ExpNo: 06

Date : 12/10/24

IMPLEMENTING ARTIFICIAL NEURAL NETWORKS FOR AN APPLICATION USING PYTHON - CLASSIFICATION

AIM:

To implement artificial neural networks for an application using python classification.

ABOUT:

* Contains artificial neurons
* The neurons are connected to each other.
* They are arranged in layers to constitute a neural networks.
* The data process through these multiple layers are get processed.
* The output layer provides output for the networks.

ALGORITHM:

(i) Start by importing necessary libraries.

(ii) Load the Iris dataset.

(iii) Split data set into training and testing

(iv) Create simple forward neural network

(v) Fit model to training data

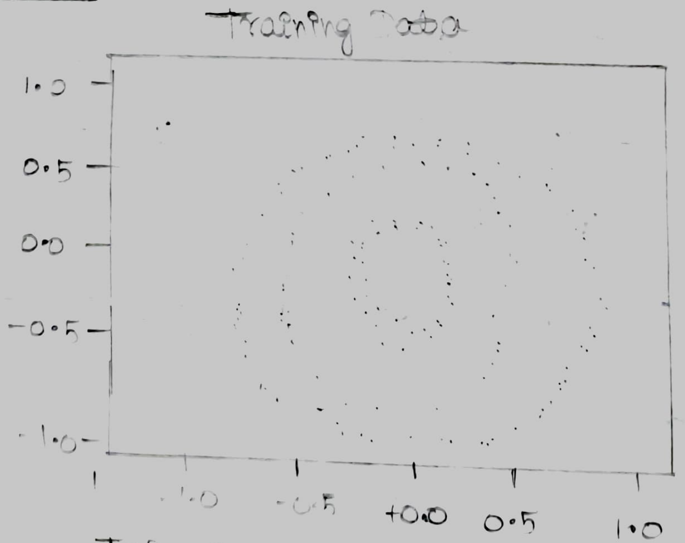
(vi) Check models performance on data.

PROGRAM:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import make_circles
from sklearn.neural_network import MLP clas
from sklearn.metrics import r2_score
    x_train, y_train = make_circles (n_sample
                                   = 700, noise = 0.05,
                                   factor=0.5)
    x_test, y_test = make_circles (n_sample=300,
                                   noise = 0.05, faster=0.
  sns. scatterplot (x = x_train [ : , 0], y=x_trai
                                                  [:,1], hue
                                                  =y,
     train, palette="viridis", style =y_train)
   plt.title ("Train Data")
   plt.show()
     clf = MLPclassifier (max_iter =1000)
     clf.fit (x_train, y_train)
     print (f "R2 score for training data = {
             clf score (x_train, y_train): .2f}")
     print (f" R2 score for test data =
                  {clf. score (x_test, y_test):
                                              .2f}")
```

```
Y_pred = clf.predict(X_test)

fig, ax = plt.subplots (1,2, figsize=(12,6))

sns.scatterplot (x=X_test [:,0], y= X_test [:,1],
        hue = Y_pred , palette = "viridis",
    style = Y_pred , ax = ax [0])
    ax [0].set_title ("Predicted data")

sns.scatterplot (X=X_test [:,0], Y=X_test [:,1],
            hue = Y_test , palette = "viridis",
    style =Y_test, ax=ax [1])
    ax [1].set_title ("Actual test data")

plt.tight_layout()
plt.show()
```

OUTPUT:



Train R2 : 1.00

Test R2 : 1.00

RESULT:

Thus, the implementation of artificial neutral networks using python classification is executed successfully and output is verified.