## **Test problem validation from the book**

clc

clear all

% Load input data and target values

input\_data = readmatrix('validation/input\_p.txt');

target\_data = readmatrix('validation/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 1; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

validation\_perceptron = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[validation\_perceptron, stop\_code] = validation\_perceptron.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = 'validation/w.txt';

b\_file = 'validation/b.txt';

% Write weights & bias to file

writematrix(validation\_perceptron.Weights, w\_file, 'Delimiter', 'tab');

writematrix(validation\_perceptron.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(validation\_perceptron.Weights);

disp('Bias:');

disp(validation\_perceptron.Bias);

% Validate results

validation\_perceptron.classify(input\_data(1,:)') % 1

validation\_perceptron.classify(input\_data(2,:)') % 0

validation\_perceptron.classify(input\_data(3,:)') % 0

## Example 1 classification 2 classes

clc

clear all

% Load input data and target values

input\_data = readmatrix('1/input\_p.txt');

target\_data = readmatrix('1/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 1; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

perceptron1 = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[perceptron1, stop\_code] = perceptron1.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = '1/w.txt';

b\_file = '1/b.txt';

% Write weights & bias to file

writematrix(perceptron1.Weights, w\_file, 'Delimiter', 'tab');

writematrix(perceptron1.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(perceptron1.Weights);

disp('Bias:');

disp(perceptron1.Bias);

% Validate results

perceptron1.classify(input\_data(1,:)') % 1

perceptron1.classify(input\_data(2,:)') % 1

perceptron1.classify(input\_data(3,:)') % 1

perceptron1.classify(input\_data(4,:)') % 0

perceptron1.classify(input\_data(5,:)') % 0

perceptron1.classify(input\_data(6,:)') % 0

## Example 2 classification 2 classes

From book P4.1

clc

clear all

% Load input data and target values

input\_data = readmatrix('2/input\_p.txt');

target\_data = readmatrix('2/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 1; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

perceptron2 = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[perceptron2, stop\_code] = perceptron2.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = '2/w.txt';

b\_file = '2/b.txt';

% Write weights & bias to file

writematrix(perceptron2.Weights, w\_file, 'Delimiter', 'tab');

writematrix(perceptron2.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(perceptron2.Weights);

disp('Bias:');

disp(perceptron2.Bias);

% Validate results

perceptron2.classify(input\_data(1,:)') % 1

perceptron2.classify(input\_data(2,:)') % 0

perceptron2.classify(input\_data(3,:)') % 0

perceptron2.classify(input\_data(4,:)') % 0

perceptron2.classify(input\_data(5,:)') % 0

perceptron2.classify(input\_data(6,:)') % 1

perceptron2.classify(input\_data(7,:)') % 1

perceptron2.classify(input\_data(8,:)') % 1

## Example 3 classification 2 classes

From book P4.4

clc

clear all

% Load input data and target values

input\_data = readmatrix('3/input\_p.txt');

target\_data = readmatrix('3/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 1; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

perceptron3 = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[perceptron3, stop\_code] = perceptron3.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = '3/w.txt';

b\_file = '3/b.txt';

% Write weights & bias to file

writematrix(perceptron3.Weights, w\_file, 'Delimiter', 'tab');

writematrix(perceptron3.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(perceptron3.Weights);

disp('Bias:');

disp(perceptron3.Bias);

% Validate results

perceptron3.classify(input\_data(1,:)') % 0

perceptron3.classify(input\_data(2,:)') % 1

perceptron3.classify(input\_data(3,:)') % 0

perceptron3.classify(input\_data(4,:)') % 1

perceptron3.classify(input\_data(5,:)') % 0

perceptron3.classify(input\_data(6,:)') % 1

## Example 4 classification 4 classes

From book P4.5

clc

clear all

% Load input data and target values

input\_data = readmatrix('4/input\_p.txt');

target\_data = readmatrix('4/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 2; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

perceptron4 = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[perceptron4, stop\_code] = perceptron4.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = '4/w.txt';

b\_file = '4/b.txt';

% Write weights & bias to file

writematrix(perceptron4.Weights, w\_file, 'Delimiter', 'tab');

writematrix(perceptron4.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(perceptron4.Weights);

disp('Bias:');

disp(perceptron4.Bias);

% Validate results

perceptron4.classify(input\_data(1,:)') % 0,0

perceptron4.classify(input\_data(2,:)') % 0,0

perceptron4.classify(input\_data(3,:)') % 0,1

perceptron4.classify(input\_data(4,:)') % 0,1

perceptron4.classify(input\_data(5,:)') % 1,0

perceptron4.classify(input\_data(6,:)') % 1,0

perceptron4.classify(input\_data(7,:)') % 1,1

perceptron4.classify(input\_data(8,:)') % 1,1

## Example 5 classification 4 classes

clc

clear all

% Load input data and target values

input\_data = readmatrix('5/input\_p.txt');

target\_data = readmatrix('5/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 2; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

perceptron5 = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[perceptron5, stop\_code] = perceptron5.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = '5/w.txt';

b\_file = '5/b.txt';

% Write weights & bias to file

writematrix(perceptron5.Weights, w\_file, 'Delimiter', 'tab');

writematrix(perceptron5.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(perceptron5.Weights);

disp('Bias:');

disp(perceptron5.Bias);

% Validate results

perceptron5.classify(input\_data(1,:)') % 0,0

perceptron5.classify(input\_data(2,:)') % 0,0

perceptron5.classify(input\_data(3,:)') % 0,0

perceptron5.classify(input\_data(4,:)') % 1,0

perceptron5.classify(input\_data(5,:)') % 1,0

perceptron5.classify(input\_data(6,:)') % 1,0

perceptron5.classify(input\_data(7,:)') % 0,1

perceptron5.classify(input\_data(8,:)') % 0,1

perceptron5.classify(input\_data(9,:)') % 0,1

perceptron5.classify(input\_data(10,:)') % 1,1

perceptron5.classify(input\_data(11,:)') % 1,1

perceptron5.classify(input\_data(12,:)') % 1,1

Example 6 classification 4 classes

clc

clear all

% Load input data and target values

input\_data = readmatrix('6/input\_p.txt');

target\_data = readmatrix('6/target\_t.txt');

% Format data into prototypes

prototypes = cell(size(input\_data, 1), 1);

for i = 1:size(input\_data, 1)

prototypes{i} = {input\_data(i, :)', target\_data(i, :)'};

end

% Hyperparameters

s = 2; % Number of neurons (2^s to get number of classes)

input\_size = size(input\_data, 2);

max\_epochs = 100; % Maximum number of epochs

% Create and train the perceptron network

stop\_code = 0;

ntry = 0;

number\_of\_retries = 1; % If training faill retry n times

perceptron5 = perceptronNetwork(s, input\_size);

while (stop\_code ~= 1 && ntry < number\_of\_retries)

[perceptron5, stop\_code] = perceptron5.train(prototypes, max\_epochs);

ntry = ntry + 1;

end

% Save final weights and bias to files

w\_file = '6/w.txt';

b\_file = '6/b.txt';

% Write weights & bias to file

writematrix(perceptron5.Weights, w\_file, 'Delimiter', 'tab');

writematrix(perceptron5.Bias, b\_file, 'Delimiter', 'tab');

% Display weights & bias

disp('Weights:');

disp(perceptron5.Weights);

disp('Bias:');

disp(perceptron5.Bias);

% Validate results

perceptron5.classify(input\_data(1,:)') % 0,0

perceptron5.classify(input\_data(2,:)') % 0,0

perceptron5.classify(input\_data(3,:)') % 0,0

perceptron5.classify(input\_data(4,:)') % 0,1

perceptron5.classify(input\_data(5,:)') % 0,1

perceptron5.classify(input\_data(6,:)') % 0,1

perceptron5.classify(input\_data(7,:)') % 1,0

perceptron5.classify(input\_data(8,:)') % 1,0

perceptron5.classify(input\_data(9,:)') % 1,0

perceptron5.classify(input\_data(10,:)') % 1,1

perceptron5.classify(input\_data(11,:)') % 1,1

perceptron5.classify(input\_data(12,:)') % 1,1