Homework #3

RELEASE DATE: 05/29/2020

DUE DATE: 06/26/2020, BEFORE **13:00**

QUESTIONS ABOUT HOMEWORK MATERIALS ARE
WELCOMED ON THE NTU COOL FORUM.

Please upload your solutions (without the source code) to Gradescope as instructed. For problems marked with (*), please follow the guidelines on the course website and upload your source code to NTU COOL. 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.

Deep Learning Techniques

1. Recall that in neural networks, we calculate

$$s_j^{(\ell)} = \sum_{i=0}^{d^{(\ell-1)}} w_{ij}^{(\ell)} x_i^{(\ell-1)}$$

Let's simplify the summation a bit by assuming $w_{0j}^{(\ell)} = 0$. That is,

$$s_j^{(\ell)} = \sum_{i=1}^{d^{(\ell-1)}} w_{ij}^{(\ell)} x_i^{(\ell-1)}$$

Assume that $x_i^{(\ell-1)}$ are independent random variables with mean $\bar{x}$ and variance σ_x^2. Also, assume that $w_{ij}^{(\ell)}$ are zero-mean random variables with variance σ_w^2 that are independent to each other and to all $x_i^{(\ell-1)}$. Show that conditioned on $\mathbf{x}^{(\ell-1)}$, those $s_j^{(\ell)}$ are zero-mean random variables that are independent to each other.

2. Following Problem 1, express $\text{Var}(s_j^{(\ell)})$, the variance of each $s_j^{(\ell)}$, in terms of $d^{(\ell-1)}$, $\bar{x}$, σ_x^2 and σ_w^2.

3. If $s_j^{(\ell-1)}$ are independent, zero-mean, symmetric random variables, and ReLU activation is used on top of $s_i^{(\ell-1)}$ to get $x_i^{(\ell-1)}$. That is, $x_i^{(\ell-1)} = \max\left(s_i^{(\ell-1)}, 0\right)$. Show that $E\left[(x_i^{(\ell-1)})^2\right] = \frac{1}{2} E\left[(s_i^{(\ell-1)})^2\right]$.

4. Following Problem 1, if $s_i^{(\ell-1)}$ are independent, zero-mean, symmetric random variables and $w_{ij}^{(\ell)}$ are independent, zero-mean, symmetric random variables, and ReLU activation is used on top of $s_i^{(\ell-1)}$ to get $x_i^{(\ell-1)}$. Prove that

$$\text{Var}(s_j^{(\ell)}) = \frac{d^{(\ell-1)}}{2}\sigma_w^2 \text{Var}(s_i^{(\ell-1)}).$$

(*Note: It is not hard to prove that some conditional-independence results of $s_j^{(\ell)}$ like Problem 1 as well as properties like zero-mean and symmetry but we do not require you to prove so.*)

5. (Bonus 20%) Following Problem 4, derive an initialization scheme for the leaky ReLU activation $x = \max(s, a \cdot s)$ such that $\text{Var}(s_j^{(\ell)}) = \text{Var}(s_i^{(\ell-1)})$.

6. Recall that the exponential averaging formula in momentum is

$$\mathbf{v}_t = \beta \mathbf{v}_{t-1} + (1-\beta)\boldsymbol{\Delta}_t$$

where $\boldsymbol{\Delta}_t$ is the gradient vector, $\mathbf{v}_t$ is the moving direction, and $0 < \beta < 1$ is the averaging parameter. A common initialization is to take $\mathbf{v}_0 = \mathbf{0}$. At $t = T$, the formula results in

$$\mathbf{v}_T = \sum_{t=1}^{T} \alpha_t \boldsymbol{\Delta}_t$$

Express α_t in terms of β.

7. Following Problem 6, what is the smallest T such that $\alpha_1 \leq \frac{1}{2}$?

8. Following Problem 6, you may realize that $\sum_{t=1}^{T} \alpha_t \neq 1$. That is, $\mathbf{v}_T$ is actually a weighted sum of $\boldsymbol{\Delta}_t$, but **not a weighted average**. One way to correct so is to take

$$\mathbf{v}_T' = \frac{\mathbf{v}_T}{\sum_{t=1}^{T}\alpha_t} = \sum_{t=1}^{T} \underbrace{\left(\frac{\alpha_t}{\sum_{t=1}^{T}\alpha_t}\right)}_{\alpha_t'} \boldsymbol{\Delta}_t$$

as the moving direction instead. Express $\alpha_t' = \frac{\alpha_t}{\sum_{t=1}^{T}\alpha_t}$ with the simplest formula that you can think of.

9. Following Problem 8, what is the smallest T such that $\alpha_1' \leq \frac{1}{2}$?

10. Consider a **linear** neural network with no hidden layers and a squared error function. That is, the network just performs linear regression! Now, apply a 50% dropout on the inputs of the neural network by performing linear regression on about half the feature vectors only. That is, the dropout procedure tries to (stochastically) find $\mathbf{w}$ such that

$$\min_{\mathbf{w}} E_{\mathbf{p}}\left(\|\mathbf{y} - \text{X}(\mathbf{w} \odot \mathbf{p})\|^2\right)$$

where $\odot$ is a component-wise multiplication, $\mathbf{p}$ is a random binary vector whose components are sampled from a fair coin, and $E_{\mathbf{p}}$ is the expectation over $\mathbf{p}$. Derive a closed-form solution for the optimal $\mathbf{w}$. (*Hint: think about* $E_{\mathbf{p}}(\mathbf{p}^T\text{X}^T\text{X}\mathbf{p})$.)

Aggregation

11. Consider a uniform blending classifier G that consists of three binary classifiers $\{g_k\}_{k=1}^3$, where each classifier is of test 0/1 error $E_{\text{out}}(g_1) = 0.08$, $E_{\text{out}}(g_2) = 0.16$, $E_{\text{out}}(g_3) = 0.32$. What is the possible range of $E_{\text{out}}(G)$? Justify your answer.

12. Consider a uniform blending classifier G that consists of K binary classifiers $\{g_k\}_{k=1}^K$, 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 $E_{\text{out}}(G)$.

13. If bootstrapping is used to sample $N' = pN$ examples out of N examples and N is very large, argue that approximately $N - (e^{-p} \cdot N)$ of the examples are sampled at least once.

Kernel for Decision Stumps

When talking about non-uniform voting in aggregation, we mentioned that $\boldsymbol{\alpha}$ can be viewed as a weight vector learned from any linear algorithm coupled with the following transform:

$$\boldsymbol{\phi}(\mathbf{x}) = \Big(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_T(\mathbf{x})\Big).$$

When studying kernel methods, we mentioned that the kernel is simply a computational short-cut for the inner product $(\boldsymbol{\phi}(\mathbf{x}))^T(\boldsymbol{\phi}(\mathbf{x}'))$. In this problem, we mix the two topics together using the decision stumps as our $g_t(\mathbf{x})$.

14. Assume that the input vectors contain only integers between (including) L and R.

$$g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}\Big(x_i - \theta\Big),$$
$$\text{where} \quad i \in \{1, 2, \cdots, d\}, d \text{ is the finite dimensionality of the input space,}$$
$$s \in \{-1, +1\}, \theta \in \mathbb{R}, \text{ and } \text{sign}(0) = +1$$

Two decision stumps g and $\hat{g}$ are defined as the *same* if $g(\mathbf{x}) = \hat{g}(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$. Two decision stumps are different if they are not the same. How many different decision stumps are there for the case of $d = 4$, $L = 0$, and $R = 5$? Explain your answer.

15. Continuing from the previous question, let $\mathcal{G} = \{$ all different decision stumps for $\mathcal{X}$ $\}$ and enumerate each hypothesis $g \in \mathcal{G}$ by some index t. Define

$$\boldsymbol{\phi}_{ds}(\mathbf{x}) = \left(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_t(\mathbf{x}), \cdots, g_{|\mathcal{G}|}(\mathbf{x})\right).$$

Derive a simple equation that evaluates $K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}'))$ efficiently for arbitrary integer tuples (d, L, R) with $d > 0$, and prove your answer.

~~We would give full credit if your solution works for the specific (d, L, R) given by Problem 8~~

16. (Bonus 20%) Solve Problem 15 for "non-integer" input vectors, where there are infinitely many decision stumps.

Yes, A Lighter Homework :-)

17. Which one of our lectures do you like most? Why?

18. Which one of our lectures do you like least? Why?