**Python for Deep Learning**

**Module 1: Deep Learning programming – Lesson 1**

1. Adding more dense layers, so I have changed from 20 to 25 Neurons in my Dense Layer in the Hidden Layer.

my\_first\_nn = Sequential() # create model

my\_first\_nn.add(Dense(25, input\_dim=8, 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)

A screenshot of a social media post

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Loading the data set.

dataset = pd.read\_csv("/content/drive/My Drive/Colab Notebooks/DL1/breastcancer.csv").values

from sklearn import preprocessing

pre\_process = preprocessing.LabelEncoder()

pre\_process.fit(dataset[:,1])

dataset[:,1] = pre\_process.transform(dataset[:,1])

mapping = {'M': 1, 'B': 2}

A screenshot of a social media post

Description automatically generated

1. Using Standard Scalar Normalization

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

print (sc.fit(dataset))

1. Using different optimizers

* Using Adam optimizer

my\_first\_nn = Sequential() # create model

my\_first\_nn.add(Dense(50, input\_dim=29, 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)

A screenshot of a computer

Description automatically generated

* Using rmsprop optimizer

my\_first\_nn = Sequential() # create model

my\_first\_nn.add(Dense(50, input\_dim=29, activation='relu')) # hidden layer

my\_first\_nn.add(Dense(1, activation='sigmoid')) # output layer

my\_first\_nn.compile(loss='binary\_crossentropy', optimizer='rmsprop', metrics=['acc'])

my\_first\_nn\_fitted = my\_first\_nn.fit(X\_train, Y\_train, epochs=100,initial\_epoch=0)

A screenshot of a computer

Description automatically generated