Shri Ramdeobaba College of Engineering and Management
Nagpur, 440013

Department of Computer Science Engineering (AIML)

Deep Learning Lab

Name : *Shantanu Mane*
Roll No. : *E63*
Batch : *CSE-AIML*
Date : *12/3/2023*

AIM - To implement autoencoder for real valued data.

Importing Dependencies

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Autoencoder Class

```
import numpy as np


class AutoEncoders:
    def __init__(self):
        self.xi_hat = None
        self.xi_dash = None
        self.ho_hat = None
        self.h_o = None
        self.xi = np.array([0.1, 0.2, 0.3])
        self.W = np.random.randn(3, 2)
        self.W_dash = np.random.randn(2, 3)
        self.learning_rate = 5
        self.epochs = 1000
```

```
def ForwardPass(self):
    self.h_o = np.dot(self.xi, self.W)
    self.ho_hat = self.Sigmoid(self.h_o)
    self.xi_dash = np.dot(self.ho_hat, self.W_dash)
    self.xi_hat = self.Sigmoid(self.xi_dash)

def Sigmoid(self, x, derivative=False):
    if derivative:
        return np.exp(-x) / (1 + np.exp(-x)) ** 2
    return 1 / (1 + np.exp(-x))

def LossFunction(self, xi, xi_hat, derivative=False, lossType="Linear"):
    if lossType == 'Linear':
        pass
    if derivative:
        return 2 * (xi_hat - xi) / len(xi)
    else:
        return (xi_hat - xi) ** 2 / len(xi)

def BackwardPass(self, epoch):
    L_dash = self.LossFunction(self.xi, self.xi_hat)
    print(f"{epoch}. Loss = {L_dash.reshape(1, 3)},\nxi = {self.xi}, xi_hat = {self.xi_ha

    del_x_hat = L_dash * self.Sigmoid(self.xi_hat, derivative=True)

    L = np.dot(del_x_hat, self.W_dash.T)
    del_ho_hat = L * self.Sigmoid(self.ho_hat, derivative=True)

    self.W_dash -= self.learning_rate * np.dot(self.ho_hat.reshape(2, 1), del_x_hat.resha
    self.W -= self.learning_rate * np.dot(self.xi.reshape(3, 1), del_ho_hat.reshape(1, 2)

def Train(self):
    for epoch in range(self.epochs):
        self.ForwardPass()
        self.BackwardPass(epoch)

def Test(self):
    self.ForwardPass()
    print(f"xi_hat = {self.xi_hat}")
```

Main Function

```
from Autoencoders_From_Scratch import AutoEncoders
import numpy as np

if __name__ == '__main__':
    AE = AutoEncoders()

    AE.Train()
    # AE.Test()
```

Output

```
OUTPUT:
epoch 100. Loss = [[5.57726675e-05 2.35102033e-05 1.51284274e-05]],
xi = [0.1 0.2 0.3], xi_hat = [0.11293515 0.20839825 0.30673686]
```