CS436/536: Introduction to Machine Learning
Homework 4
Due 11/13 **before the class**

Instructions: Please start this homework early.

To solve these problems, you are allowed to consult your classmates, as well as the class textbook (*Learning from Data* by Abu-Mostafa, Magdon-Ismail, and Lin, which we will call LFD) and the slides posted on Brightspace, but no other sources. If you have bought the book (in print or electronic form), you can access the online 'e-chapters' at http://www.amlbook.com/eChapters.html. You are encouraged to collaborate with other students, while respecting the collaboration policy (please see the module on Academic Honesty on Brightspace). Please write the names of all the other students you collaborated with on the homework. Everyone must write up their assignments separately.

Please write clearly and concisely, and use rigorous, formal arguments. Homework is due at the beginning of lecture, and homework turned in later will be considered late and will use up one of your late days. You must use Brightspace to submit the homework as a single neatly typed pdf file. Hand-drawn formulas or figures are okay and may be included as images within the pdf. If a programming assignment calls for plotting the results, axes must be clearly labeled, and its meaning must be obvious to anyone with only a rudimentary knowledge of machine learning and computer science. Emailed copies will not be accepted.

There is no supporting code for this homework. Please feel free to explore the documentation and examples for the `scikit-learn` python library. One of the objectives of this homework is to become familiar with reading through the documentation of existing python libraries and making the best use of the documentation, the small examples and recipes they provide to accomplish common tasks in machine learning. All the best!

In this homework, you will use the data you generated in previous homeworks from the MNIST Digits Dataset for classifying 1s vs. Not 1s, where you created $\mathcal{D}$ with 300 randomly selected data points and $\mathcal{D}_{\text{test}}$ consisting of the remaining data points.

(1) [400 points] Neural Networks and Backpropagation.

Implement neural networks and *stochastic* gradient descent using backpropagation for a neural network architecture specified by a vector $[d^{(0)}, d^{(1)}, \ldots, d^{(L)}]$, where $d^{(L)} = 1$. Use $\theta(s) = \tanh(s)$ as the transformation function for every node in a hidden layer. For the output layer, allow the user to specify whether to use $\theta(s) = s$, $\theta(s) = \tanh(s)$ or $\theta(s) = \text{sign}(s)$. Set the architecture to $[2, m, 1]$, i.e. 2 inputs or $d^{(0)} = 2$, m hidden units, i.e, $d^{(1)} = m$, and 1 output node, i.e. $d^{(L)} = 1$ and $L = 2$. Implement *stochastic* gradient descent on the squared error $E_{\text{in}}(\mathbf{w}) = \frac{1}{4N}\sum_{n=1}^{N}(h(x_n; w) - y)^2$, and check your gradient computation as follows:

(a) Use a network with $m = 2$. Set all weights to 0.25 and consider a dataset with 1 data point: $x_1 = \begin{bmatrix}1\\2\end{bmatrix}; y_1 = 1$. For the output node using either $\theta(s) = s$ or $\theta(s) = \tanh(s)$, obtain the gradient of $E_{\text{in}}(\mathbf{w})$ using the backpropagation algorithm. Report this result and check its dimensionality, there must be as many numbers as parameters in this network.

(b) Now, obtain the gradient numerically by perturbing each weight, one at a time, by 0.0001. Report this result. If the result is not similar to the prevoius result, there is likely something wrong with your backpropagation gradient computation.

(2) [600 points] Neural Networks for MNIST Digits Dataset. Use your netural networks implementation with $m = 10$ hidden units to build a classifier for classifying 1s vs. Not 1s. Use the two features and data you generated in previous homeworks and the 300 input data points you selected as the training set.

Randomly initialize the weights to small values. Then, use $\theta(s) = \tanh(s)$ as the transformation function for the hidden units. During *training*, use the $\theta(s) = s$ linear transformation function for the *output layer* only, i.e., use the regression for classification paradigm. Once the *training phase is complete*, switch the transformation function used in the *output layer* to the sign function, i.e., set $\theta(s) = \text{sign}(s)$ to classify data points.

(a) Plot $E_{\text{in}}(w)$ vs. iterations for the variable learning rate *stochastic* gradient descent heuristic and 2×10^6 iterations. Plot the decision boundary for the resulting classifier.

(b) Now, use weight decay with $\lambda = 0.01/N$ and use variable learning rate *stochastic* gradient descent to minimize the augmented error (E_{aug}). Plot the decision boundary for the resulting classifier.

(c) Now, use early stopping with a validation set of size 50 and a training set of size 250. Plot the decision boundary for the resulting classifier that had minimum validation error (E_{val}).