Homework 6

November 2, 2017

Due: November 9, 2017, 11:59 PM EST

Instructions

Your homework submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. If you do work with others, you must list the people you worked with. Submit your solutions as a PDF to the E-Learning at UF (http://elearning.ufl.edu/).

Your programs must be written in either MATLAB or Python. The relevant code to the problem should be in the PDF you turn in. If a problem involves programming, then the code should be shown as part of the solution to that problem. If you solve any problems by hand just digitize that page and submit it (make sure the problem is labeled).

If you have any questions address them to:

- Catia Silva (TA) – catiaspsilva@ufl.edu
- Sheng Zou (TA) – shengzou@ufl.edu

Problems - 10 points

In this homework, you will be implementing the **backpropagation algorithm**. Code the backpropagation algorithm for neural networks on your own and include your code with the homework solution. Test your implementation on the following provided two-class dataset ("HW6_Data.txt", Figure 1).

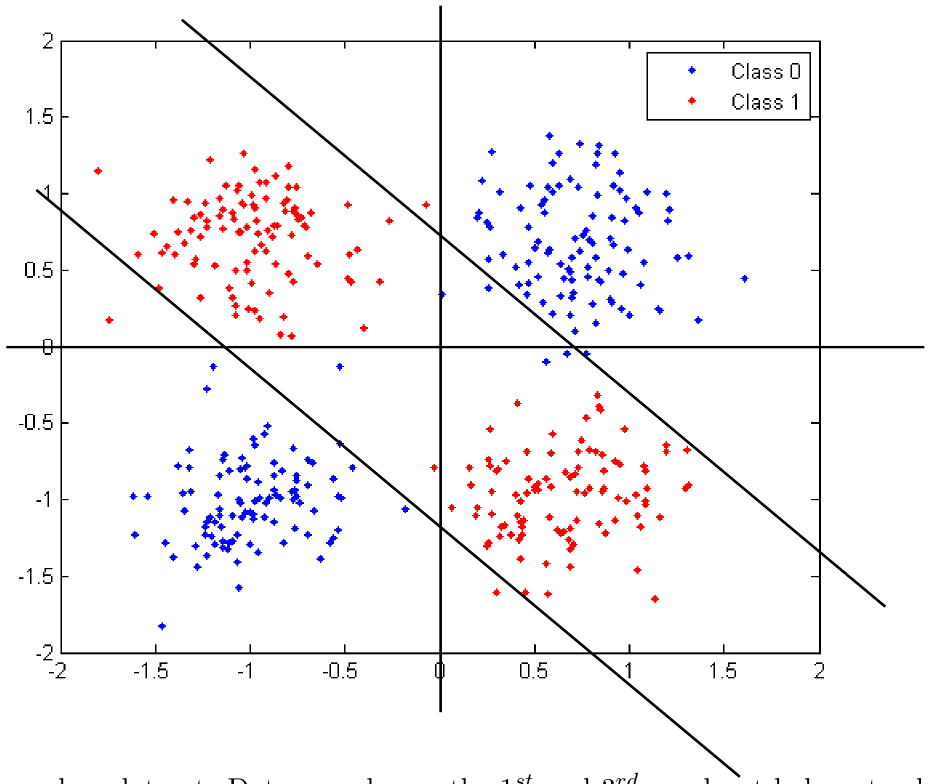

Figure 1: Two-class dataset. Data samples on the 1st and 3rd quadrant belong to class 0, and data amples on the 2nd and 4rd belong to class 1.

As you have seen in class, the solution to this XOR problem is two parallel decision boundaries (lines in the $\mathbb{R}^2$ space), with either positive slopes or negative slopes.

- Define your network's architecture (processing units in the input, hidden and output layers) and use a sigmoid activation function in the hidden layer. Explain why you used that number of hidden processing elements.
- Plot the corresponding decision boundaries after the networks have finished training. As well, see how well your solution performs on samples that do not belong to the training set by generating additional samples that follow the same pattern. Keep these generated points within a square that has length 2 on each of its sides.

- Can multiple architectures solve this problem? If no, why not? If yes, give at least two examples. Which one would you choose and why? Justify your answer.
- Choose 8 points for each class, 4 points in each quadrant, as your training data. Are the decision boundaries *optimal*? If not, what is the best solution? Comment on the generalization ability of your network with such fewer points.
- Lastly, explain what solution you would choose if you were going to hand-select the parameters of the network. Does the multi-layer neural network learn the solution that you expected?
- How would you improve the generalization accuracy of the multi-layer neural network?