Assignment 3

Due Date: 11/07/2019

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. Please note that you can also use datasets such as here: http://data.princeton.edu/wws509/datasets to test your algorithm before testing it on the BSOM dataset. Assignment 2 contains three sections. Please address the subparts in **each** section to receive full credit. Analysis is a crucial aspect of the assignment, so for each subpart try to answer the question in more detail. Please use **Feature Scaling** for this assignment. Also, please divide the data into **training and test data** and use the **test case to evaluate performance**.

1. Neural Network with BSOM Dataset (100 points):
    a. Can you implement a backpropagation Neural Network using variables 'all_mcqs_avg_n20', 'all_NBME_avg_n4', 'CBSE_01', and 'CBSE_02' to classify 'LEVEL' with a single hidden layer and 5 hidden nodes?
    b. Test different number of hidden nodes (**at least** 4 more options) with a single hidden layer and compare the performance. Does the highest number of hidden nodes have the highest performance?
    c. Does adding more hidden layers improve performance? Compare **at least** 3 different configurations.
    d. Evaluate performance for each case using metrics (such as confusion matrix, precision, recall, F1 scores and AUC). Which metrics are better in comparing model performances for this classification problem and **why**?

2. Regularization (30 points):
    a. Does regularization improve the performance for the best model in Question 1? Test at least 5 different regularization values to support your answer.

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 metrics (such as confusion matrix, precision, recall, F1 scores and AUC).

3. Neural Network with More Variables (20 points):
    a. Add one more variable that you think might improve the performance for the best model in Question 2.
    b. Add another variable that you think might improve the performance into the model in Question 3a.
    c. Compare performances of three models using metrics (such as confusion matrix, precision, recall, F1 scores and AUC).

Please make sure to submit a zipped file in Dropbox on Pilot titled YourName_Assignment3 with the report in 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.