In this homework, you will implement an artificial neural network (ANN) classifier. You will learn the weights of this classifier implementing the backpropagation algorithm. In your implementation, you SHOULD NOT use pretraining, which means you should select the initial weights randomly. However, your implementation SHOULD allow using dropout regularization.

You will test your ANN classifier on the following datasets. In particular, you will use the first dataset for Part 1 and the second dataset for Part 2.

1. The Thyroid dataset, which is taken from the UCI repository and available on the course web page. Its details are given as follows:
   - It contains separate training ("ann-train.data") and test ("ann-test.data") sets.
   - The training set contains 3772 instances and the test set contains 3428 instances.
   - There are a total of 3 classes. Note that this dataset has unbalanced class distributions.
   - In the data files, each line corresponds to an instance that has 21 features (15 binary and 6 continuous features) and 1 class label.

2. The MNIST dataset of handwritten digits, whose details are given in http://yann.lecun.com/exdb/mnist/.

**Part 1:**

Implement a neural network classifier with a single hidden layer. Set your dropout factor to 0 (in other words, do not use dropout regularization). Test your implementation on the Thyroid dataset.

(a) List all the parameters (except the hidden unit number) that your implementation uses. Explain how you select the parameter values. Here you are not required to conduct cross-validation to select the values (of course, if you’d like, you can). Just very briefly explain why you select these values.

(b) Explain how you manage the situation of having unbalanced class distributions. Also explain if you use any normalization on the features.

(c) For different values of the hidden unit number (at least five different values), obtain training and test set accuracies. Report the overall accuracy as well as the class-based accuracies. You may report these accuracies in a "nice" table.

(d) Fix the hidden unit number to a value that gives good class-based test set accuracies. Now conduct the following experiments with your selected hidden unit number.
   - Obtain the results with and without normalization. Compare the overall test set accuracy and the class-based test set accuracies. Discuss how the normalization affects the results.
   - Obtain the results when you use the data as it is and when you handle the situation of having unbalanced class distributions (for example, by duplicating the samples from the minority classes). Compare the overall test set accuracy and the class-based test set accuracies. Discuss how the unbalanced class distributions affect the results.
   - Use the stochastic, batch, and mini-batch approaches for the weight updates. Compare their results and discuss the difference in between the use of these approaches, if any.
Part 2:
Implement a neural network with **TWO hidden layers**. Test your implementation on the MNIST dataset.

(a) First set your dropout factor to 0 (do not use dropout regularization). Use at least three different architectures. List the number of hidden units in each architecture. Report training and test set accuracies for each selected architecture. (You do not need to specify the values of the other model parameters or you do not need to explain how or why you select these values. However, to get acceptable results, you may need to try different values of the other parameters.)

(b) Then set your dropout factor to some positive value (use dropout regularization). Of course, you should select a dropout factor that yields good accuracies. For each of the three architectures that you select for (a), report training and test set accuracies. Compare these accuracies with the ones that you obtain in (a). Discuss how dropout regularization affects the results.

This homework asks you to implement a neural network classifier by **writing your own codes**. Thus, you are not allowed using any machine learning package. In your implementation, you may use any programming language you would like.

You will have two weeks to complete your implementation and your runs. However, the runs of this homework may take a considerable amount of time, especially if you use Matlab in your implementation. Therefore, do not leave this homework to the last minute; make sure to give yourselves enough time to finish your runs before the deadline.

You are expected to write your report neatly and properly. The format, structure, and writing style of your report as well as the quality of the tables and figures will be a part of your grade. Use reasonable font sizes, spacing, margin sizes, etc. You may submit either a one-column or a double-column document. In your report, do not give any screen shots or outputs of your program but summarize what you have found at the end of your runs. Do not forget to address the questions specifically asked to you. **Your report should be a maximum of 4 pages.**

Please submit the hardcopy of your report before the deadline. DO NOT submit the printout of your source code. However, you need to email the source code of your implementation before the deadline. The subject line of your email should be **CS 550: HW1.**