Name: Tarun Tanmay

Class: MBATech CE

Sem: 6

Roll No: N049

#Experiment 3

#ANN with Multiple Layers for classification

Importing required libraries

from numpy import loadtxt #loading data from a text file or a csv file

from keras.models import Sequential #(used to build the ann model with sequential layers)

from keras.layers import Dense #(used for entering parameters for each layer)

Importing the Dataset

dataset=loadtxt('pima-indians-diabetes.csv', delimiter=',') #importing the dataset text, separated with ,

dataset

#dataset.shape #(768rows and 9 columns)

#dataset[0,:]

array([[ 6. , 148. , 72. , ..., 0.627, 50. , 1. ],

[ 1. , 85. , 66. , ..., 0.351, 31. , 0. ],

[ 8. , 183. , 64. , ..., 0.672, 32. , 1. ],

...,

[ 5. , 121. , 72. , ..., 0.245, 30. , 0. ],

[ 1. , 126. , 60. , ..., 0.349, 47. , 1. ],

[ 1. , 93. , 70. , ..., 0.315, 23. , 0. ]])

Explanation of Dataset:

column1: number of times person has undrgone surgery

column2: glucose concentration

column3: blood pressure

column4: skinfold thickness

column5: serum insulin

column6: Body Mass Index (BMI)

coliumn7: medical parameter

column8: age

column9: person has diabetes (1) or not (0)

X=dataset[:,0:8] #all rows and columns from 1 to 8

y=dataset[:,8] #all rows and only 8th column  
  
Training and Testing the Model

from sklearn.model\_selection import train\_test\_split #randomly splitting the dataset for training and testing the model

train\_x,test\_x,train\_y,test\_y=train\_test\_split(X,y,test\_size=0.2, random\_state=4) #test size is given as 20%, random\_state is kept same for same answer

test\_x.shape

(154, 8)

model=Sequential()

model.add(Dense(12,input\_dim=8, activation='relu')) #relu function is used as the activation function

#8 attributes, one per each sample (column)

#12 neurons are there in the next hidden layer, after input layer having 8 neurons

model.add(Dense(8,activation='relu'))

model.add(Dense(1,activation='sigmoid')) #classifier: logic 1 or logic 0

#In last layer only 1 neuron is present and thus binary classifier is used 

Compiling the Model

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

#{(probability of actual class) \* (log of probability of predicted class)}

#loss=act\*log(pred)

model.fit(train\_x,train\_y,epochs=150,batch\_size=20)

#after claclulating loss, we backpropogate and see the gradient

#after weights are changed, input is given again and output is claculated again

Epoch 1/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4648 - accuracy: 0.7638 Epoch 2/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4557 - accuracy: 0.7638 Epoch 3/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4633 - accuracy: 0.7720 Epoch 4/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4596 - accuracy: 0.7704 Epoch 5/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4554 - accuracy: 0.7687 Epoch 6/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4603 - accuracy: 0.7704 Epoch 7/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4672 - accuracy: 0.7557 Epoch 8/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4569 - accuracy: 0.7671 Epoch 9/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4707 - accuracy: 0.7590 Epoch 10/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4759 - accuracy: 0.7590 Epoch 11/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4570 - accuracy: 0.7785 Epoch 12/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4565 - accuracy: 0.7785 Epoch 13/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4588 - accuracy: 0.7655 Epoch 14/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4553 - accuracy: 0.7785 Epoch 15/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4563 - accuracy: 0.7752 Epoch 16/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4589 - accuracy: 0.7736 Epoch 17/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4593 - accuracy: 0.7834 Epoch 18/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4711 - accuracy: 0.7622 Epoch 19/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4566 - accuracy: 0.7818 Epoch 20/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4792 - accuracy: 0.7524 Epoch 21/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4574 - accuracy: 0.7736 Epoch 22/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4610 - accuracy: 0.7638 Epoch 23/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4562 - accuracy: 0.7687 Epoch 24/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4581 - accuracy: 0.7590 Epoch 25/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4639 - accuracy: 0.7818 Epoch 26/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4616 - accuracy: 0.7671 Epoch 27/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4622 - accuracy: 0.7671 Epoch 28/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4524 - accuracy: 0.7736 Epoch 29/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4599 - accuracy: 0.7704 Epoch 30/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4641 - accuracy: 0.7704 Epoch 31/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4510 - accuracy: 0.7801 Epoch 32/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4616 - accuracy: 0.7801 Epoch 33/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4652 - accuracy: 0.7720 Epoch 34/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4600 - accuracy: 0.7622 Epoch 35/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4563 - accuracy: 0.7736 Epoch 36/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4628 - accuracy: 0.7671 Epoch 37/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4535 - accuracy: 0.7769 Epoch 38/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4752 - accuracy: 0.7687 Epoch 39/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4628 - accuracy: 0.7769 Epoch 40/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4600 - accuracy: 0.7752 Epoch 41/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4613 - accuracy: 0.7769 Epoch 42/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4545 - accuracy: 0.7736 Epoch 43/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4569 - accuracy: 0.7638 Epoch 44/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4592 - accuracy: 0.7785 Epoch 45/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4655 - accuracy: 0.7769 Epoch 46/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4893 - accuracy: 0.7638 Epoch 47/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4582 - accuracy: 0.7899 Epoch 48/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4922 - accuracy: 0.7345 Epoch 49/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4557 - accuracy: 0.7671 Epoch 50/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4546 - accuracy: 0.7883 Epoch 51/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4743 - accuracy: 0.7590 Epoch 52/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4566 - accuracy: 0.7720 Epoch 53/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4566 - accuracy: 0.7850 Epoch 54/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4606 - accuracy: 0.7736 Epoch 55/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4610 - accuracy: 0.7704 Epoch 56/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4842 - accuracy: 0.7655 Epoch 57/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4551 - accuracy: 0.7850 Epoch 58/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4633 - accuracy: 0.7687 Epoch 59/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4593 - accuracy: 0.7818 Epoch 60/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4503 - accuracy: 0.7704 Epoch 61/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4455 - accuracy: 0.7850 Epoch 62/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4595 - accuracy: 0.7736 Epoch 63/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4551 - accuracy: 0.7801 Epoch 64/150 31/31 [==============================] - 0s 2ms/step - loss: 0.4537 - accuracy: 0.7736 Epoch 65/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4569 - accuracy: 0.7785 Epoch 66/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4582 - accuracy: 0.7801 Epoch 67/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4541 - accuracy: 0.7752 Epoch 68/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4542 - accuracy: 0.7720 Epoch 69/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4496 - accuracy: 0.7801 Epoch 70/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4504 - accuracy: 0.7883 Epoch 71/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4511 - accuracy: 0.7769 Epoch 72/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4536 - accuracy: 0.7769 Epoch 73/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4660 - accuracy: 0.7622 Epoch 74/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4526 - accuracy: 0.7769 Epoch 75/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4768 - accuracy: 0.7671 Epoch 76/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4606 - accuracy: 0.7622 Epoch 77/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4494 - accuracy: 0.7834 Epoch 78/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4502 - accuracy: 0.7752 Epoch 79/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4551 - accuracy: 0.7736 Epoch 80/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4512 - accuracy: 0.7883 Epoch 81/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4497 - accuracy: 0.7785 Epoch 82/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4459 - accuracy: 0.7866 Epoch 83/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4532 - accuracy: 0.7883 Epoch 84/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4532 - accuracy: 0.7769 Epoch 85/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4572 - accuracy: 0.7948 Epoch 86/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4673 - accuracy: 0.7785 Epoch 87/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4515 - accuracy: 0.7785 Epoch 88/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4649 - accuracy: 0.7687 Epoch 89/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4729 - accuracy: 0.7655 Epoch 90/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4575 - accuracy: 0.7638 Epoch 91/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4592 - accuracy: 0.7736 Epoch 92/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4745 - accuracy: 0.7638 Epoch 93/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4631 - accuracy: 0.7720 Epoch 94/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4503 - accuracy: 0.7720 Epoch 95/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4459 - accuracy: 0.7818 Epoch 96/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4490 - accuracy: 0.7736 Epoch 97/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4520 - accuracy: 0.7883 Epoch 98/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4527 - accuracy: 0.7655 Epoch 99/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4459 - accuracy: 0.7899 Epoch 100/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4511 - accuracy: 0.7866 Epoch 101/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4533 - accuracy: 0.7801 Epoch 102/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4649 - accuracy: 0.7785 Epoch 103/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4529 - accuracy: 0.7915 Epoch 104/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4439 - accuracy: 0.7769 Epoch 105/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4520 - accuracy: 0.7834 Epoch 106/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4563 - accuracy: 0.7932 Epoch 107/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4491 - accuracy: 0.7818 Epoch 108/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4442 - accuracy: 0.7866 Epoch 109/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4505 - accuracy: 0.7785 Epoch 110/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4589 - accuracy: 0.7671 Epoch 111/150 31/31 [==============================] - 0s 2ms/step - loss: 0.4518 - accuracy: 0.7769 Epoch 112/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4652 - accuracy: 0.7801 Epoch 113/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4507 - accuracy: 0.7769 Epoch 114/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4539 - accuracy: 0.7769 Epoch 115/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4437 - accuracy: 0.7850 Epoch 116/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4480 - accuracy: 0.7638 Epoch 117/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4440 - accuracy: 0.7866 Epoch 118/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4538 - accuracy: 0.7720 Epoch 119/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4643 - accuracy: 0.7850 Epoch 120/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4543 - accuracy: 0.7638 Epoch 121/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4529 - accuracy: 0.7801 Epoch 122/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4497 - accuracy: 0.7752 Epoch 123/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4504 - accuracy: 0.7769 Epoch 124/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4574 - accuracy: 0.7818 Epoch 125/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4468 - accuracy: 0.7834 Epoch 126/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4429 - accuracy: 0.7866 Epoch 127/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4461 - accuracy: 0.7752 Epoch 128/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4479 - accuracy: 0.7769 Epoch 129/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4499 - accuracy: 0.7704 Epoch 130/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4535 - accuracy: 0.7655 Epoch 131/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4405 - accuracy: 0.7964 Epoch 132/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4634 - accuracy: 0.7606 Epoch 133/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4738 - accuracy: 0.7590 Epoch 134/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4584 - accuracy: 0.7801 Epoch 135/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4489 - accuracy: 0.7671 Epoch 136/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4525 - accuracy: 0.7687 Epoch 137/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4459 - accuracy: 0.7850 Epoch 138/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4430 - accuracy: 0.7752 Epoch 139/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4464 - accuracy: 0.7818 Epoch 140/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4596 - accuracy: 0.7704 Epoch 141/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4493 - accuracy: 0.7736 Epoch 142/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4684 - accuracy: 0.7638 Epoch 143/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4545 - accuracy: 0.7736 Epoch 144/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4484 - accuracy: 0.7850 Epoch 145/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4427 - accuracy: 0.7818 Epoch 146/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4472 - accuracy: 0.7866 Epoch 147/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4474 - accuracy: 0.7866 Epoch 148/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4477 - accuracy: 0.7801 Epoch 149/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4421 - accuracy: 0.7801 Epoch 150/150 31/31 [==============================] - 0s 1ms/step - loss: 0.4622 - accuracy: 0.7818

<tensorflow.python.keras.callbacks.History at 0x7fe7d70c71d0>

training\_accuracy=model.evaluate(train\_x,train\_y)

print("%s:%.2f%%" % (model.metrics\_names[0], training\_accuracy[0]\*100))

20/20 [==============================] - 0s 952us/step - loss: 0.4364 - accuracy: 0.7866 loss:43.64%

training\_accuracy

[0.4363956153392792, 0.7866449356079102]

test\_accuracy=model.evaluate(test\_x,test\_y)

print("%s:%.2f%%" % (model.metrics\_names[1], test\_accuracy[1]\*100))

5/5 [==============================] - 0s 2ms/step - loss: 0.5403 - accuracy: 0.7727 accuracy:77.27%

We got almost the same accuracy for both training and testing set, so model is correct.

If network was not trained well, then high bias

If network was not tested well, then high variance

predict=model.predict\_classes(test\_x)

for i in range(10):

  print('%s ==> %d (expected %d)' % (test\_x[i].tolist(), predict[i],test\_y[i]))

[1.0, 100.0, 66.0, 29.0, 196.0, 32.0, 0.444, 42.0] ==> 0 (expected 0) [9.0, 57.0, 80.0, 37.0, 0.0, 32.8, 0.096, 41.0] ==> 0 (expected 0) [0.0, 100.0, 70.0, 26.0, 50.0, 30.8, 0.597, 21.0] ==> 0 (expected 0) [1.0, 119.0, 88.0, 41.0, 170.0, 45.3, 0.507, 26.0] ==> 0 (expected 0) [2.0, 102.0, 86.0, 36.0, 120.0, 45.5, 0.127, 23.0] ==> 0 (expected 1) [13.0, 126.0, 90.0, 0.0, 0.0, 43.4, 0.583, 42.0] ==> 1 (expected 1) [3.0, 171.0, 72.0, 33.0, 135.0, 33.3, 0.199, 24.0] ==> 0 (expected 1) [12.0, 92.0, 62.0, 7.0, 258.0, 27.6, 0.926, 44.0] ==> 1 (expected 1) [12.0, 151.0, 70.0, 40.0, 271.0, 41.8, 0.742, 38.0] ==> 1 (expected 1) [11.0, 85.0, 74.0, 0.0, 0.0, 30.1, 0.3, 35.0] ==> 0 (expected 0)

/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `model.predict\_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:\* `np.argmax(model.predict(x), axis=-1)`, if your model does multi-class classification (e.g. if it uses a `softmax` last-layer activation).\* `(model.predict(x) > 0.5).astype("int32")`, if your model does binary classification (e.g. if it uses a `sigmoid` last-layer activation). warnings.warn('`model.predict\_classes()` is deprecated and '

First ten samples are taken, and predicted and actual data is compared