**Neural Network Deep Learning**

Assignment – 6

Name: Kishor Kumar Andekar

Student ID: 700744713

Github Link : <https://github.com/kishorreyansh/Neural-Network-Deep-Learning/tree/main/Assignment-6>

Use Case Description: Predicting the diabetes disease

Programming elements: Keras Basics

1.Use the use case in the class:

A screenshot of a computer program

Description automatically generated

A computer code with text

Description automatically generated with medium confidence

**Output:**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer code

Description automatically generated

A table of numbers and lines

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer code

Description automatically generated

A computer code with many colorful text

Description automatically generated with medium confidence

**OUTPUT:**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA table of text with numbers

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated**

**A table of numbers and lines

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. 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

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Output:**

**A screenshot of a computer code

Description automatically generated**

**A table of numbers and lines

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

**A table of numbers and lines

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A white background with black text

Description automatically generated**

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()

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**OUTPUT:-**

**A table of numbers and lines

Description automatically generated**

**A table of numbers and lines

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A table of numbers and lines

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

In class programming:

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

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**OUTPUT:-**

**A screenshot of a computer

Description automatically generated**

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

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

**OUTPUT:-**

**A screenshot of a computer code

Description automatically generated**

**A graph of loss and model loss

Description automatically generated**

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.

**A screenshot of a computer

Description automatically generated**

**A computer screen shot of a program

Description automatically generated**

**A computer code with black and green text

Description automatically generated**

**OUTPUT:-**

**A screenshot of a computer

Description automatically generated**

A black and white image of a number

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

A computer screen shot of a code

Description automatically generated

**OUTPUT:-**

**A screenshot of a computer

Description automatically generated**

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

**A screenshot of a computer program

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**OUTPUT:-**

A screenshot of a computer screen

Description automatically generated