```python
# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import plot

def sigmoid(x):
    y=1/(1+np.exp(-x))
    return y
def dsigmoid(x):
    y=(1-x)*x
    return y
# SGD (online training)
#DATA SET ------------------------------------------------------------------
#X=np.linspace(-2,2,9)
X=np.array([-2,-1.5,-1,-0.5, 0,0.5,1,1.5,2],dtype='float32')
#T=1+np.sin(X*np.pi/4)
T=np.array([0, 0.075,0.292,0.617,1.0, 1.382,1.707,1.923,2],dtype='float32')
# Initialize trainable parameters -------------------------------------------
W1=np.random.rand(2,1)
b1=np.random.rand(2,1)
W2=np.random.rand(1,2)
b2=np.random.rand(1)
alfa=0.3#learning rate
epoch=10

SSE=np.empty(epoch)
for k in range(epoch):   #stochastic gradient descent
    for i in range(X.size):
        #Forward propagation
        #layer 1
        y1=sigmoid( W1*X[i]+b1)
        #layer 2
        y2=np.matmul(W2,y1)+b2

        # Back propagation
        F2=1
        d2=-2*F2*(T[i]-y2 )

        F1=np.array([[ dsigmoid(y1[0]) , 0],
                     [0 , dsigmoid(y1[1]) ] ])

        d1= np.matmul(F1.astype('float32'), W2.reshape(2,1))*d2

        # update weights in layer2
        W2=W2-alfa*d2*y1.reshape(1,2) #y1'
        b2=b2-alfa*d2
        # update weights in layer1
        W1=W1-alfa*d1*X[i]
        b1=b1-alfa*d1
    #print loss at the end of the epoch
    #forward propagation
    err=0
    for i in range(len(X)):
        #layer 1
        Y1=sigmoid( W1*X[i]+b1)
        #layer 2
        Y2=np.matmul(W2,Y1)+b2
        err= err+(T[i]-Y2)**2
    SSE[k]=err

plt.figure(0)
plt.plot(range(epoch),SSE,'r-o')
plt.xlabel("iteration")
plt.ylabel("loss (SSE)")
print("W1=",W1,"\nW2",W2)
print("b1=",b1,"\nb2",b2)
#test network with trained weights
Y=np.empty(len(X))
for i in range(len(X)):
    #layer 1
    Y1=sigmoid( W1*X[i]+b1)
```

```
73        #layer 2
74        Y[i]=np.matmul(W2,Y1)+b2
75
76    plt.figure(1)
77    plt.plot(X,T,'b*')
78    plot(X,Y,'r-*')
79    plt.xlabel("IN")
80    plt.ylabel("OUT")
81    plt.legend(['target','predict'])
82
```