Assignment 3

Due Date: 11/7/2017

Total Points: 150

In this exercise, you will implement a neural network and evaluate its performance. You have the option to either use your earlier implementation of the Gradient Descent Algorithm or use a **built-in implementation** of the Gradient Descent technique in the programming language of your choice. However, you have to implement the forward and backpropagation ***from scratch*** using the programming language of your choice (**without using a toolbox from R, Matlab, Python or any other programming language, please make sure you attach your code in the folder!**). For implementing some of the principles of programming, try to modularize the code as much as possible and consider **testing** your algorithm on a smaller known dataset before starting the assignment (such as XNOR data since we know from class that it is **not** linearly separable). In this assignment, we will use the flu dataset to detect the occurrence of flu-like symptoms in the past year. We will also use the wine dataset located here: https://archive.ics.uci.edu/ml/datasets/Wine to test the algorithm before testing it on the noisy dataset. Assignment 3 contains 3 sections. Please use **Feature Scaling** for this assignment. Please address the subparts in **each** section to receive full credit. Also, analysis is a crucial aspect of the assignment, so for each subpart try to answer the question in more detail.

1. Neural Network with Wine Dataset
    a. Can you implement a backpropagation Neural Network to classify the three class, 13 attribute wine dataset using a single hidden layer? Please divide the data into training, validation, and test data and use the test case to evaluate performance.
    b. Test different number of hidden nodes (at least 5 options) and compare the performance; does the highest number of hidden nodes improve performance?
    c. **Evaluate performance using a metric (such as confusion matrix, F measure, AUC). You may also use graphs for explaining your observations.**

2. Neural Network with WPBC Dataset
    a. Can you map the input variables (**at least** 5 features) to Flu (y) using a multilayer perceptron (aka neural network)? Please divide the data into training, validation, and test data and use the test case to evaluate performance.
    b. Start with a single hidden layer with different number of hidden nodes (similar to 1 b)
    c. Does adding more hidden layers improve performance? Again, compare at least **three** different configurations.
    d. **Evaluate performance using a metric (such as confusion matrix, F measure, AUC).**

3. Regularization:
    a. Does regularization improve the performance for Q 2 above?
    b. Can you test the performance of your neural network using random initialization of weights vs. all weights equal to the same value? Is symmetry breakage a critical issue here?
    c. **Evaluate performance for each case using a metric (such as confusion matrix, F measure, AUC).**
    d. Are there any observations on the performance comparison for the two datasets? What are the possible reasons for the difference (if you think that is the case) in the neural network performance? Please have specific comments to justify your answer.

Please make sure to submit the code and report as a zipped file in Dropbox on Pilot (titled "Assignment 3"). Please make sure that the report is in a pdf format.

Academic Integrity

Discussion of course contents with other students is an important part of the academic process and is encouraged. However, it is expected that course programming assignments, homework assignments, and other course assignments will be completed on an individual basis (unless specified otherwise). Students may discuss general concepts with one another, but may not, under any circumstances, work together on the actual implementation of any course assignment. If you work with other students on "general concepts" be certain to acknowledge the collaboration and its extent in the assignment. Unacknowledged collaboration will be considered dishonest. "Code sharing" (including code from previous quarters) is strictly disallowed. "Copying" or significant collaboration on any graded assignments will be considered a violation of the university guidelines for academic honesty.
If the same work is turned in by two or more students, all parties involved will be held equally accountable for violation of academic integrity. You are responsible for ensuring that other students do not have access to your work: do not give another student access to your account, do not leave printouts in the recycling bin, pick up your printouts promptly, do not leave your workstation unattended, etc. If you suspect that your work has been compromised notify me immediately. If you have any questions about collaboration or any other issues related to academic integrity, please see me immediately for clarification. In addition to the policy stated in this syllabus, students are expected to comply with the Wright State University Code of Student Conduct (http://www.wright.edu/students/judicial/conduct.html) and in particular the portions pertaining to Academic Integrity (http://www.wright.edu/students/judicial/integrity.html) at all times.