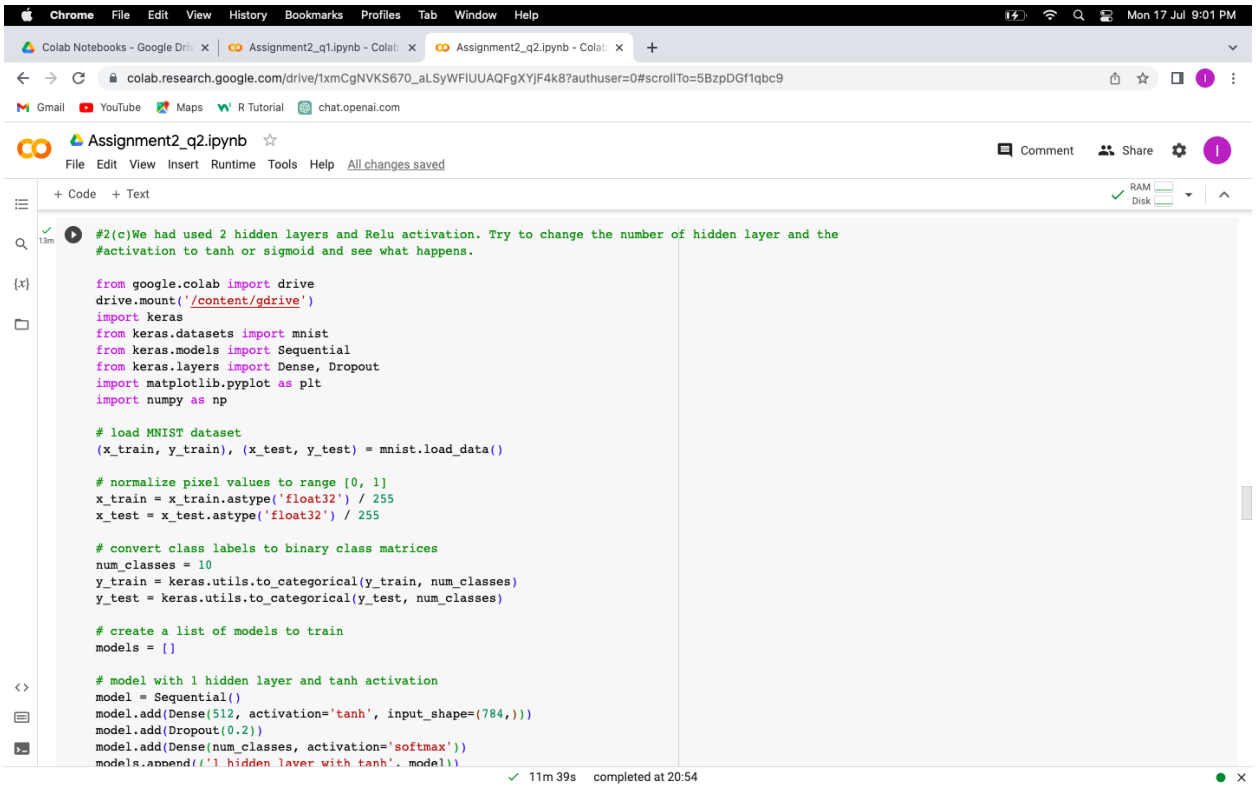

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 9:00 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# train the model
model.fit(x_train.reshape(-1, 784), y_train, validation_data=(x_test.reshape(-1, 784), y_test),
epochs=20, batch_size=128)
# plot one of the images in the test data
plt.imshow(x_test[0], cmap='gray')
plt.show()
# make a prediction on the image using the trained model
prediction = model.predict(x_test[0].reshape(1, -1))
print('Model prediction:', np.argmax(prediction))
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
Epoch 1/20
469/469 [==============================] - 13s 26ms/step - loss: 0.2477 - accuracy: 0.9259 - val_loss: 0.1050 - val_accuracy: 0.9659
Epoch 2/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0998 - accuracy: 0.9691 - val_loss: 0.0770 - val_accuracy: 0.9754
Epoch 3/20
469/469 [==============================] - 13s 27ms/step - loss: 0.0718 - accuracy: 0.9775 - val_loss: 0.0655 - val_accuracy: 0.9794
Epoch 4/20
469/469 [==============================] - 11s 24ms/step - loss: 0.0567 - accuracy: 0.9824 - val_loss: 0.0631 - val_accuracy: 0.9805
Epoch 5/20
469/469 [==============================] - 13s 28ms/step - loss: 0.0440 - accuracy: 0.9853 - val_loss: 0.0635 - val_accuracy: 0.9819
Epoch 6/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0384 - accuracy: 0.9874 - val_loss: 0.0717 - val_accuracy: 0.9792
Epoch 7/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0349 - accuracy: 0.9884 - val_loss: 0.0654 - val_accuracy: 0.9806
Epoch 8/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0305 - accuracy: 0.9898 - val_loss: 0.0704 - val_accuracy: 0.9813
Epoch 9/20
469/469 [==============================] - 11s 24ms/step - loss: 0.0277 - accuracy: 0.9906 - val_loss: 0.0795 - val_accuracy: 0.9791
Epoch 10/20
469/469 [==============================] - 11s 23ms/step - loss: 0.0241 - accuracy: 0.9919 - val_loss: 0.0680 - val_accuracy: 0.9824
11m 39s completed at 20:54


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 9:01 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Epoch 16/20
469/469 [==============================] - 12s 25ms/step - loss: 0.0198 - accuracy: 0.9933 - val_loss: 0.0731 - val_accuracy: 0.9821
Epoch 17/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0178 - accuracy: 0.9937 - val_loss: 0.0698 - val_accuracy: 0.9836
Epoch 18/20
469/469 [==============================] - 11s 23ms/step - loss: 0.0161 - accuracy: 0.9949 - val_loss: 0.0780 - val_accuracy: 0.9831
Epoch 19/20
469/469 [==============================] - 12s 25ms/step - loss: 0.0150 - accuracy: 0.9950 - val_loss: 0.0714 - val_accuracy: 0.9839
Epoch 20/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0158 - accuracy: 0.9949 - val_loss: 0.0856 - val_accuracy: 0.9824
0
5
10
15
20
25
0 5 10 15 20 25
1/1 [==============================] - 0s 88ms/step
Model prediction: 7
11m 39s completed at 20:54

c.)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.

Ans)

Code

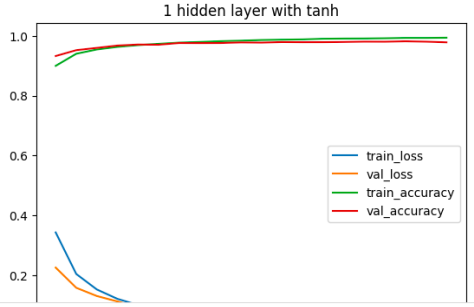


```
    model.add(Dense(num_classes, activation='softmax'))
    models.append(('1 hidden layer with tanh', model))

    # model with 1 hidden layer and sigmoid activation
    model = Sequential()
    model.add(Dense(512, activation='sigmoid', input_shape=(784,)))
    model.add(Dropout(0.2))
    model.add(Dense(num_classes, activation='softmax'))
    models.append(('1 hidden layer with sigmoid', model))

    # model with 2 hidden layers and tanh activation
    model = Sequential()
    model.add(Dense(512, activation='tanh', input_shape=(784,)))
    model.add(Dropout(0.2))
    model.add(Dense(512, activation='tanh'))
    model.add(Dropout(0.2))
    model.add(Dense(num_classes, activation='softmax'))
    models.append(('2 hidden layers with tanh', model))

    # model with 2 hidden layers and sigmoid activation
    model = Sequential()
    model.add(Dense(512, activation='sigmoid', input_shape=(784,)))
    model.add(Dropout(0.2))
    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
```

Output:

```
        # 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))
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```



Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 17 Jul 9:01 PM

Colab Notebooks - Google Dri | Assignment2_q1.ipynb - Colab | Assignment2_q2.ipynb - Colab

colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9

Gmail YouTube Maps R Tutorial chat.openai.com

Assignment2_q2.ipynb

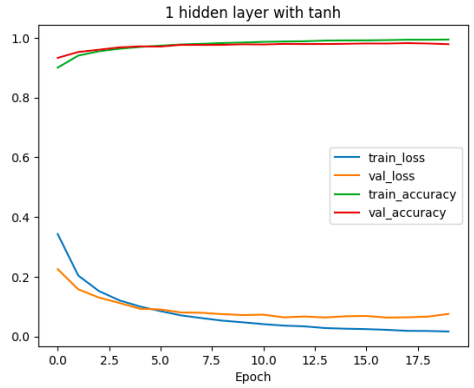
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

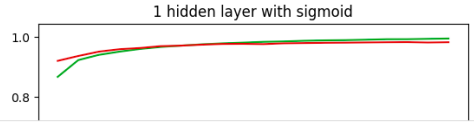
+ Code + Text

RAM Disk

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```




```
1 hidden layer with tanh - Test loss: 0.0758, Test accuracy: 0.9789
```



11m 39s completed at 20:54

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 17 Jul 9:02 PM

Colab Notebooks - Google Dri | Assignment2_q1.ipynb - Colab | Assignment2_q2.ipynb - Colab

colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9

Gmail YouTube Maps R Tutorial chat.openai.com

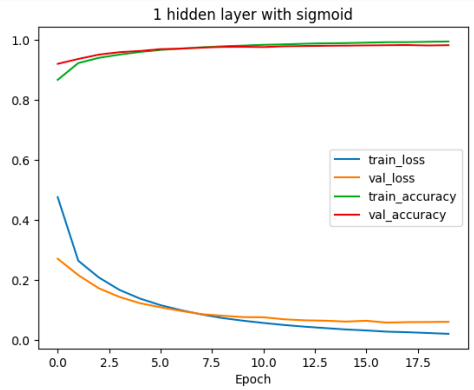
Assignment2_q2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

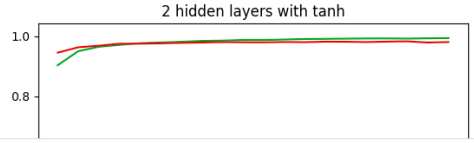
Comment Share

+ Code + Text

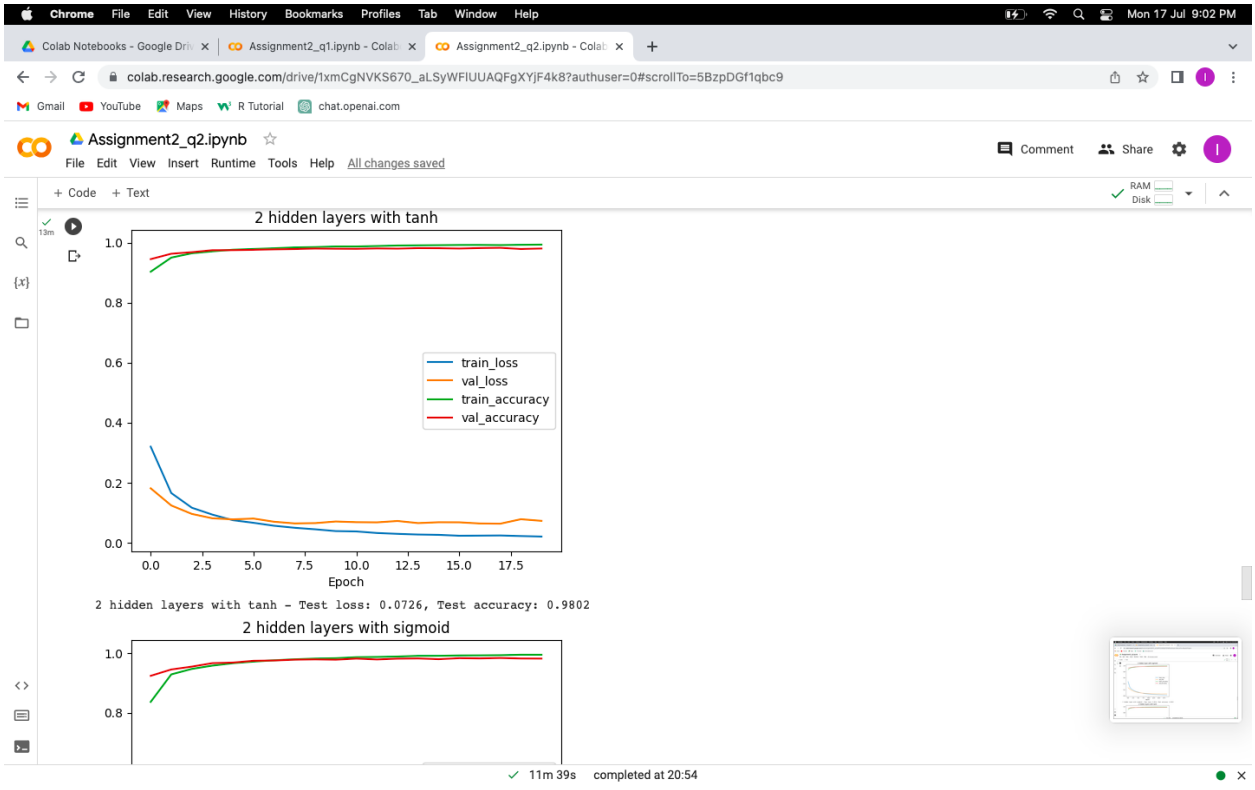
RAM Disk

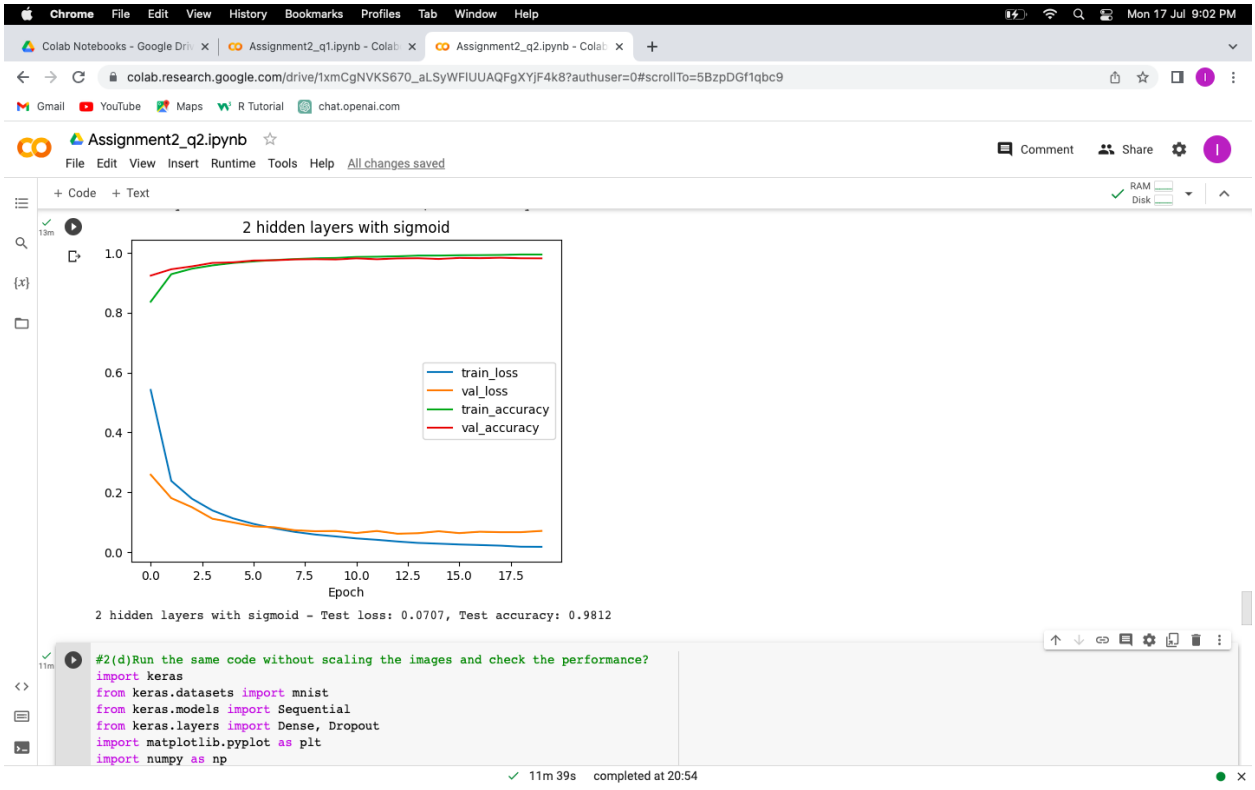



```
1 hidden layer with sigmoid - Test loss: 0.0612, Test accuracy: 0.9819
```



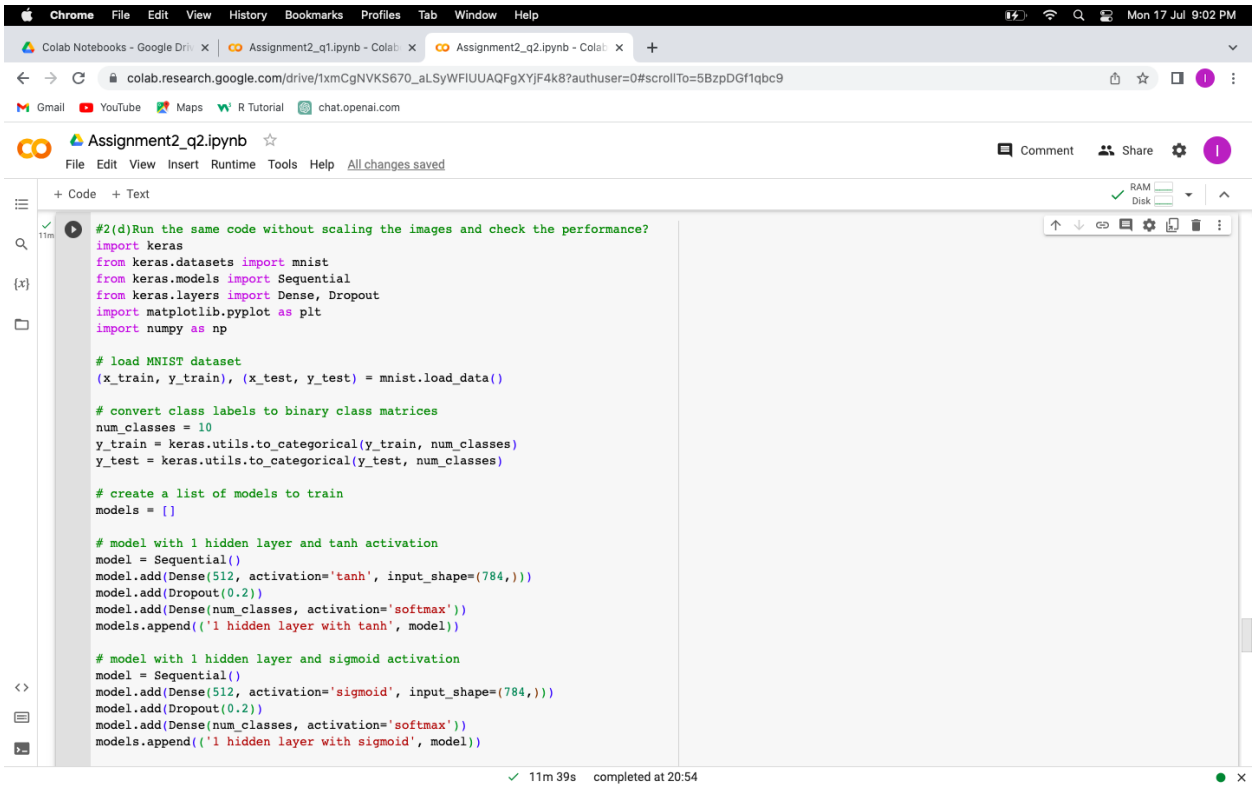
11m 39s completed at 20:54


2 hidden layers with tanh
train_loss
val_loss
train_accuracy
val_accuracy
Epoch
2 hidden layers with tanh - Test loss: 0.0726, Test accuracy: 0.9802
2 hidden layers with sigmoid


2 hidden layers with sigmoid
train_loss
val_loss
train_accuracy
val_accuracy
Epoch
2 hidden layers with sigmoid - Test loss: 0.0707, Test accuracy: 0.9812
#2(d)Run the same code without scaling the images and check the performance?
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
import matplotlib.pyplot as plt
import numpy as np

d.) Run the same code without scaling the images and check the performance

Ans)
Code

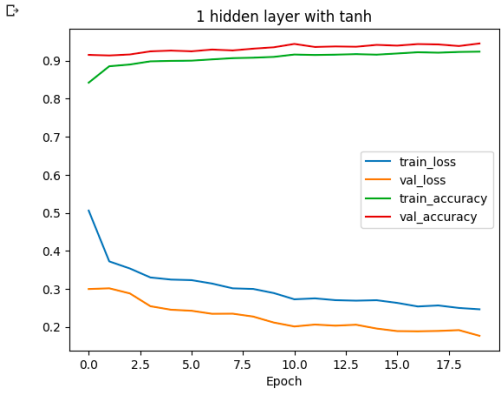


```
# model with 2 hidden layers and tanh activation
model = Sequential()
model.add(Dense(512, activation='tanh', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='tanh'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('2 hidden layers with tanh', model))
# model with 2 hidden layers and sigmoid activation
model = Sequential()
model.add(Dense(512, activation='sigmoid', input_shape=(784,)))
model.add(Dropout(0.2))
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
```

11m 39s completed at 20:54

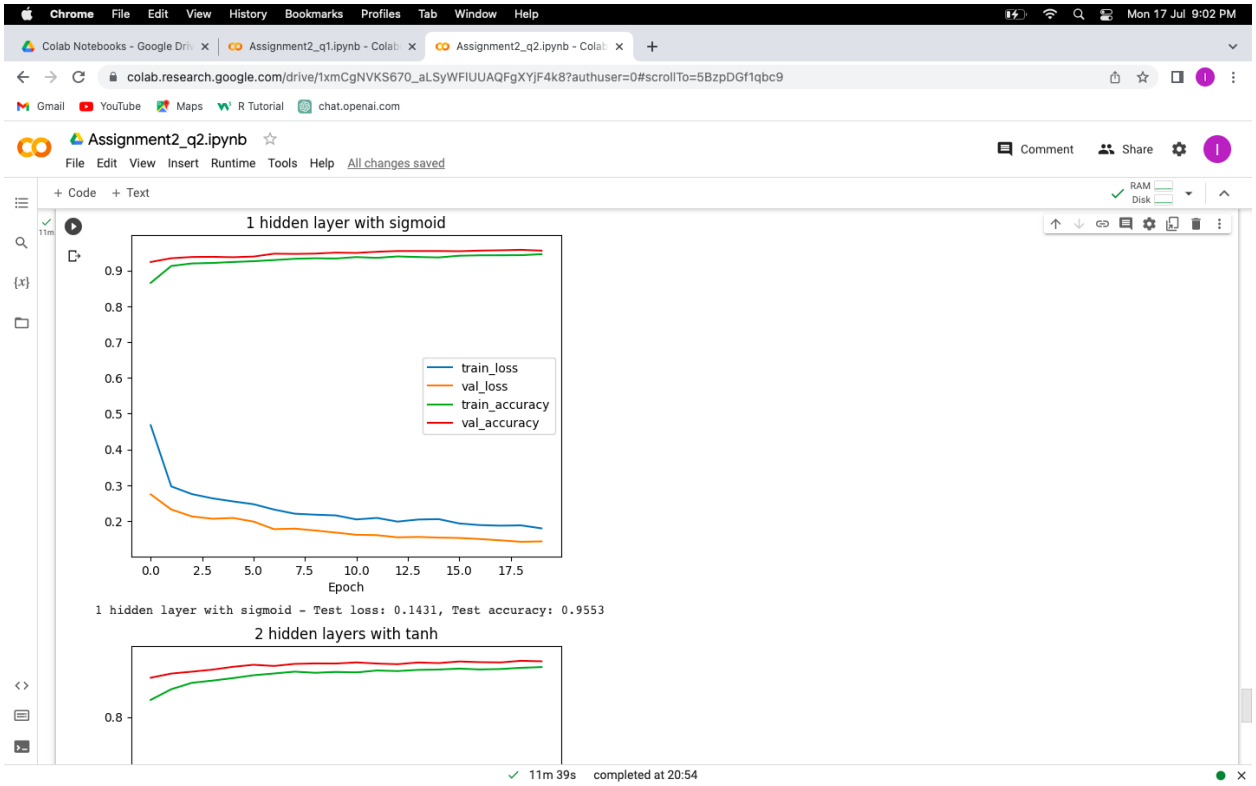
Output:

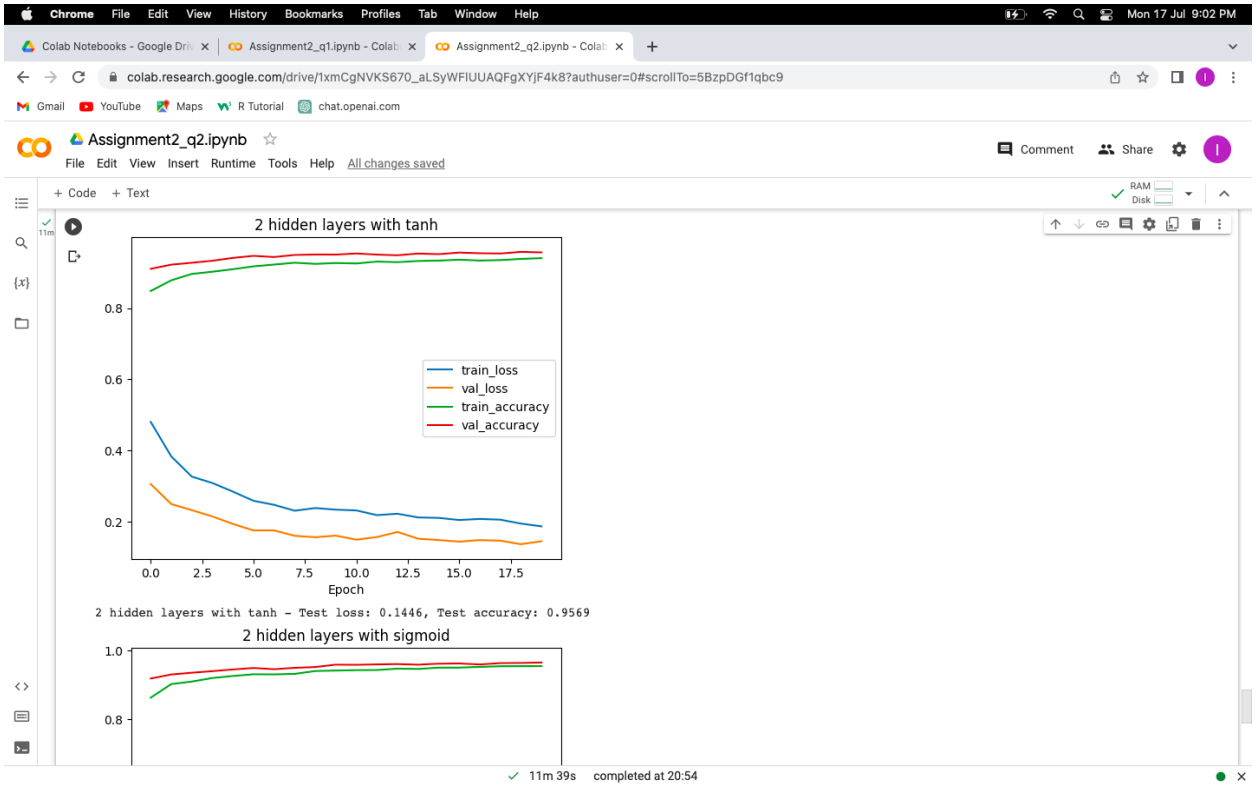
```
    plt.xlabel('Epoch')
    plt.legend()
    plt.show()

    # 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))
```

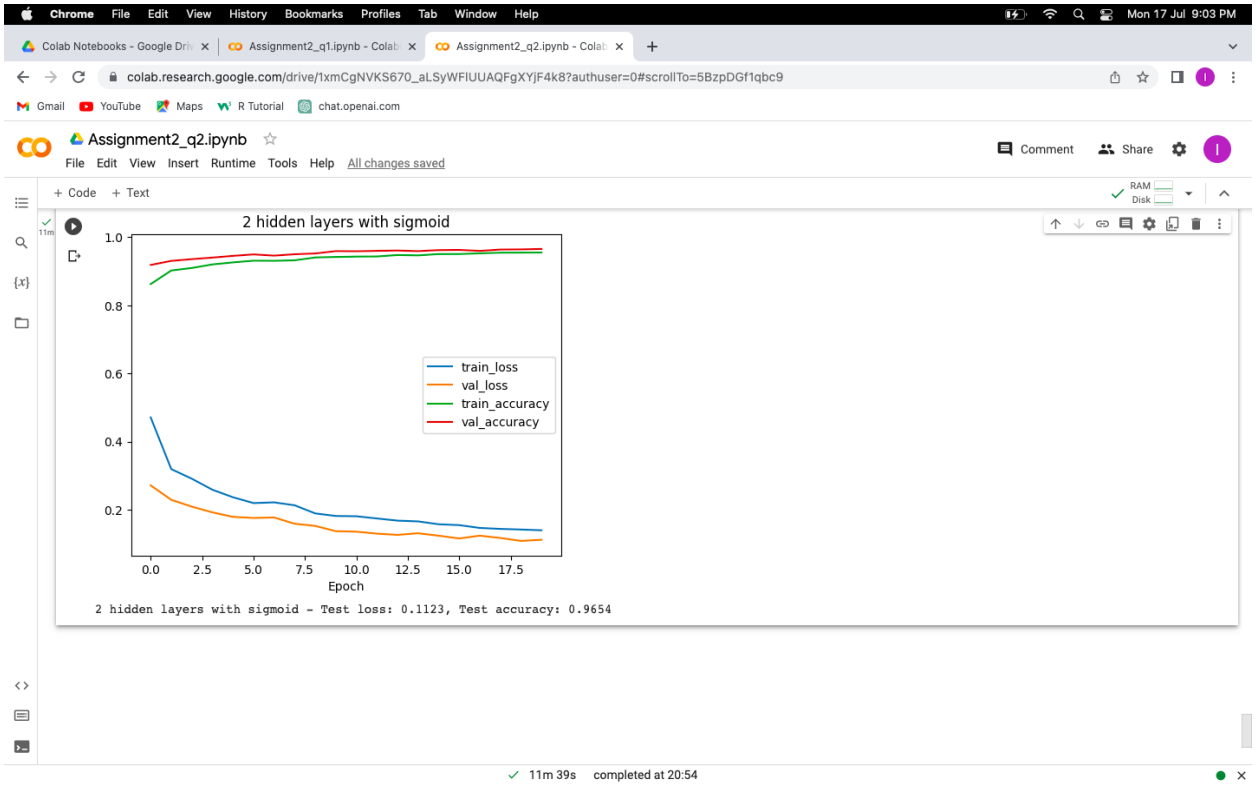



```
1 hidden layer with tanh - Test loss: 0.1766, Test accuracy: 0.9453
```

11m 39s completed at 20:54


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 9:02 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
1 hidden layer with sigmoid
train_loss
val_loss
train_accuracy
val_accuracy
Epoch
1 hidden layer with sigmoid - Test loss: 0.1431, Test accuracy: 0.9553
2 hidden layers with tanh
11m 39s completed at 20:54


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 17 Jul 9:02 PM
Colab Notebooks - Google Dri
Assignment2_q1.ipynb - Colab
Assignment2_q2.ipynb - Colab
colab.research.google.com/drive/1xmCgNVKS670_aLSyWFlUUAQFgXYjF4k8?authuser=0#scrollTo=5BzpDGf1qbc9
Gmail YouTube Maps R Tutorial chat.openai.com
Assignment2_q2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
2 hidden layers with tanh
train_loss
val_loss
train_accuracy
val_accuracy
Epoch
2 hidden layers with tanh - Test loss: 0.1446, Test accuracy: 0.9569
2 hidden layers with sigmoid
11m 39s completed at 20:54



Github Link:

https://github.com/LahariKollipara/NNDL_Assignment2