

Ex.No.04.

Back propagation in single neuron

Aim:

To write a python program to perform back propagation in single neuron.

Equipment's Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Moodle-Code Runner / Google Colab

Concept:

NUMPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

SK LEARN

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines.

MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using generalpurpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Algorithm:

- Start the program.
- Import libraries required as per requirement.
- Define neuron weight, eeta, i, y.
- summarize observations by class label
- summarize first few examples
- use for loop for range.
- stop the program.

Program:

```
#back propagation in single neuron

#BACK PROPOGATION IN SINGLE WINDOW

import numpy as np

i=1.5

w_o=0.0

y=0.5

#eeta

r=0.01

def dc_dw(a,y,i):

    dc_da=2*(a-y)

    da_dw=i

    return dc_da*da_dw

w=[w_o]

a=[w_o*i]

for j in range(0,100):

    a.append(w[-1]*i)

    w.append(w[-1]-r*dc_dw(a[-1],y,i))

    if(a[-1]-y)**2<0.001:

        break

print(a)

print(" ")

print(w)
```

Output:

```
[0.0, 0.0, 0.0225, 0.0439875, 0.06450806249999999, 0.08410519968749999,
0.10282046570156249, 0.12069354474499219, 0.13776233523146753,
0.1540630301460515, 0.1696301937894792, 0.18449683506895262,
0.19869447749084973, 0.2122532260037615, 0.22520183083359224,
0.23756774844608058, 0.24937719976600697, 0.26065522577653666,
0.2714257406165925, 0.28171158228884585, 0.29153456108584785,
0.3009155058369847, 0.30987430807432037, 0.31842996421097597,
0.32660061582148203, 0.33440358810951537, 0.3418554266445872,
0.3489719324455808, 0.3557681954855296, 0.36225862668868075,
0.36845698848769015, 0.3743764240057441, 0.3800294849254856,
0.3854281581038388, 0.390583890989166, 0.39550761589465355,
0.40020977317939416, 0.40470033338632144, 0.40898881838393697,
0.41308432155665975, 0.41699552708661014, 0.42073072836771264,
0.42429784559116557, 0.4277044425395631, 0.43095774262528275,
0.43406464420714497, 0.43703173521782346, 0.4398653071330214,
0.4425713683120355, 0.44515565673799384, 0.44762365218478417,
0.4499805878364689, 0.45223146138382775, 0.45438104562155546,
0.4564338985685854, 0.458394373132999, 0.4602666263420141,
0.46205462815662346, 0.4637621698895754, 0.4653928722445445,
0.46695019299354, 0.46843743430883067]
```

```
[0.0, 0.015, 0.029324999999999997, 0.043005375, 0.056070133125,
0.06854697713437499, 0.08046236316332812, 0.09184155682097836,
0.10270868676403433, 0.11308679585965278, 0.12299789004596841,
0.13246298499389983, 0.14150215066917435, 0.1501345538890615,
0.15837849896405373, 0.16625146651067133, 0.17377015051769112,
0.18095049374439504, 0.18780772152589725, 0.19435637405723188,
0.20061033722465646, 0.20658287204954692, 0.21228664280731732,
0.21773374388098804, 0.22293572540634357, 0.22790361776305812,
0.2326479549637205, 0.23717879699035307, 0.24150575112578718,
0.24563799232512676, 0.24958428267049607, 0.25335298995032374,
0.2569521054025592, 0.260389260659444, 0.26367174392976905,
0.26680651545292944, 0.26980022225754763, 0.27265921225595796,
0.27538954770443985, 0.2779970180577401, 0.28048715224514176,
0.2828652303941104, 0.2851362950263754, 0.2873051617501885,
0.28937642947143, 0.29135449014521564, 0.2932435380886809,
0.2950475788746903, 0.29677043782532925, 0.29841576812318943,
0.2999870585576459, 0.3014876409225518, 0.30292069708103697,
0.3042892657123903, 0.3055962487553327, 0.30684441756134273,
0.3080364187710823, 0.3091747799263836, 0.31026191482969634,
0.31130012866236, 0.3122916228725538, 0.3132384998432889]
```

Result:

Thus, the python program performed back propagation in single neuron.