Assignment 3

Due on November 30, 2016 (23:59:59)

Instructions. There are two parts in this assignment. The first part involves a series of theory questions and the second part involves coding. The goal of this problem set is to make you understand and familiarize with Neural Network algorithm.

PART I: Theory Questions

1. What are differences between logistic regression and linear regression?
2. What are differences between logistic regression and naive bayes methods?
3. Assume you have a network as given below and $tanh(x)$ function is used as activation function. Show forward and backward propagations with notations.

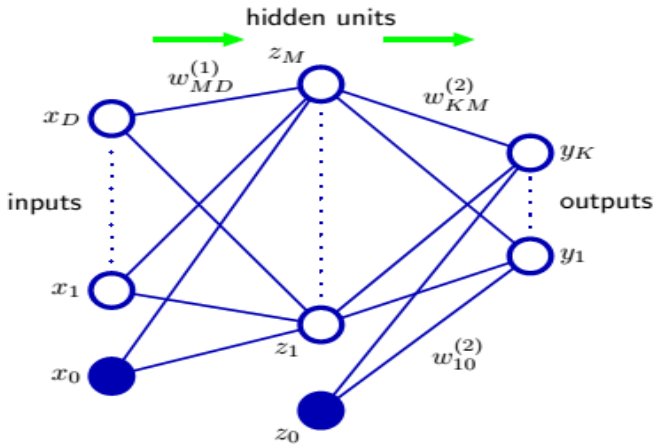

Figure 1

4. Suppose you are using a 2 layer neural network that has 2 input units (+1 bias), a hidden layer with 2 input units (+1 bias) and one output unit. Suppose you have learned

$$\theta^{(1)} = \begin{bmatrix} 1 & -1.5 & 3.7 \\ 1 & 5.1 & 2.3 \end{bmatrix}$$

which is used to compute second layer's activations, and $\theta^{(2)} = [1 \quad 0.6 \quad -0.8]$ which is used to compute output layer's activation ($h_\theta(x)$). Suppose we swap the weights as

$$\theta^{(1)} = \begin{bmatrix} 1 & 5.1 & 2.3 \\ 1 & -1.5 & 3.7 \end{bmatrix}$$

and $\theta^{(2)} = [1 \quad -0.8 \quad 0.6]$. How will this change the $h_\theta(x)$? Explain.

PART II: Classification

As you learned in the class neural network models take their inspiration from the brain. They are nonparametric estimators that can be used for classification and regression. In this part of the assignment, you will try to predict the class of images that are given. You will implement two different neural network architecture for classification and verify their performances on the given dataset's test set.

You will use MNIST dataset which is explained in Dataset section 3, for both classification methods.

1 Single Layer Neural Network

As you learned in class, you'll implement a single layer neural network for classification problem. In other words you'll have only two layers: Input and output layers. You can use different activation functions for the output layer. You can interpret this part of the assignment as a logistic or softmax regression problem.

2 Multi Layer Neural Network

For this part of the assignment, you have to implement multi layer neural network for classification. In other words your network consists of one input layer, n hidden layer(s) and one output layer. You will implement forward and backward propagations.

Training a network

- You should determine the number of units in your hidden layer.
- You should determine batch size as you learned in the class.
- You should determine a learning rate for your gradient descent method.
- Remember, learning rate parameter may be a problem (too big - may not converge, too small - very slow convergence). For this reason you can define a learning rate decay parameter. You will start with a learning rate value and after each epoch you will reduce the learning rate by multiplying it by a decay rate. This operation can deal with mentioned problem.
- You can use different activations functions: Sigmoid, tanh, ReLU etc.
- You can use different objective functions: Cross-entropy error, sum of squared error (SSE) etc.

- You can control your implementation by plotting loss. You can see if it converges or if it needs a different parameter setting.
- You should discuss about your each experiment in the report. Comment about their effects.

In brief:

- You will implement single layer neural network and run experiments on MNIST dataset. You'll change parameters (activation func., objective func. etc.) and report results with a table format in your reports.
- You will implement a neural network which contains one hidden layer. You'll change the mentioned parameters (unit number in the hidden layer, activations function etc.) and report the results.
- Then you'll change your architecture and use a network that contains two hidden layers. Repeat same experiments and comment about the results.
- You have to comment about results and parameters' effects.
- If your implementation is flexiable, you'll get bonus points (in other words, try not to write separate code for each architecture. If you use n hidden layers, your method should create a n-layer network and learn the classifier.)

3 Dataset

- MNIST dataset `http://yann.lecun.com/exdb/mnist/` contains handwritten digits and corresponding labels. Dataset contains 60000 training and 10000 test images.
- Dataset and the code to load it are provided to you. You can donwload them from `ftp://cs.hacettepe.edu.tr/pub/dersler/BBM4XX/BBM409_ML/Assignment_3/load/`. The dataset is split into three sets: Training (50000 images), test (10000 images)and validation (10000 images). You can use validation set for training (50000+10000 images).
- When you load the dataset, you'll see that every image is represented with [784x1] vector and has a label. You'll use the given representation (do not have to extract new features).

Grading

- Code (58) : 18 points for single-layer neural network, 40 points for multi layer neural network.
- Report(42): Theory part: 12 points, Analysis of the results for classification: 30 points. You have to write your report with LaTeX.

  Notes for the report: You should analyse the method you employed. How did you improve your results? Explain every step you choose. Comment about the results.

Compare single layer and multi layer neural network. Comment about the activation functions, loss functions etc. that you used for your experiments. Your reports have to include your classification accuracy.

Late Policy

You may use up to five extension days (in total) over the course of the semester for the three problem sets you will take. Any additional unapproved late submission will be weighted by 0.5.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specic problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else