ONE LAYER PERCEPTRON

```
% One Layer Perceptron
clear;
clc;
% Reading the data
training_set = readmatrix('training_set.csv');
validation_set = readmatrix('validation_set.csv');

% Normalizing the data
training_first_input_norm = normalize(training_set(:,1));
training_second_input_norm = normalize(training_set(:,2));
validation_first_input_norm =
normalize(validation_set(:,1));
validation_second_input_norm =
normalize(validation_set(:,2));


x_training = [training_first_input_norm,
training_second_input_norm];
x_validation = [validation_first_input_norm,
validation_second_input_norm];
target_training = training_set(:, 3);
target_validation = validation_set(:, 3);

nInput = 2; % no of input neurons
nHidden = 30; % no of hidden neurons
nVisible = 1; % no of output neurons
nTrials = 10000000;
eta = 0.01;
%Initializing the weights and the thresholds
W1 = randn(nHidden, nInput);
W2 = randn(nVisible, nHidden);
theta1 = zeros(nHidden, 1);
theta2 = zeros(nVisible, 1);


deltaW1 = zeros(nHidden, nInput);
deltaW2 = zeros(nVisible, nHidden);
deltaTheta1 = zeros(nHidden, 1);
deltaTheta2 = zeros(nVisible, 1);

% Training
for ntrials = 1:nTrials
```

```
    mu = randi(length(x_training));
    x =  x_training(mu, :)';
    t = training_set(mu,3);

    localField1 = (W1*x) - theta1;
    output_Hidden = tanh(localField1);

    localField2 = (W2*output_Hidden) - theta2;
    output_Visible = tanh(localField2);

    output_error2 = (t - output_Visible)*(1-
tanh(localField2)^2);
    deltaW2 = transpose(eta*(output_error2*output_Hidden));
    deltaTheta2 = -eta*( output_error2);

    output_error1 = output_error2*W2'.*(1-
tanh(localField1).^2);
    deltaW1 = eta*(output_error1*x');
    deltaTheta1 = -eta*( output_error1);

    W1 = W1 + deltaW1;
    W2 = W2 + deltaW2;
    theta1 = theta1 + deltaTheta1;
    theta2 = theta2 + deltaTheta2;
end
error = 0;
for i =1 : length(x_validation)
    x =  x_validation(i, :)';
    t = validation_set(i,3);

    localField1 = (W1*x) - theta1;
    output_Hidden = tanh(localField1);

    localField2 = (W2*output_Hidden) - theta2;
    output_Visible = tanh(localField2);

    error = error + abs(t-sign(output_Visible));
end
classification_error = error/(2*length(x_validation))
csvwrite('w1.csv', W1);
csvwrite('w2.csv', W2');
csvwrite('t1.csv', theta1);
csvwrite('t2.csv', theta2);
```