CARNEGIE MELLON UNIVERSITY

10601a: Homework #8 - "Neural Networks" (autolab)

TA-in-charge:
Kartik Goyal (kartikgo@cs.cmu.edu)

Assigned: 02/28/2014
Due: Friday, 03/07/2014, 11:59 p.m.
Late Penalty: 25% per day.

Policy on Collaboration among Students

The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone, and the student should be ready to reproduce their solution upon request. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved, on the first page of their assignment. Specifically, **each assignment must contain a file named collaboration.txt where you will answer the following questions**:

- Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered 'yes', give full details? (e.g."Jane explained to me what is asked in Question 3.4").
- Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered 'yes', give full details? (e.g. "I pointed Joe to section 2.3 to help him with Question 2").

Collaboration without full disclosure will be handled severely, in compliance with CMU?s Policy on Cheating and Plagiarism. All violations (even first one) will carry severe penalties, up to failure in the course, and will in addition always be reported to the university authorities.
Some of the homework assignments used in this class may have been used in prior versions of this class, or in classes at other institutions. Avoiding the use of heavily tested assignments will detract from the main purpose of these assignments, which is to reinforce the material and stimulate thinking. Because some of these assignments may have been used before, solutions to them may be (or may have been) available online, or from other people. It is explicitly forbidden to use any such sources, or to consult people who have solved these problems before. You must solve the homework assignments completely on your own. I will strictly enforce this policy, and if a violation is detected

it will be dealt with harshly. Collaboration with other students who are currently taking the class is allowed, but only under the conditions stated above.

General Instructions

The goal of this assignment is to familiarize you with several implementation decisions involved in training neural networks, by ***implementing a neural network with one hidden layer***, entirely from scratch. You will be using datasets from the two domains similar to the ones used in the 'Decision Tree' assignment(assignment 6). A major change in this assignment is that some input attributes are multi-valued or continuous-valued instead of the binary-valued attributes of Assignment 6. Notice that Neural networks are well suited for continuous valued inputs. They are also different from Linear regression models as they accommodate non-linearity, which allows us to consider a larger space of functions than simple linear functions. However, training (or "fitting") the model by minimizing the training set error poses several challenges, as there might be multiple local minima in the parameter (weight) space.

Your goal is to achieve the lowest error rates on the test sets for both domains. You can (and should) experiment with different configurations for the two domains.

The first task is to predict whether a song was a 'hit', meaning it made it onto the Billboard Top 50- each instance has a label hit equal to 'yes(1)' or 'no(0)'. The attributes are: year of release(multi-valued discrete), length of recording (continuous), jazz(yes/no), rock and roll(yes/no).

The second task is to predict the final score for high school students. The attributes are student grades on 2 multiple choice assignments M1 and M2, 2 programming assignments P1 and P2, and the final exam F. The scores of all the components are in the range [0,100]. All the attributes are multi-valued discrete. Again, check the csv files to see the attribute values. The final output also has a range [0,100]. Notice that we are performing regression for this problem instead of classification which was done in the decision tree assignment.

Before you start coding your neural network , think about various decisions you need to make about your implementation. A few of them are:

- Initialization of weights
- Number of units in the hidden layer
- Network connectivity
- Number of iterations (or, when to stop the training)

Generally, the number of hidden layers too is an important decision to be made, but in this assignment we require you to implement a Neural Network with one hidden layer. There may be many more decisions to be made. All such decisions affect the speed and performance of your neural network, some more than others.
You must use the sigmoid function as discussed in the class for introducing non-linearity, even though other non-linear functions exist

Remember that autolab won't allow your code to run for a long period of time(upper limit of 3 minutes). Hence, decide upon your stopping criteria(an upper limit on the number of iterations might be best but try to converge).

You may use Matlab(Octave), Python or Java to complete this assignment. (Throughout, examples are shown assuming you're using Octave, but Python and Java are fine.) We recommend you to use Matlab/Octave because using objects like vectors and matrices will result in an efficient code for back-propagation. Also, it will be easier to code Backpropagation with Matlab and the teaching staff prefers to write backpropagation code in Matlab. If you want to use python, you can look into 'numpy' and 'scipy' libraries for using vectors and matrices. We've provided you with attributes and labels split into training and development data in files 'music*.csv' and 'education*.csv'.The format is comma separated, one row per observation, one column per attribute. Your program will take two files as input: the first file, containing labelled data, to be used to train the network (fit the weights). The second file, containing unlabeled data, to be used to make predictions.
IMPORTANT: Do not use any standard Neural Network packages, especially while using MATLAB.

0 Neural Network Development

We expect two files ***'NN_music.m'***,***'NN_education.m'***. You have training and development sets for both the domains. You will estimate the weights on your training set, print the squared error after every 10 iterations while training and then use the learned weights to print the output decisions on the development set. The syntax is:

```
$ octave -q  NN_music.m <training_file> <test_file>
1022.6
1012.7
1005.8
...(do not print these dots; they just signify continuation)
TRAINING COMPLETED! NOW PREDICTING.
no
yes
yes
no
yes
...
$ octave -q  NN_education.m <training_file> <test_file>
632.6
600.7
578.8
...
TRAINING COMPLETED! NOW PREDICTING.
23.0
55.0
65.0
60.0
80.0
50.0
```

. . .

The output values shown in the examples do not represent actual output.
In the above syntax '<training_file>' is 'music_train.csv' and 'education_train.csv', and '<test_file>' is 'music_dev.csv' or 'education_dev.csv' for the music and education domains respectively. You can compare your decisions on development file with the true labels in the development files and calculate the error. The autograder will run your program on the training files and the test datasets, and then evaluate your performance(error) on the test datasets. This error will be shown on the leaderboard!
Is your performance on the test set similar to your performance on the development set? Does improving your performance on development set *always* result in a better performance on your test set?
A correct implementation for both the datasets will be sufficient for full credit. A good performance in the competition will earn you extra credit.
Please keep in mind to use ';'s in your Matlab/Octave code as you don't want to print any spurious data.
Beware! You are allowed to run your code on the autograder only 10 times. Hence, work with the development set and test your performance on autolab only when you are very confident!

1 AUTOLAB SUBMISSION

Submit a .tgz containing your source code(2 files) and a file collaboration.txt). You can create that file by running 'tar -cvf hw8.tgz *.m collaboration.txt'. DO NOT put the above files in a folder and then tar gzip the folder. You must submit this file to the 'homework8' link on Autolab.

Good Luck! May the global minimum be with you.