CO542 - NEURAL NETWORKS AND FUZZY SYSTEMS

MULTI LAYER PERCEPTRONS (MLP)

F.S.MARZOOK

E/16/232

LAB 04

09/10/2021

TASK 2

CODE

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.neural_network import MLPRegressor


# 1. Generate data (inputs and outputs) for the following function,
# y = -x^4 + x^3 + 23x^2 - 21x + 32
# x should be between -100 and 100; use an interval of 0.001 between the values.
x = np.arange(-100, 100, 0.001)
y = -np.power(x, 4) + np.power(x, 3) + (23 * np.power(x, 2)) - (21 * x) + 32
X = pd.DataFrame(data=x, columns=["column1"])
Y = pd.DataFrame(data=y, columns=["column1"])
# print(Y.head())


# 2. What is the number of inputs and outputs of the network?
print('number of inputs: ', X.shape[0])
print('number of outputs: ', Y.shape[0])


# 3. Model the MLP using MLPRegressor instead of MLPClassifier. (They are almost the same; except for very subtle differences.
# Try to find their differences).
X_train, X_test, y_train, y_test = train_test_split(X, Y, random_state=1, test_size=0.2)

sc_X = StandardScaler()
X_trainscaled = sc_X.fit_transform(X_train)
X_testscaled = sc_X.transform(X_test)

# model = MLPClassifier(hidden_layer_sizes=(256, 128, 64, 32), activation="relu", random_state=1)
model = MLPRegressor(hidden_layer_sizes=(256, 128, 64, 32), activation="relu", random_state=1)
print('******************************')
clf = model.fit(X_trainscaled, y_train.values.ravel())
print('******************************')
y_pred = clf.predict(X_testscaled)
print(y_pred)



# 4. Generate a test set (of your choice) and test it with the network.
# 5. Plot the train data and model predictions on a same plot and observe up to what extend the
# predicted values are fitting with the original data set.
```

2.

```
number of inputs:  200000
number of outputs:  200000
******************************
******************************
```

3 .MLPClassifier is used to classify inputs into distinct classes. It has discrete outputs.

MLPRegressor is used to get continuous outputs.