Name: Xinrun Zhang

ML Homework 5 Report

Instructor: Prof. Kaliappan Gopalan

Time: 23/03/2019

1. Files

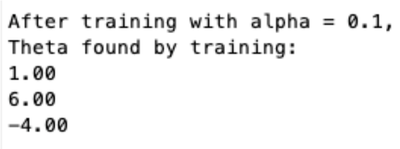
(1) main_part1.m – Includes the main program of the homework 5 part 1.

(2) main_part2.m – Includes the main program of the homework 5 part 2.

(3) trainingNueron.m – This function is used to train the neuron.

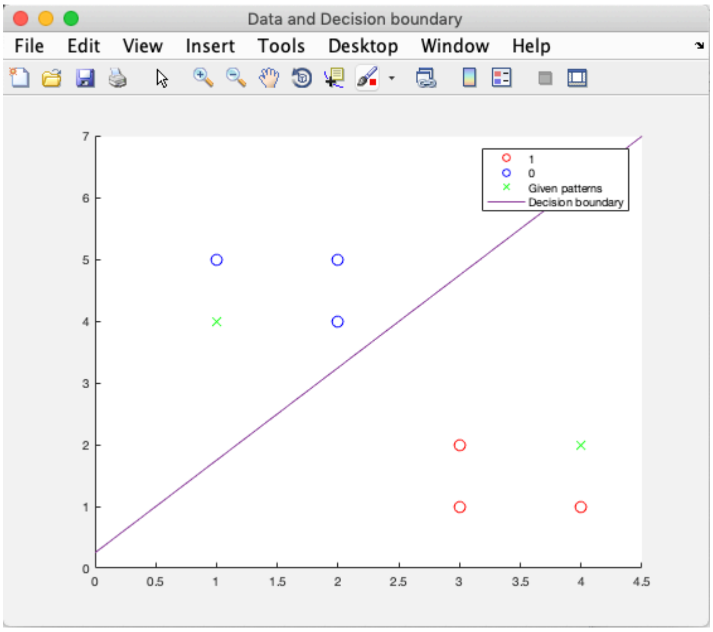
2. Part one

(1) Theta found by training:

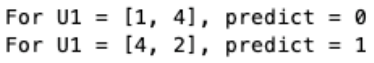


```
After training with alpha = 0.1,
Theta found by training:
1.00
6.00
-4.00
```

(2) Training data with given data and decision boundary:



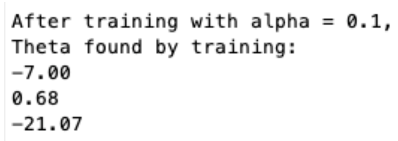
(3)　Predict the given patterns



```
For U1 = [1, 4], predict = 0
For U1 = [4, 2], predict = 1
```

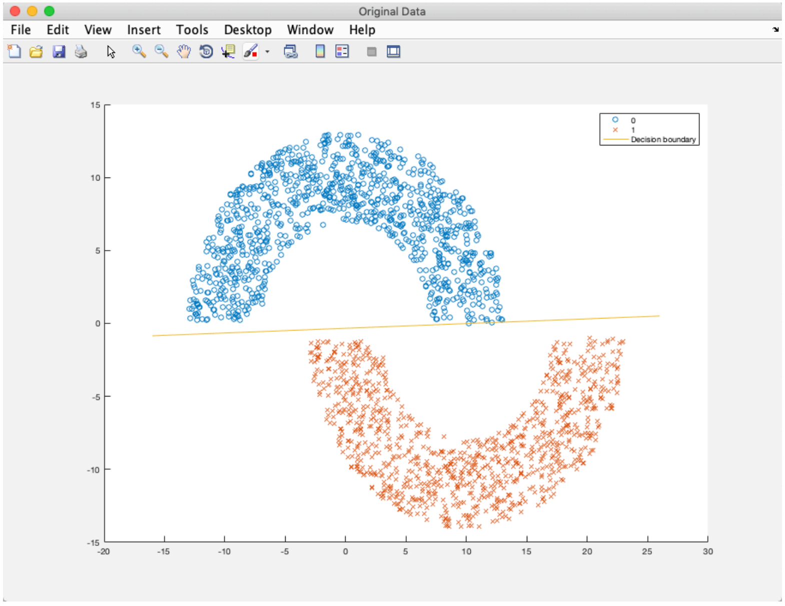
3.　Part two

(1)　Theta found by training:

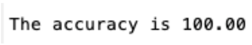


```
After training with alpha = 0.1,
Theta found by training:
-7.00
0.68
-21.07
```

(2)　Training data and decision boundary:



(3)　Validation accuracy:



```
The accuracy is 100.00
```

4.　Codes

(1)　main_part1.m

```
%% Machine Learning Homework 5 part 1

% Author: Xinrun Zhang

% Time: 03/23/2019 17:20
% =====================================================================

%% Initialization
clear ; close all; clc

fprintf('Initializing...\n')
% Initial the data
p = [3 1; 3 2; 4 1; 1 5; 2 4; 2 5];
t = [1; 1; 1; 0; 0; 0];

% Initial the theta vector
theta = [0; 0; 0];

% Initial the learning rate and iteration times
alpha = 1;
iteration = 10;

% Data processing
fprintf('Data processing...\n\n')
p = [ones(6, 1), p(:,1:2)];
% =====================================================================

%% Training the neuron
fprintf('Start training the neuron...\n')
i = 0;
for i = 1:iteration
    theta = trainingNueron( theta, p, t, alpha);
end

fprintf('\nAfter training with alpha = 0.1, ')
fprintf('\nTheta found by training:\n');
fprintf('%.2f\n', theta);
fprintf('\n')
```

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

%% Prediction
u1 = [1; 1; 4];
u2 = [1; 4; 2];

predict1 = round(logsig( u1'*theta ));
predict2 = round(logsig( u2'*theta ));

fprintf('For U1 = [%d, %d], predict = %d\n',u1(2), u1(3), predict1);
fprintf('For U1 = [%d, %d], predict = %d\n',u2(2), u2(3), predict2);
% ======================================================================

%% Plot
x1 = [3; 3; 4]; y1 = [1; 2; 1];
x2 = [1; 2; 2]; y2 = [5; 4; 5];
x3 = [1; 4];  y3 = [4; 2];
m = 0:0.1:4.5;
n = 1.5*m + 0.25;

figure('Name','Data and Decision boundary','NumberTitle','off');
scatter(x1, y1, 80, 'o', 'r');
hold on;
scatter(x2, y2, 80, 'o', 'b');
hold on;
scatter(x3, y3, 80, 'x', 'g')
hold on;
plot(m, n);
hold off;
legend('1', '0', 'Given patterns', 'Decision boundary');
% ======================================================================
```

(2) main_part2.m

```
%% Machine Learning Homework 5 part 2
% Author: Xinrun Zhang
% Time: 03/23/2019 21:07
% =====================================================================

%% Initialization
clear ; close all; clc

fprintf('Initializing...\n')
% Initial the data
data = importdata('halfmoon.mat'); % don't use load function
x = data(:,[1, 2]);
y = data(:, 3);
data_val = importdata('halfmoonTest.mat');
x_val = data_val(:,[1, 2]);
y_val = data_val(:, 3);

% Initial the theta vector
theta = [1; 1; 1];

% Initial the learning rate and iteration times
alpha = 1;
iteration = 10;

% Data processing
fprintf('Data processing...\n')
X = [ones(2000, 1), x(:,1:2)];
X_val = [ones(240, 1), x_val(:, 1:2)];
% =====================================================================

%% Training the neuron
fprintf('Start training the neuron...\n\n')
```

```
i = 0;
for i = 1:iteration
    theta = trainingNueron( theta, X, y, alpha);
end

fprintf('After training with alpha = 0.1,\n')
fprintf('Theta found by training:\n');
fprintf('%.2f\n', theta);
fprintf('---------------------------------------------------------\n');
% =====================================================================

%% Plot the original data
fprintf('Plotting the data...\n')
x_0 = x(1:1000, [1, 2]);
x_1 = x(1001:2000, [1, 2]);
m = -16:0.1:26;
n = 0.0323*m - 0.3322;

figure('Name','Original Data','NumberTitle','off');
scatter(x_0(:, 1), x_0(:, 2), 'o');
hold on;
scatter(x_1(:, 1), x_1(:, 2), 'x');
hold on;
plot(m, n);
legend('0', '1', 'Decision boundary');
fprintf('---------------------------------------------------------\n');
% =====================================================================

%% Validation
predict = round(logsig(X_val*theta));
accuracy = mean( double(predict == y_val) * 100);
fprintf('The accuracy is %.2f\n', accuracy);
fprintf('---------------------------------------------------------\n');
```

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

（3） trainingNueron.m

```
function theta = trainingNueron( theta, p, t, alpha )

iter = size(t);

for i = 1:iter

    h = round(logsig(p(i, :)*theta));

    error = t(i) - h;

    theta = theta + alpha * p(i, :)' * error;

end


end
```