Lakehead UNIVERSITY

Masters in Computer Science

Topics in Machine Learning & Neural Net

(COMP-5011)

Name:

Jaykumar Nariya (1116571)

Assignment1:

1.(5 points) Repeat the computer experiment mentioned in the class, this time, however, positioning the two moons Figure to be on the edge of separability, that is, d=0. Determine the classification error rate produced by the algorithm over 2,000 test data points.

Code:

```
clear all;clc; % clear output

%============================== Variable Declaration============================================
[~, data] = halfmoon(10,6,0,3000); % taking data from halfmoon.m provided where halfmoon(rad =10,width=6,d=0,n_samp=1000+2000)
n = etaseries(0.9,1E-5,1000); % it will give list of learning rate where etaseries( startpoint, end point, No of rates)
bias = 5; % bias
weight    = [bias;zeros(2,1)];% initialize weights

%=============================Training Perceptron using generated data=============================
for epoch = 1:50, % No of Epoch is 50
    shuffle_training_data = data(:,randperm(1000)); % Shuffle data and take 1000 samples from 3000 as training
    miss = 0;
    for i = 1:1000, % for one epoch for no of instance
        X_train = [1 ; shuffle_training_data(1:2,i)]; % getting input X training data from dataset
        d = shuffle_training_data(3,i);         % getting true label from dataset
        Y_train = sigmoid(weight'*X_train); % find predicted value Y = sig(W.X)
        error(i) = d-Y_train; % find error e = ( d - Y)

        weight_update = weight + n(i)*(d-Y_train)*X_train; % Calculate update weight using W(n+1)= W(n) + eta.( d - Y). X
        weight = weight_update; % make update weight as weight W(n) = W(n+1)
        if (Y_train - shuffle_training_data(3,i)) ~= 0, % calculate error rate for training
            miss = miss + 1;
        end
    end

    mse(epoch) = mean(error.^2); % calculate Mean Square Error per epoch
    Accuracy = ((1000-miss)/1000)*100; % calculate Training Accuracy
    fprintf(' For epoch %f Training Accuracy is %f \n',epoch, Accuracy);
    fprintf(' For epoch %f Training Error is %f \n',epoch, 100- Accuracy);
end
```

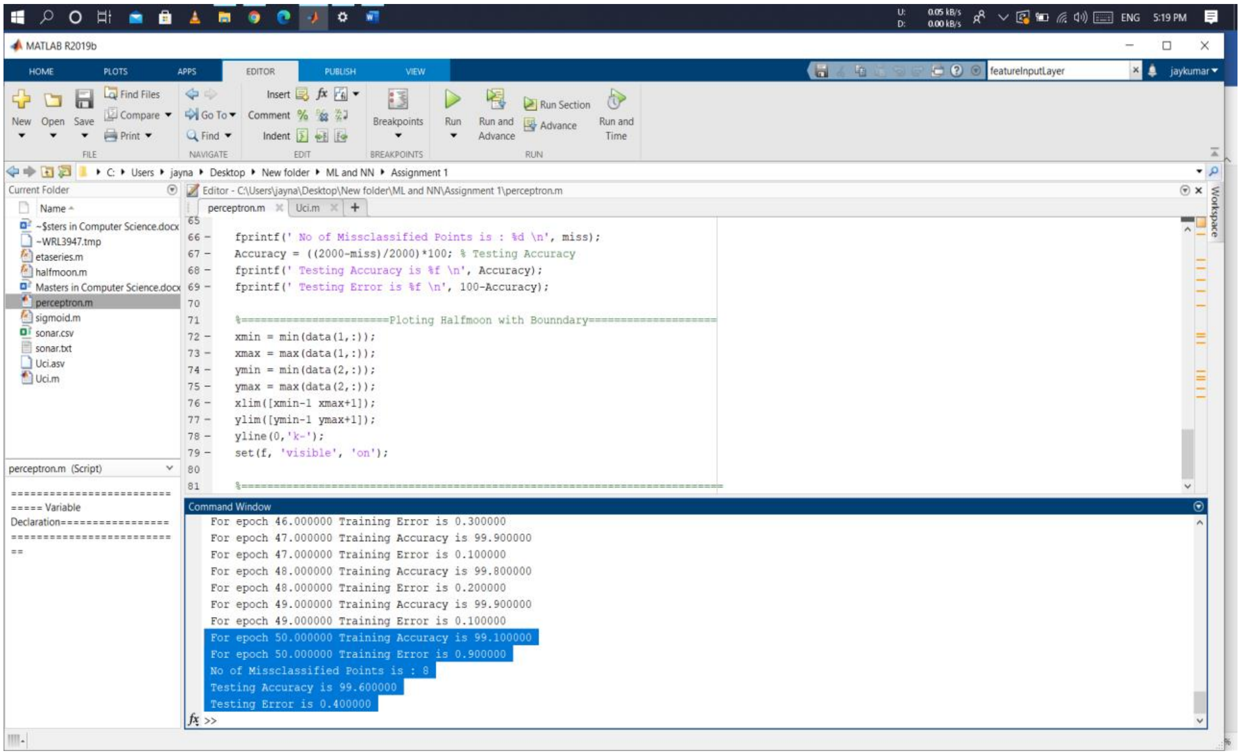
```
%=======================Ploating Data Points For Training Samples=========================

f = figure('visible','off');
hold on;
for i=1:1000,
    if shuffle_training_data(3,i) == 1,
        plot(shuffle_training_data(1,i),shuffle_training_data(2,i),'r+');
    else,
        plot(shuffle_training_data(1,i),shuffle_training_data(2,i),'kx');
    end
end



%===================Testing Dataset Using Trained Perceptron=====================
miss = 0;

for i = 1 : 2000, % for 2000 samples testing perceptron
    X_test = [1 ; data(1:2,i+1000)]; % getting input X testing data from dataset
    Y_test(i) = sigmoid(weight'*X_test); % find predicted value Y = sig(W.X)
    if Y_test(i) == 1 , % plot data if predicted label is 1
        plot(data(1,i+1000),data(2,i+1000),'r+');
    end
    if Y_test(i) == -1, % plot data if predicted label is -1
        plot(data(1,i+1000),data(2,i+1000),'kx');
    end
     if (Y_test(i) - data(3,i+1000)) ~=0,% Find error using ( Y pred - Y true)
        miss = miss + 1;
     end
end




fprintf(' No of Missclassified Points is : %d \n', miss);
Accuracy = ((2000-miss)/2000)*100; % Testing Accuracy
fprintf(' Testing Accuracy is %f \n', Accuracy);
fprintf(' Testing Error is %f \n', 100-Accuracy);

%=======================Ploting Halfmoon with Bounndary====================
xmin = min(data(1,:));
xmax = max(data(1,:));
ymin = min(data(2,:));
ymax = max(data(2,:));
```

```
xlim([xmin-1 xmax+1]);
ylim([ymin-1 ymax+1]);
yline(0,'k-');
set(f, 'visible', 'on');

%======================================================================
======
```

Results:

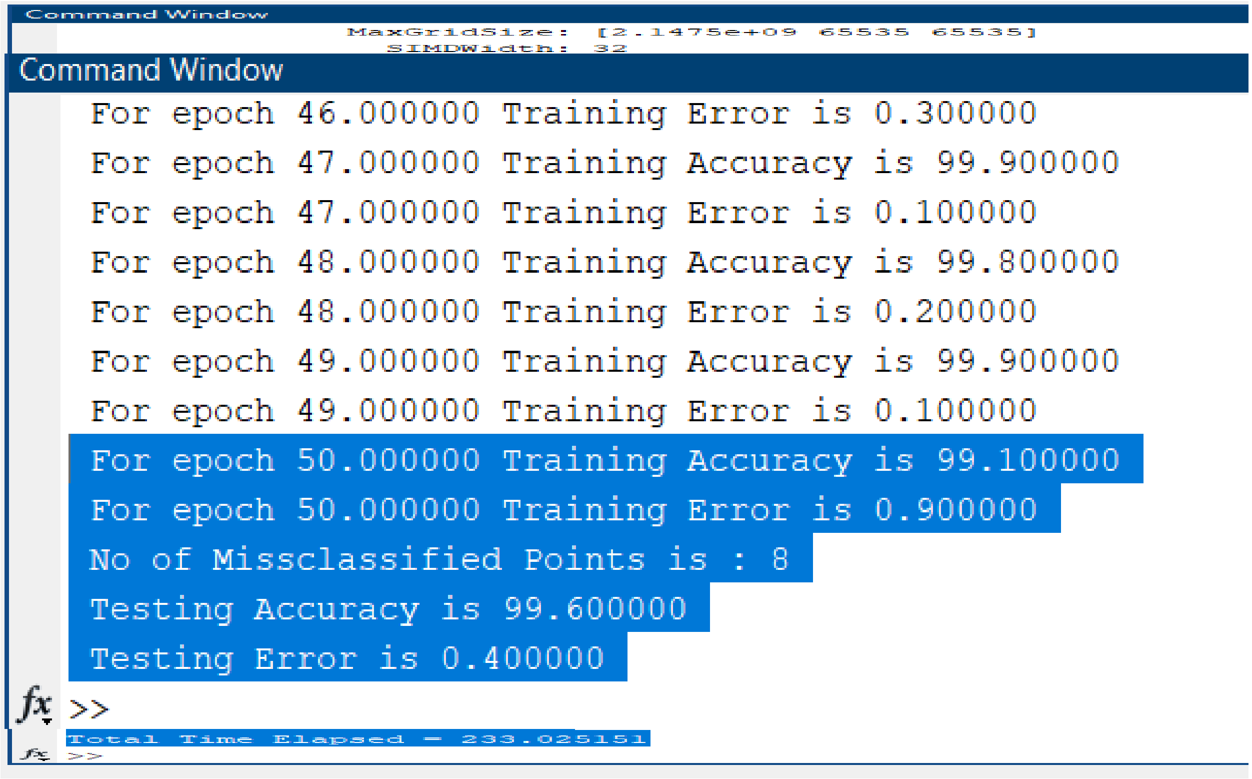
- **Training Accuracy:** **99.10%**
- **Training Error:** **0.9%**
- **Testing Accuracy:** **99.6%**
- **Testing Error:** **0.4%**
- **No of Missclassified Point:** **8**

Outputs:



```
fprintf(' No of Missclassified Points is : %d \n', miss);
Accuracy = ((2000-miss)/2000)*100; % Testing Accuracy
fprintf(' Testing Accuracy is %f \n', Accuracy);
fprintf(' Testing Error is %f \n', 100-Accuracy);

%======================Ploting Halfmoon with Bounndary====================
xmin = min(data(1,:));
xmax = max(data(1,:));
ymin = min(data(2,:));
ymax = max(data(2,:));
xlim([xmin-1 xmax+1]);
ylim([ymin-1 ymax+1]);
yline(0,'k-');
set(f, 'visible', 'on');

%=========================================================================
```

```
For epoch 46.000000 Training Error is 0.300000
For epoch 47.000000 Training Accuracy is 99.900000
For epoch 47.000000 Training Error is 0.100000
For epoch 48.000000 Training Accuracy is 99.800000
For epoch 48.000000 Training Error is 0.200000
For epoch 49.000000 Training Accuracy is 99.900000
For epoch 49.000000 Training Error is 0.100000
For epoch 50.000000 Training Accuracy is 99.100000
For epoch 50.000000 Training Error is 0.900000
No of Missclassified Points is : 8
Testing Accuracy is 99.600000
Testing Error is 0.400000
fx >>
```



```
        MaxGridSize: [2.1475e+09 65535 65535]
          SIMDWidth: 32
```

```
For epoch 46.000000 Training Error is 0.300000
For epoch 47.000000 Training Accuracy is 99.900000
For epoch 47.000000 Training Error is 0.100000
For epoch 48.000000 Training Accuracy is 99.800000
For epoch 48.000000 Training Error is 0.200000
For epoch 49.000000 Training Accuracy is 99.900000
For epoch 49.000000 Training Error is 0.100000
For epoch 50.000000 Training Accuracy is 99.100000
For epoch 50.000000 Training Error is 0.900000
No of Missclassified Points is : 8
Testing Accuracy is 99.600000
Testing Error is 0.400000
fx >>
Total Time Elapsed = 233.025151
fx >>
```

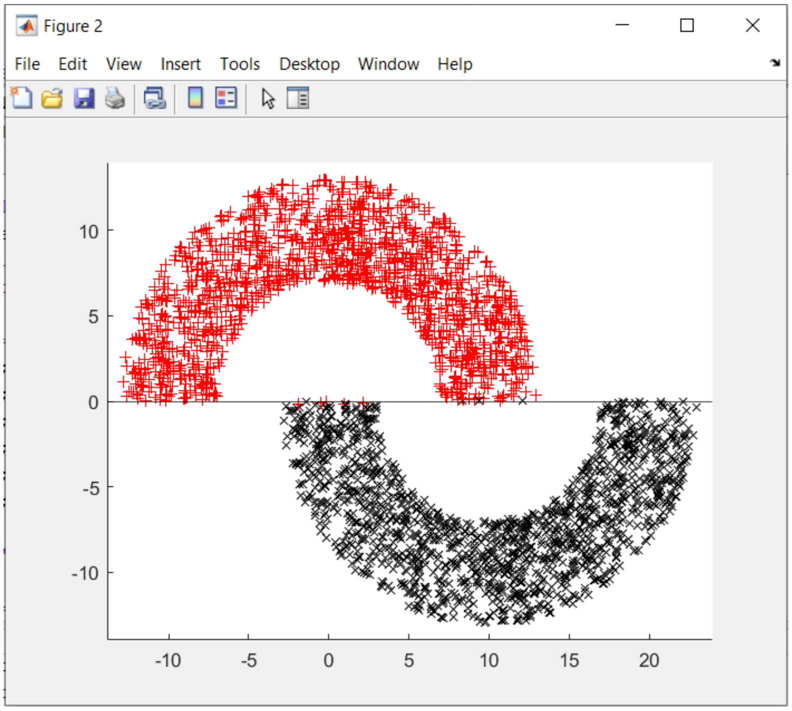


Figure 2
File Edit View Insert Tools Desktop Window Help
10
5
0
-5
-10
-10
-5
0
5
10
15
20

2.(5 points) Download one of the UCI dataset, reuse your own perceptron codes to get the testing accuracy of the selected dataset. The UCI dataset is available at https://archive.ics.uci.edu/ml/datasets.php?format=&task=cla&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=tableCode

Code:

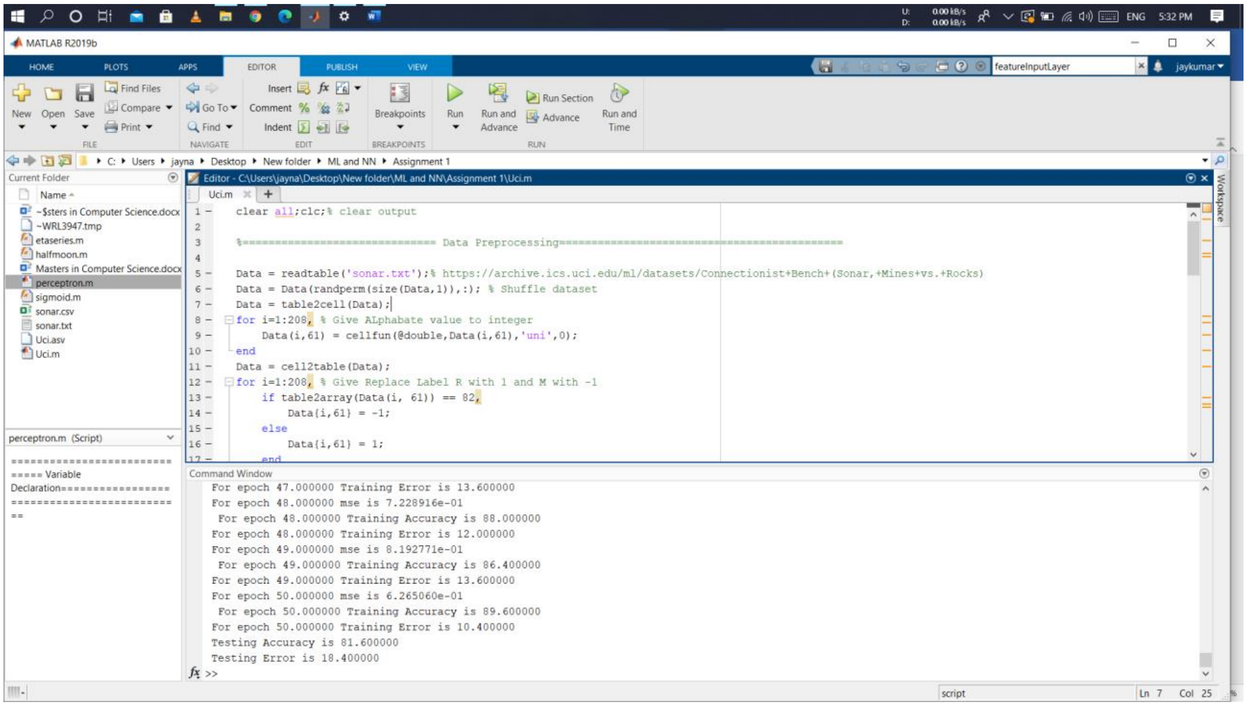
```
clear all;clc;% clear output

%============================== Data Prepro-
cessing============================================

Data = readtable('sonar.txt');% https://archive.ics.uci.edu/ml/da-
tasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)
Data = Data(randperm(size(Data,1)),:); % Shuffle dataset
Data = table2cell(Data);
for i=1:208, % Give ALphabate value to integer
    Data(i,61) = cellfun(@double,Data(i,61),'uni',0);
end
Data = cell2table(Data);
for i=1:208, % Give Replace Label R with 1 and M with -1
    if table2array(Data(i, 61)) == 82,
        Data{i,61} = -1;
    else
        Data{i,61} = 1;
    end
end
Data = Data{:,:};

%============================== Variable Declara-
tion============================================

n = etaseries(0.9,1E-5,83);  % it will give list of learning rate
where etaseries( startpoint, end point, No of rates)
weight    = [1;zeros(60,1)];
test_data = Data(1:125,1:61); % Taking 60% data as testing
test_data =test_data';
tran_data = Data(126:208,1:61); % Taking 40% data as trainging
tran_data = tran_data';
Data = Data';

%=============================Training Perceptron using generated
data=============================
for epoch = 1:50, % No of Epoch is 50
    miss = 0;
    for i = 1:83, % for one epoch for no of instance 83
        X_train = [1;tran_data(1:60,i)]; % getting input X training
data from dataset
        d = tran_data(61,i); % getting true label from dataset
        Y_train = sigmoid(weight'*X_train); % find predicted value Y =
sig(W.X)
        error(i) = d-Y_train; % find error e = ( d - Y)
```

```
weight_update = weight + n(i)*(d-Y_train)*X_train;  % Calculate update weight using W(n+1)= W(n) + eta.( d - Y). X
        weight = weight_update; % make update weight as weight W(n) = W(n+1)

        if (Y_train - tran_data(61,i)) ~= 0, % calculate error rate for training
        miss = miss + 1;
    end

    end
    mse(epoch) = mean(error.^2);
    fprintf(' For epoch %f mse is %d \n ',epoch,mse(epoch));
    Accuracy = ((125-miss)/125)*100; % calculate Mean Square Error per epoch
    fprintf(' For epoch %f Training Accuracy is %f \n',epoch, Accuracy);  % calculate Training Accuracy
    fprintf(' For epoch %f Training Error is %f \n',epoch, 100- Accuracy);
end

%==================Testing Dataset Using Trained Perceptron=====================

miss=0;
for i = 1 : 125, % for 125 samples testing perceptron
    X_test = [1;test_data(1:60,i)]; % getting input X testing data from dataset
    Y_test(i) = sigmoid(weight'*X_test); % find predicted value Y = sig(W.X)
    if (Y_test(i) - test_data(61,i)) ~= 0, % Find error using ( Y pred - Y true)
        miss = miss + 1;
    end

end

Accuracy = ((125-miss)/125)*100;
fprintf(' Testing Accuracy is %f \n', Accuracy); % Testing Accuracy
fprintf(' Testing Error is %f \n', 100-Accuracy);
```

Results:

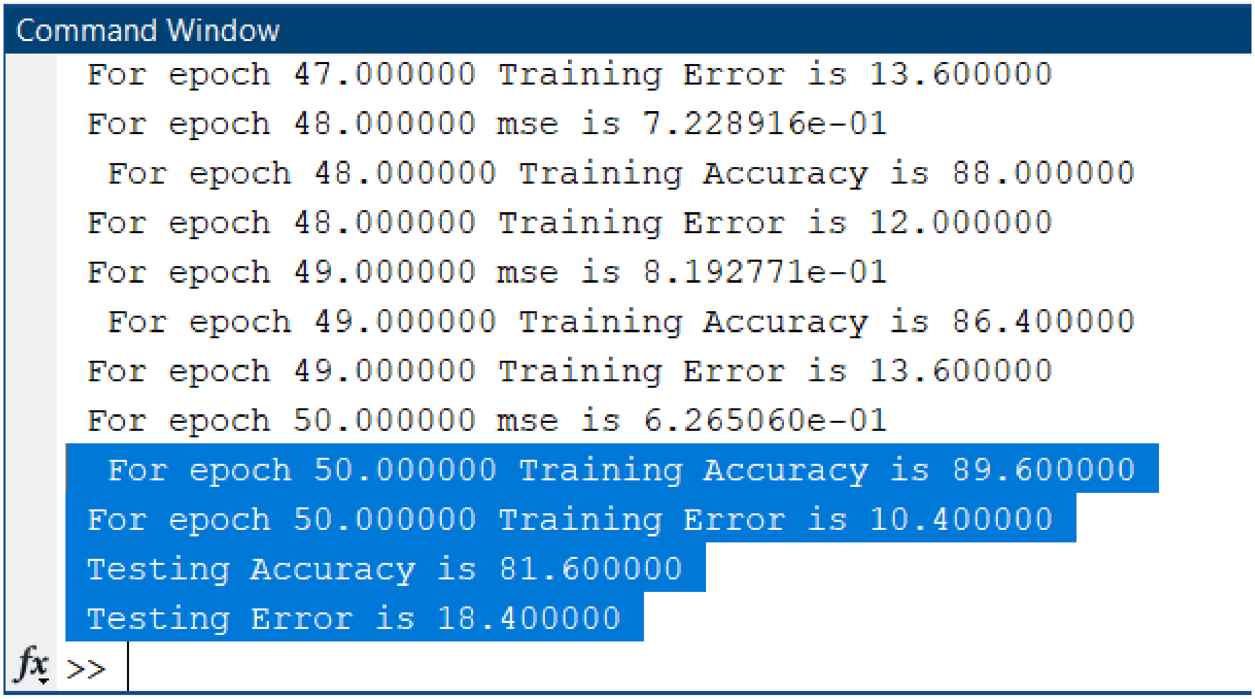
- **Training Accuracy: 89.60%**
- **Training Error: 10.40%**
- **Testing Accuracy: 81.6%**
- **Testing Error: 18.4%**

Outputs:



```
clear all;clc;% clear output

%============================= Data Preprocessing=========================================

Data = readtable('sonar.txt');% https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)
Data = Data(randperm(size(Data,1)),:); % Shuffle dataset
Data = table2cell(Data);
for i=1:208, % Give ALphabate value to integer
    Data(i,61) = cellfun(@double,Data(i,61),'uni',0);
end
Data = cell2table(Data);
for i=1:208, % Give Replace Label R with 1 and M with -1
    if table2array(Data(i, 61)) == 82,
        Data{i,61} = -1;
    else
        Data{i,61} = 1;
    end
```

```
For epoch 47.000000 Training Error is 13.600000
For epoch 48.000000 mse is 7.228916e-01
 For epoch 48.000000 Training Accuracy is 88.000000
For epoch 48.000000 Training Error is 12.000000
For epoch 49.000000 mse is 8.192771e-01
 For epoch 49.000000 Training Accuracy is 86.400000
For epoch 49.000000 Training Error is 13.600000
For epoch 50.000000 mse is 6.265060e-01
 For epoch 50.000000 Training Accuracy is 89.600000
For epoch 50.000000 Training Error is 10.400000
Testing Accuracy is 81.600000
Testing Error is 18.400000
fx >>
```



```
For epoch 47.000000 Training Error is 13.600000
For epoch 48.000000 mse is 7.228916e-01
 For epoch 48.000000 Training Accuracy is 88.000000
For epoch 48.000000 Training Error is 12.000000
For epoch 49.000000 mse is 8.192771e-01
 For epoch 49.000000 Training Accuracy is 86.400000
For epoch 49.000000 Training Error is 13.600000
For epoch 50.000000 mse is 6.265060e-01
 For epoch 50.000000 Training Accuracy is 89.600000
For epoch 50.000000 Training Error is 10.400000
Testing Accuracy is 81.600000
Testing Error is 18.400000
fx >>
```

Reference:

- https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)