Home Work 2

Due Date: March 17th

Exercise

Code in any programming language a 2 layer fully connected neural network for binary classification with a variable number of hidden units. Use a steepest descent with a fixed learning rate and cross-entropy for the cost function. For the output layer, use a sigmoid nonlinearity (i.e. $\frac{1}{1+\exp(-a)}$) and, for the hidden layer, a ReLu nonlinearity (i.e. $\max(0, a)$).

Dataset 1: The input is a three dimensional vector where each entry is uniformly distributed between 0 and 2 ($x_i \sim \mathcal{U}(0, 2)$ for $i = 1$, 2 and 3). The output is obtained by doing the sum of the greatest integer of each input. If the sum is even then y is zero, if the sum is odd then y is 1. (i.e. $y = (\sum_{i=1}^{3} \lfloor x_i \rfloor)\%2$).

Dataset 2: Like Dataset 1 but x_i can take any value between 0 and 3, uniformly

Deliverables

- The gradient with respect to the weights in both layers.
- The code and instructions in how to run it in a mac. The inputs to the code should be an $N \times D$ inputs matrix $\mathbf{X}$, an $N \times 1$ outputs vector $\mathbf{y}$, the number of hidden neurons R, a nonnegative scalar ν for the regularizer weight, and a positive scalar for the step-size η. Also it should have a test set $\mathbf{X}_s$ and $\mathbf{y}_s$ to decide when to stop training.
- Graphs with the cross-entropy error for the training and test sets and the equivalent graphs with the training and test errors (number of misclassified points). Plot the graphs for different values of η and number of hidden units, make sure that the graphs convey different information that you can use to argue in the last item. Use at least 200 data points for the first dataset and 600 for the second for the training and, at least as much, for the test set.
- A 200-word paragraph (or less) interpreting the weight vector. Detail what surprised you or what did you find normal in the solution.

Note: You need to code the NN, it is part of the assignment. This is a machine-learning course, not a data-mining course. But, if I were you, I would definitely try any package out there that solves NNs.

Note 1: The output layer should be similar to the logistic regression that you already coded for Homework 1.

Note 2: If you limit the input data two two dimensions you will be using the same data that I used in class and you can visualized the classifier in 2D.