Homework #4

RELEASE DATE: 05/23/2017

DUE DATE: 06/20/2017, BEFORE 14:00

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE FACEBOOK FORUM.

Unless granted by the instructor in advance, you must turn in a printed/written copy of your solutions (without the source code) for all problems.

For problems marked with (*), please follow the guidelines on the course website and upload your source code to designated places. You are encouraged to (but not required to) include a README to help the TAs check your source code. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

> This homework set comes with 160 points and 40 bonus points. In general, every homework set would come with a full credit of 160 points, with some possible bonus points.

Random Forest

1. If bootstrapping is used to sample $N' = pN$ examples out of N examples and N is very large, argue that approximately $e^{-p} \cdot N$ of the examples will not be sampled at all.

2. Consider a Random Forest G that consists of three binary classification trees $\{g_k\}_{k=1}^{3}$, where each tree is of test 0/1 error $E_{\text{out}}(g_1) = 0.15$, $E_{\text{out}}(g_2) = 0.25$, $E_{\text{out}}(g_3) = 0.35$. What is the possible range of $E_{\text{out}}(G)$? Justify your answer.

3. Consider a Random Forest G that consists of K binary classification trees $\{g_k\}_{k=1}^{K}$, where K is an odd integer. Each g_k is of test 0/1 error $E_{\text{out}}(g_k) = e_k$. Prove or disprove that $\frac{2}{K+1}\sum_{k=1}^{K} e_k$ upper bounds $E_{\text{out}}(G)$.

Gradient Boosting

4. For the gradient boosted decision tree, if a tree with only one constant node is returned as g_1, and if $g_1(\mathbf{x}) = 2$, then after the first iteration, all s_n is updated from 0 to a new constant $\alpha_1 g_1(\mathbf{x}_n)$. What is s_n? Prove your answer.

5. For the gradient boosted decision tree, after updating all s_n in iteration t using the steepest η as α_t, what is the value of $\sum_{n=1}^{N} s_n g_t(\mathbf{x}_n)$? Prove your answer.

6. If gradient boosting is coupled with linear regression (without regularization) instead of decision trees. Prove or disprove that the optimal $\alpha_1 = 1$. (*A* 10% *bonus can be given if your proof for either case is rigorous and works for general polynomial regression.*)

7. If gradient boosting is coupled with linear regression (without regularization) instead of decision trees. Prove or disprove that the optimal $g_2(\mathbf{x}) = 0$. (*A* 10% *bonus can be given if your proof for either case is rigorous and works for general polynomial regression.*)

Neural Network

8. Consider Neural Network with $\text{sign}(s)$ instead of $\tanh(s)$ as the transformation functions. That is, consider Multi-Layer Perceptrons. In addition, we will take $+1$ to mean logic TRUE, and -1 to mean logic FALSE. Assume that all x_i below are either $+1$ or -1. Write down the weights w_i for the following perceptron

$$g_A(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i x_i\right).$$

to implement

$$\texttt{OR}\left(x_1, x_2, \ldots, x_d\right).$$

Explain your answer.

9. Continuing from Question 8, among the following choices of D, write down the smallest D for some 5-D-1 Neural Network to implement $\texttt{XOR}\big((x)_1, (x)_2, (x)_3, (x)_4, (x_5)\big)$. Explain your implementation. (It is not so easy to prove the smallest choice, so let's leave the proof for the bonus.)

10. For a Neural Network with at least one hidden layer and $\tanh(s)$ as the transformation functions on all neurons (including the output neuron), when all the initial weights $w_{ij}^{(\ell)}$ are set to 0, what gradient components are also 0? Justify your answer.

11. For a Neural Network with one hidden layer and $\tanh(s)$ as the transformation functions on all neurons (including the output neuron), prove that for the backprop algorithm (with gradient descent), when all the initial weights $w_{ij}^{(\ell)}$ are set to 1, then $w_{ij}^{(1)} = w_{i(j+1)}^{(1)}$ for all i and $1 \le j < d^{(1)}$.

Experiments with Random Forest

Implement the Bagging algorithm with $N' = N$ and couple it with your decision tree in HW3 to make a preliminary random forest G_{RF}. Produce $T = 30000$ trees with bagging. Compute E_{in} and E_{out} using the 0/1 error.

Run the algorithm on the following set for training (i.e. re-use HW3 datasets):

`hw3_train.dat`

and the following set for testing:

`hw3_test.dat`

12. (*) Plot a histogram of $E_{\text{in}}(g_t)$ over the 30000 trees.

13. (*) Let G_t = "the random forest with the first t trees". Plot a curve of t versus $E_{\text{in}}(G_t)$.

14. (*) Continuing from Question 13, and plot a curve of t versus $E_{\text{out}}(G_t)$. Briefly compare with the curve in Question 13 and state your findings.

Now, 'prune' your decision tree algorithm by restricting it to have one branch only. That is, the tree is simply a decision stump determined by Gini index. Make a random 'forest' G_{RS} with those decision stumps with Bagging like Questions 12-14 with $T = 30000$. Compute E_{in} and E_{out} using the 0/1 error.

15. (*) Again, let G_t = "the random forest with the first t decision stumps". Plot a curve of t versus $E_{\text{in}}(G_t)$.

16. (*) Continuing from Question 15, and plot a curve of t versus $E_{\text{out}}(G_t)$. Briefly compare with the curve in Question 15 and state your findings.

Bonus: Crazy XOR

17. (10%) Continuing from Question 8, prove or disprove that $D = d$ is the smallest D that allows for implementing $\texttt{XOR}\big((x)_1, (x)_2, \ldots, (x_d)\big)$ with a d-D-1 feed-forward neural network with $\text{sign}(s)$ as the transformation function (such a neural network is also called a Linear Threshold Circuit).

18. (10%) Continuing from Question 8, if you are allowed to use D neurons (including the one for output) to implement $\texttt{XOR}\big((x)_1, (x)_2, \ldots, (x_d)\big)$, but can connect the neurons in whatever way as long as it is feed-forward (such as connecting the input directly to neurons in other "layers"), what is the smallest D (that you can find) for implementing the function? Explain your implementation. You can refer to

http://www.nature.com/nature/journal/v475/n7356/fig_tab/nature10262_F2.html

for a possible construction using two neurons for $d = 3$.