

Computational Intelligence – Assignment 3

Aim:

To Implement a Neuro-Fuzzy Inference system using Python, execute the code and upload the output snapshot in the Moodle with the code.

Program:

```
import anfis
import membership.mfDerivs
import membership.membershipfunction
import numpy

# numpy.loadtxt('c:\\Python_fiddling\\myProject\\MF\\trainingSet.txt', usecols=[1,2,3])
ts = numpy.loadtxt("trainingSet.txt", usecols=[1, 2, 3])
X = ts[:, 0:2]
Y = ts[:, 2]

mf = [['gaussmf', {'mean': 0., 'sigma': 1.}], ['gaussmf', {'mean': -1., 'sigma': 2.}], ['gaussmf', {'mean': -4., 'sigma': 10.}], ['gaussmf', {'mean': -7., 'sigma': 7.}]],
      [['gaussmf', {'mean': 1., 'sigma': 2.}], ['gaussmf', {'mean': 2., 'sigma': 3.}], ['gaussmf', {'mean': -2., 'sigma': 10.}], ['gaussmf', {'mean': -10.5, 'sigma': 5.}]]

mfc = membership.membershipfunction.MemFuncs(mf)

anf = anfis.ANFIS(X, Y, mfc)

anf.trainHybridJangOffLine(epochs=20)

print(round(anf.consequents[-1][0], 7))
print(round(anf.consequents[-2][0], 7))
print(round(anf.fittedValues[9][0], 7))

if round(anf.consequents[-1][0], 7) == -5.275538 and round(anf.consequents[-2][0], 6) == -1.990703 and round(anf.fittedValues[9][0], 6) == 0.002249:
    print('Test is good')

print("Error Plot")

anf.plotErrors()

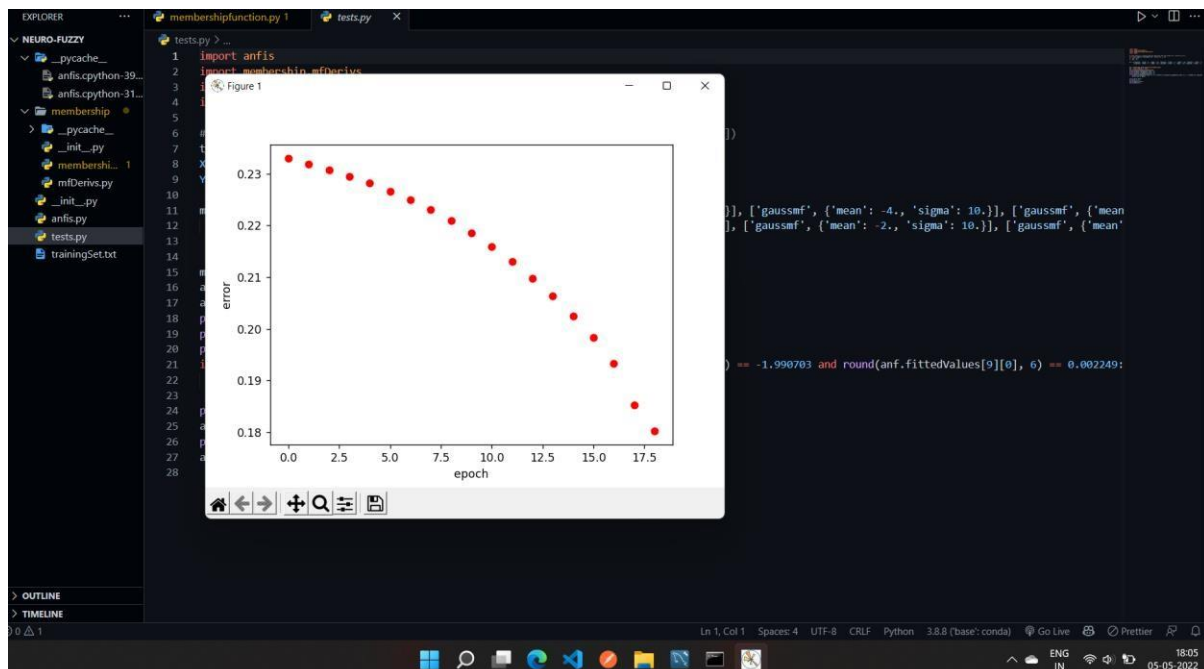
print("Results Plot")

anf.plotResults()
```

Output:

Error Plotted Graph

```
current error: 0.23296034910052635
current error: 0.231830463811818
current error: 0.23066704916231817
current error: 0.22947300226768125
current error: 0.22812866070176968
current error: 0.22661650417942178
current error: 0.22491793575814126
current error: 0.22301366839971112
current error: 0.22088433038073052
current error: 0.21851132183471828
current error: 0.2158778854344372
current error: 0.2129701563699756
current error: 0.20977743345759287
current error: 0.20628948581660947
current error: 0.20248430602957626
current error: 0.1982826228858566
current error: 0.1933472977835748
current error: 0.18530024116495003
current error: 0.18022727775340094
-0.0310883
0.0152347
-0.0088179
Error Plot
Results Plot
```



The screenshot displays a Jupyter Notebook environment with a file explorer on the left, a code editor in the center, and a plot window on the right.

File Explorer (Left): Shows a project structure with folders like `NEURO-FUZZY`, `membershipfunction.py`, `tests.py`, and `trainingSet.txt`.

Code Editor (Center): Contains a Python script for training a fuzzy inference system. The script imports `anfis`, `membership`, and `numpy`. It loads training data from `trainingSet.txt` and uses `anfis.trainHybridJangOffline` to train the model. The script also includes a test function and a plot function.

```

1 import anfis
2 import membership.mfDerivs
3 import membership.membershipfunction
4 import numpy
5
6 # nump.loadtxt('c:\Python_fiddling\myf
7 ts = numpy.loadtxt("trainingSet.txt", use
8 X = ts[:, 0:2]
9 Y = ts[:, 2]
10
11 mf = [[['gaussmf', {'mean': 0., 'sigma':
12        ['gaussmf', {'mean': 1., 'sigma':
13
14
15 mfc = membership.membershipfunction.MemFu
16 anf = anfis.ANfis(X, Y, mfc)
17 anf.trainHybridJangOffline(epochs=20)
18 print(round(anf.consequents[-1][0], 7))
19 print(round(anf.consequents[-2][0], 7))
20 print(round(anf.fittedValues[0][0], 7))
21 if round(anf.consequents[-1][0], 7) == 0.
22     print('Test is good')
23
24 print("Error Plot")
25 anf.plotErrors()
26 print("Results Plot")
27 anf.plotResults()
28

```

Plot Window (Right): Titled "Figure 1", it shows a line plot comparing the "trained" model (red line) and the "original" data (blue line). The x-axis ranges from 0 to 100, and the y-axis ranges from -0.2 to 1.0. The plot shows a highly oscillatory signal with several sharp peaks, particularly around x=60 and x=80.

Thus, implementation of a Neuro-Fuzzy Inference system using Python is executed and the code is verified