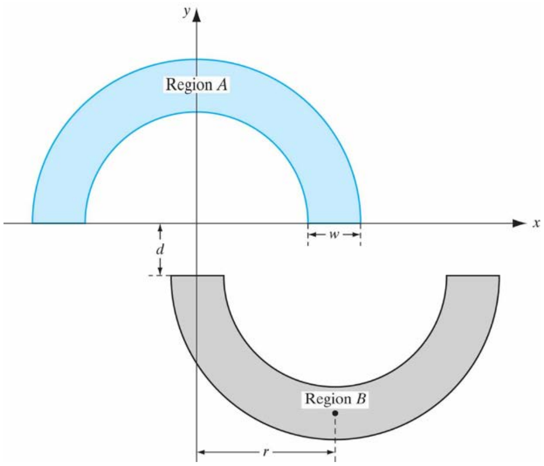
Project 1. Use a multilayer perceptron network trained by back-propagation algorithm and its variants to solve a nonlinear classification problem.

This is a team project. One team will submit one report.
Due date: 2017_9_17 (Sunday, end of day 11:59pm)

Report: No more than 4 pages in IEEE journal publication format. Include Matlab code in an appendix, separated from the report body (not counted toward the 4 page limit).



Create 2,000 training data points as illustrated in the figure, i.e., 1,000 training data points in Region A and Region B, respectively. Pay attention to the definitions of the parameters as shown in the figure. Let d denote the vertical distance between the two rings. In this project, r=10, w=6. For your reference, feel free to use neural network toolbox demos, especially back-propagation demos.

Create 1,000 testing data points (500 for each class) to generate your final test results for the report.

Implement the basic back-propgation, and back-propagation with momentum and the Levenberg–Marquardt to solve this pattern classification problem.

Design and train one multilayer perceptron network for each pattern classification problem when d=2, -4, -8, respectively (i.e., you will solve 3 pattern classification problems in this project. The difficulty increases as d decreases). Randomly initialize your weight vector (set a random seed so your TA can duplicate the result). Throughout the experiments, you decide what learning rate parameters to use. But make sure your results are comparable, so fix some parameters when you try to examine the effect of others. For example, you should keep the learning rate parameters the same when comparing the effect of the number of hidden neurons. While you perform your tasks, keep in mind of the practical issues concerning multilayer perceptron training such as data preprocessing, weight initialization, stopping criteria, learning rate, etc. etc.

Summary and outline of your tasks -

1) Pre-define a 1-hidden layer network with a specified number (your choice) of hidden neurons. Use BP, BP with momentum, and Levenberg-Marquardt to learn to solve the double moon classification problem. Change your learning rate parameters to see their effect on the learning curves. Do that for cases of d=2, d=-4, and d=-8, respectively. Plot your learning curves and the nonlinear separating planes. Summarize the results in a confusion matrix.
2) Have d= - 8. Change the number of the hidden layer neurons (but use 1 hidden layer). Plot the learning curves and also the nonlinear separating planes. Summarize the results in a confusion matrix.