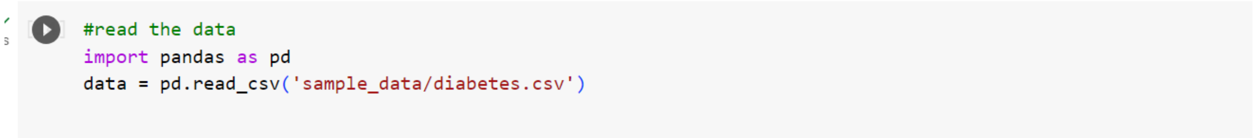
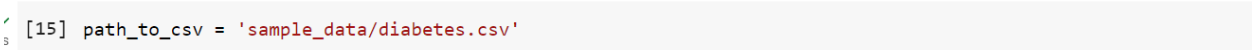
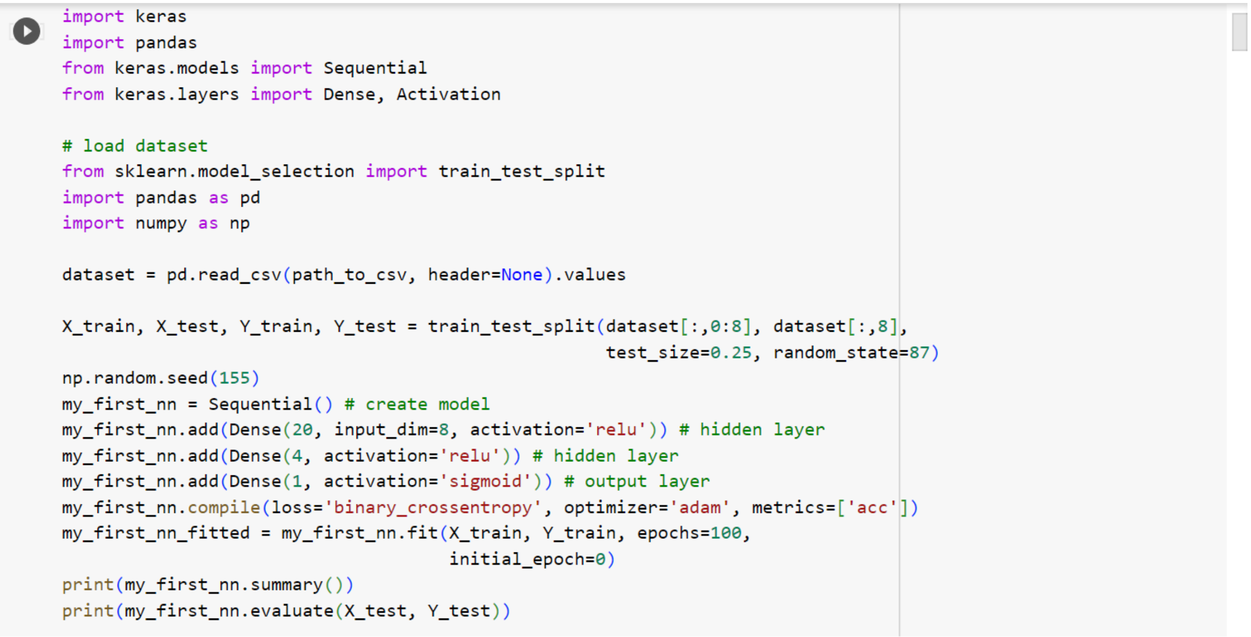
Fall 2023: CS5720 Neural Networks & Deep Learning - ICP-6
Assignment-6
NAME:RAJYALAKSHMI GOTTIPATI
STUDENT ID:700745186

Github Link:https://github.com/rajigottipati/icp-6.git
Video Link:
https://drive.google.com/file/d/1QkAT3t6cZUft_QTxecPAdrK-mJwU-Ox7/view?usp=drive_link

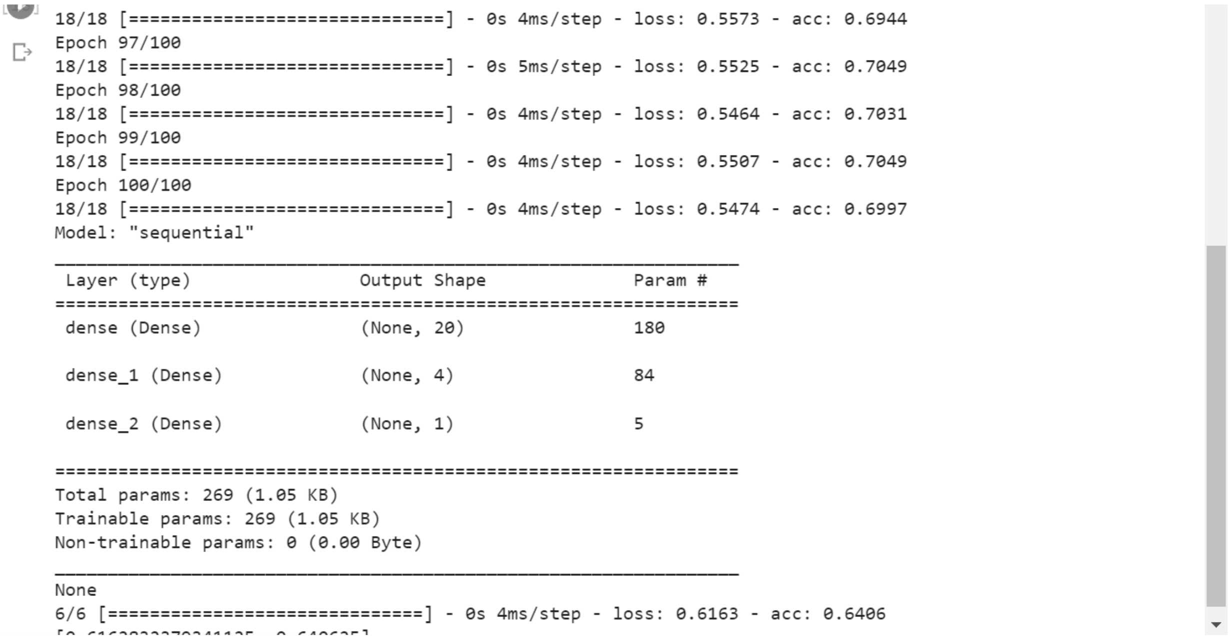
programming: 1. Use the use case in the class: a. Add more Dense layers to the existing code and check how the accuracy changes. 2. Change the data source to Breast Cancer dataset * available in the source code folder and make required changes. Report accuracy of the model. 3. Normalize the data before feeding the data to the model and check how the normalization change your accuracy (code given below). from sklearn.preprocessing import StandardScaler sc = StandardScaler() Breast Cancer dataset is designated to predict if a patient has Malignant (M) or Benign = B cancer In class programming: 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. 4. Run the same code without scaling the images and check the performance?


```
#read the data
import pandas as pd
data = pd.read_csv('sample_data/diabetes.csv')
```

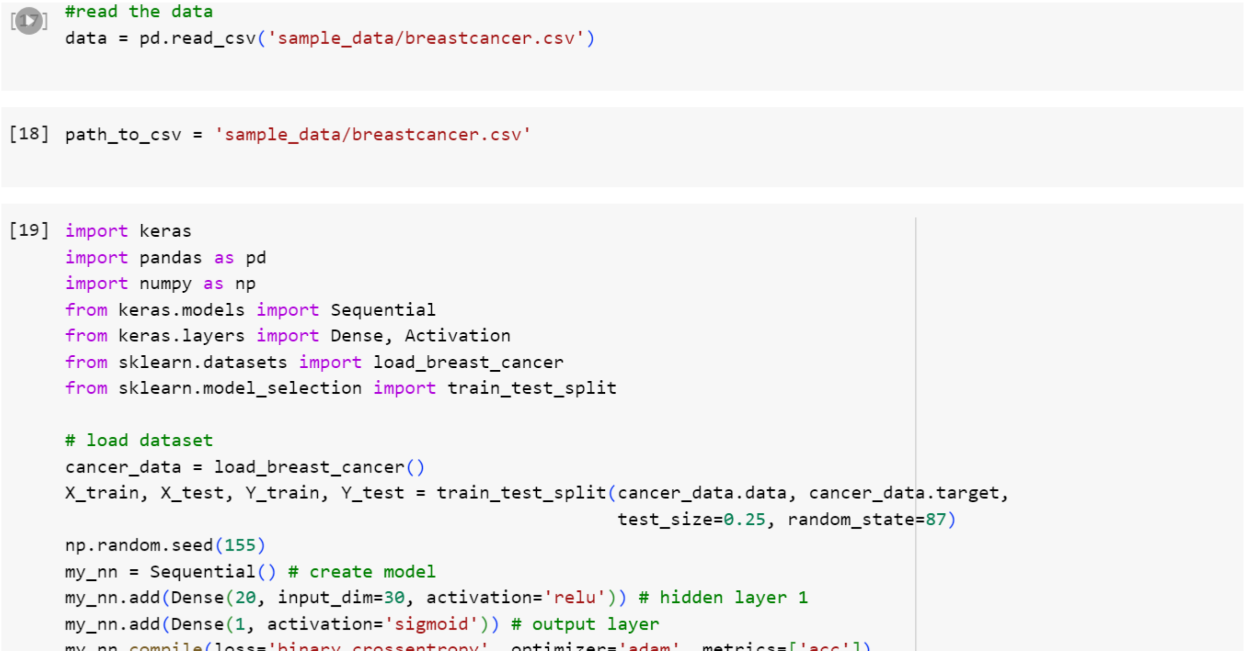

```
[15] path_to_csv = 'sample_data/diabetes.csv'
```



```
import keras
import pandas
from keras.models import Sequential
from keras.layers 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(4, activation='relu')) # hidden layer
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))
```



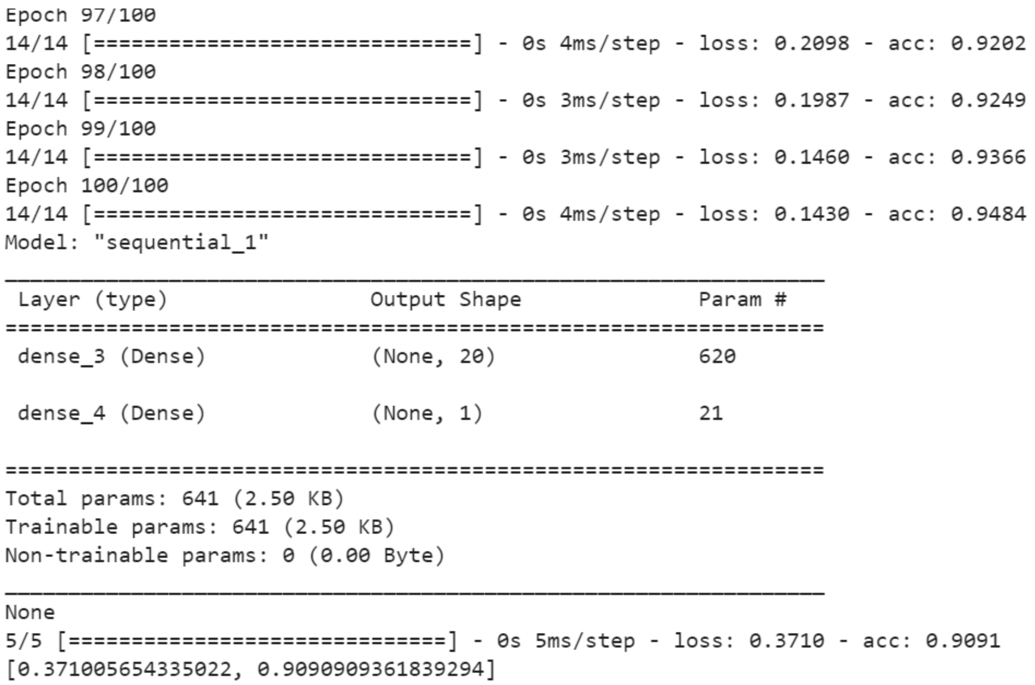
```
18/18 [==============================] - 0s 4ms/step - loss: 0.5573 - acc: 0.6944
Epoch 97/100
18/18 [==============================] - 0s 5ms/step - loss: 0.5525 - acc: 0.7049
Epoch 98/100
18/18 [==============================] - 0s 4ms/step - loss: 0.5464 - acc: 0.7031
Epoch 99/100
18/18 [==============================] - 0s 4ms/step - loss: 0.5507 - acc: 0.7049
Epoch 100/100
18/18 [==============================] - 0s 4ms/step - loss: 0.5474 - acc: 0.6997
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 20)                180

 dense_1 (Dense)             (None, 4)                 84

 dense_2 (Dense)             (None, 1)                 5

=================================================================
Total params: 269 (1.05 KB)
Trainable params: 269 (1.05 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
6/6 [==============================] - 0s 4ms/step - loss: 0.6163 - acc: 0.6406
[0.6162832379341125, 0.640625]
```



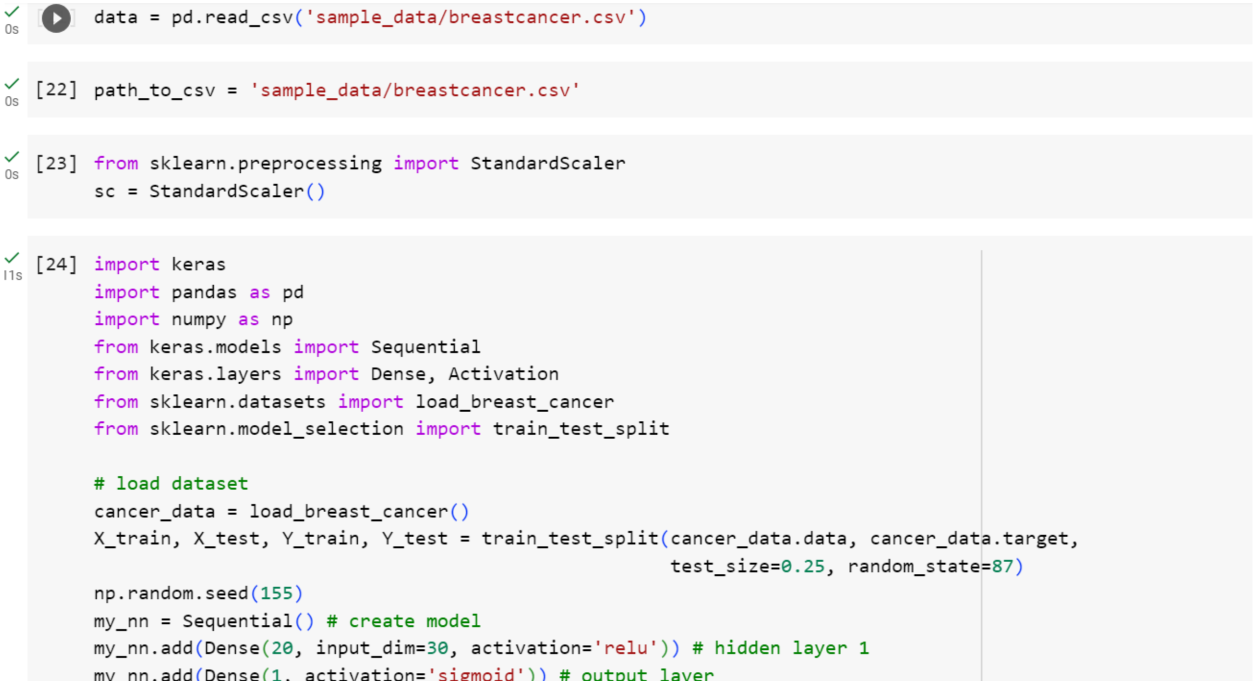
```
#read the data
data = pd.read_csv('sample_data/breastcancer.csv')
```

```
[18] path_to_csv = 'sample_data/breastcancer.csv'
```

```
[19] import keras
     import pandas as pd
     import numpy as np
     from keras.models import Sequential
     from keras.layers import Dense, Activation
     from sklearn.datasets import load_breast_cancer
     from sklearn.model_selection import train_test_split

     # load dataset
     cancer_data = load_breast_cancer()
     X_train, X_test, Y_train, Y_test = train_test_split(cancer_data.data, cancer_data.target,
                                                         test_size=0.25, random_state=87)
     np.random.seed(155)
     my_nn = Sequential() # create model
     my_nn.add(Dense(20, input_dim=30, activation='relu')) # hidden layer 1
     my_nn.add(Dense(1, activation='sigmoid')) # output layer
     my_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
```



```
Epoch 97/100
14/14 [==============================] - 0s 4ms/step - loss: 0.2098 - acc: 0.9202
Epoch 98/100
14/14 [==============================] - 0s 3ms/step - loss: 0.1987 - acc: 0.9249
Epoch 99/100
14/14 [==============================] - 0s 3ms/step - loss: 0.1460 - acc: 0.9366
Epoch 100/100
14/14 [==============================] - 0s 4ms/step - loss: 0.1430 - acc: 0.9484
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 5ms/step - loss: 0.3710 - acc: 0.9091
[0.371005654335022, 0.9090909361839294]
```

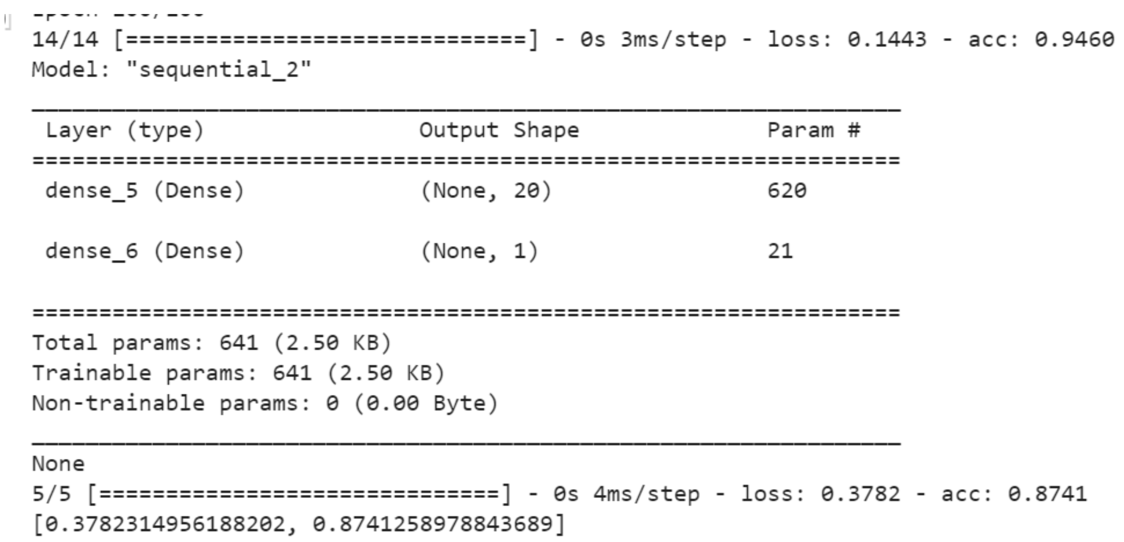


```
data = pd.read_csv('sample_data/breastcancer.csv')
```

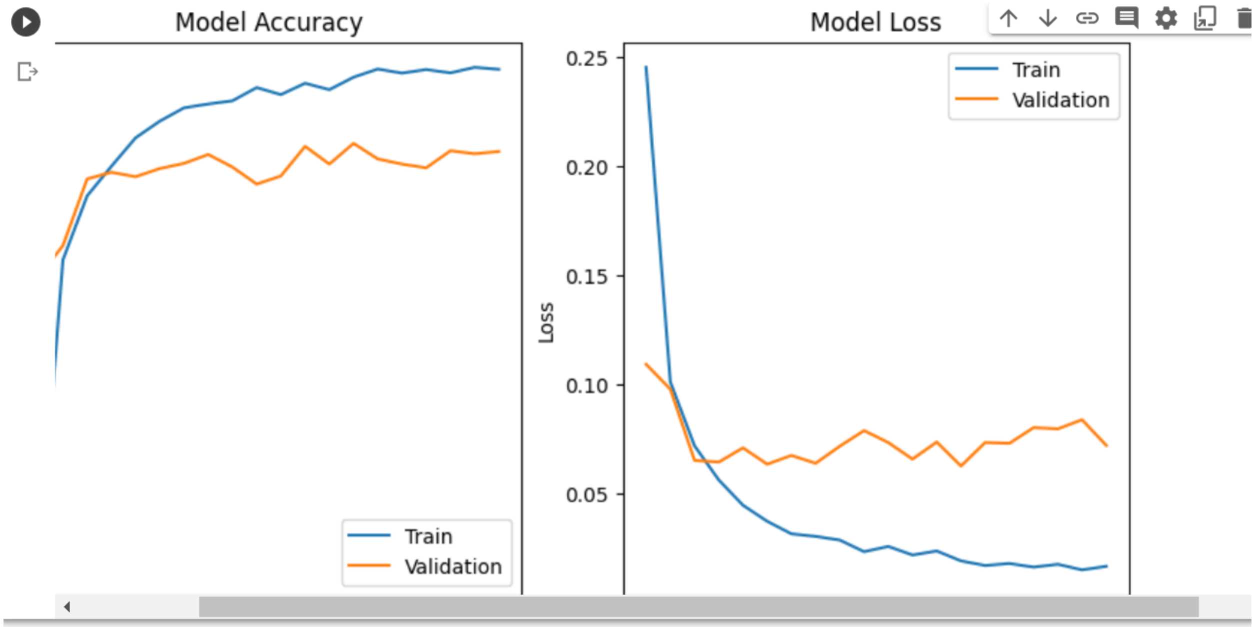
```
[22] path_to_csv = 'sample_data/breastcancer.csv'
```

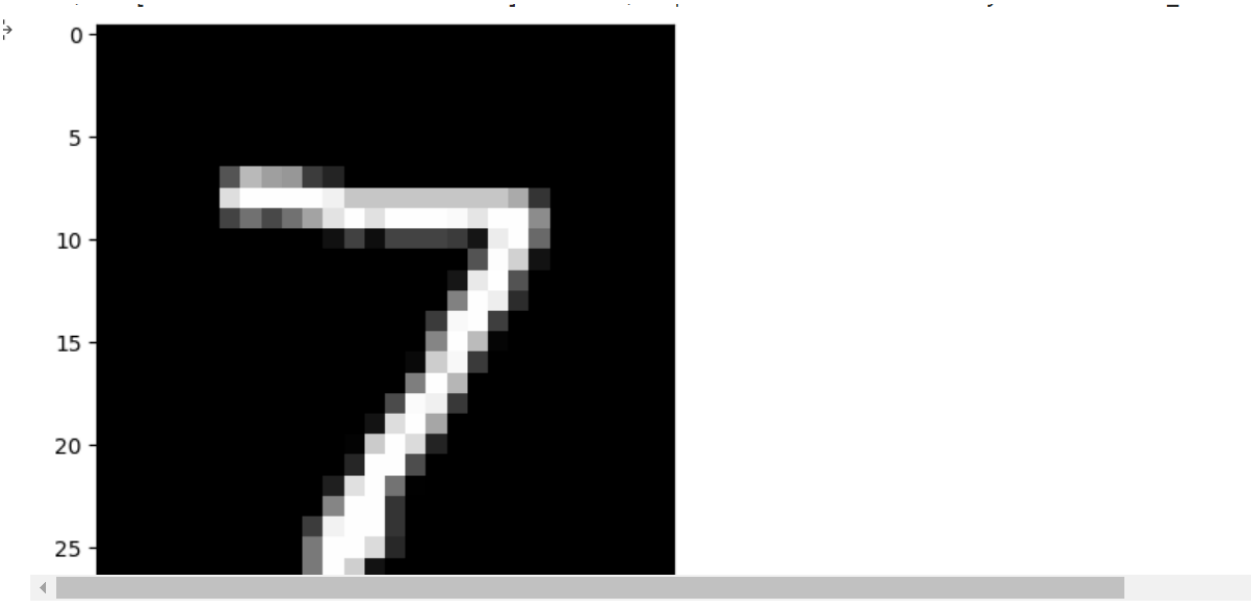
```
[23] from sklearn.preprocessing import StandardScaler
     sc = StandardScaler()
```

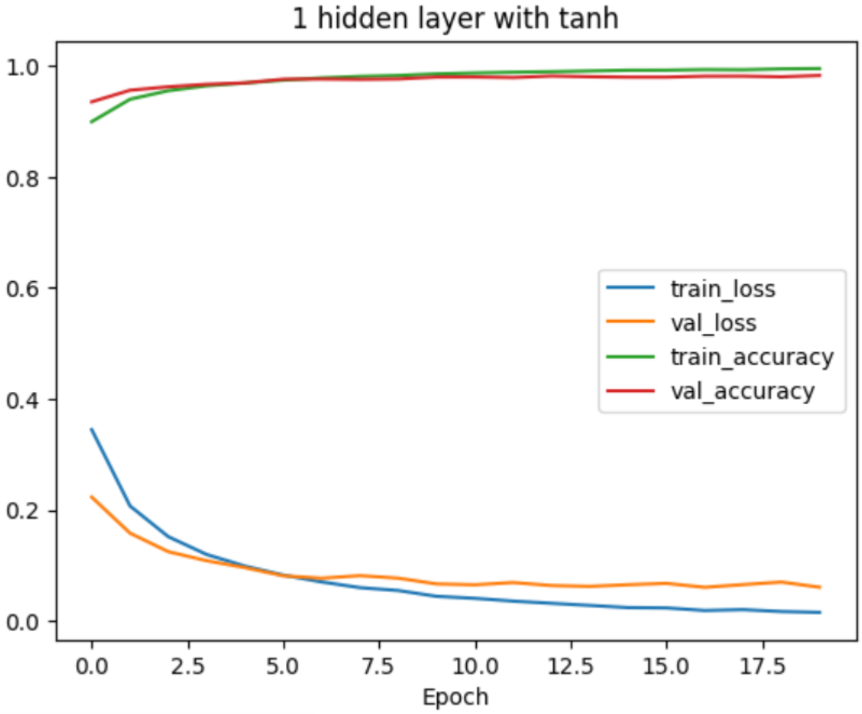
```
[24] import keras
     import pandas as pd
     import numpy as np
     from keras.models import Sequential
     from keras.layers import Dense, Activation
     from sklearn.datasets import load_breast_cancer
     from sklearn.model_selection import train_test_split

     # load dataset
     cancer_data = load_breast_cancer()
     X_train, X_test, Y_train, Y_test = train_test_split(cancer_data.data, cancer_data.target,
                                                         test_size=0.25, random_state=87)
     np.random.seed(155)
     my_nn = Sequential() # create model
     my_nn.add(Dense(20, input_dim=30, activation='relu')) # hidden layer 1
     my_nn.add(Dense(1, activation='sigmoid')) # output layer
```

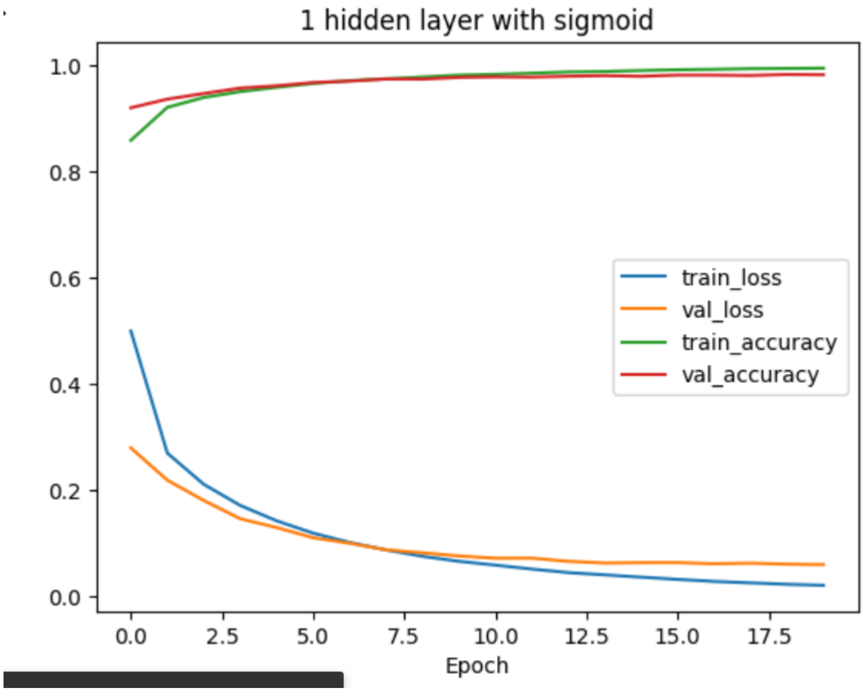


```
14/14 [==============================] - 0s 3ms/step - loss: 0.1443 - acc: 0.9460
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_5 (Dense)             (None, 20)                620

 dense_6 (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: 0.3782 - acc: 0.8741
[0.3782314956188202, 0.8741258978843689]
```

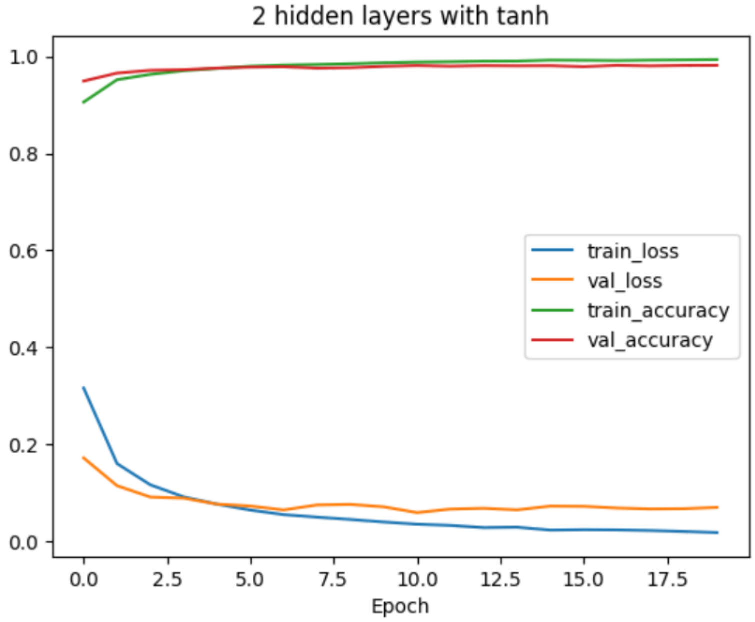

Model Accuracy
Train
Validation
Model Loss
0.25
0.20
0.15
0.10
0.05
Loss
Train
Validation


0
5
10
15
20
25

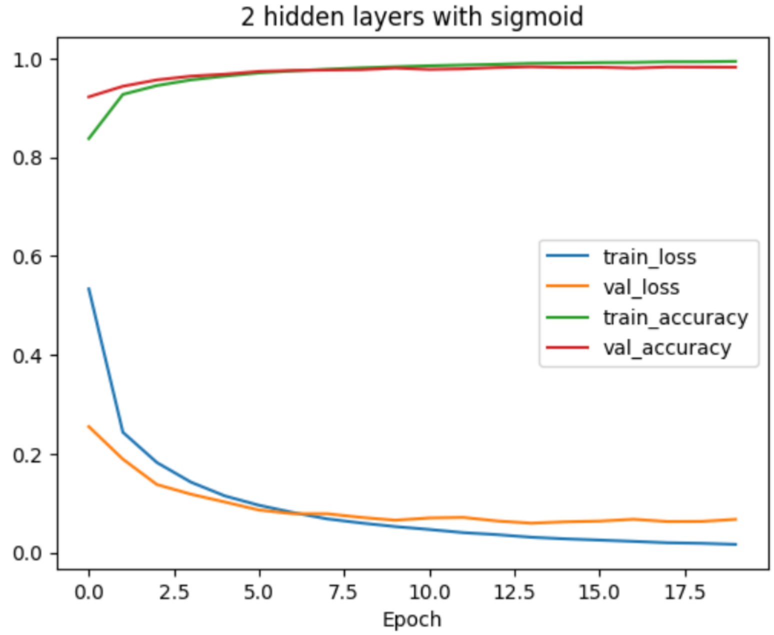

1 hidden layer with tanh
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
train_loss
val_loss
train_accuracy
val_accuracy



[redacted]



[redacted]



```
[redacted]id - Test loss: 0.0673, Test accuracy: 0.9824
```