Homework #6

RELEASE DATE: 12/25/2020

RED BUG FIX: 01/01/2021 16:30

DUE DATE: 01/15/2021, BEFORE 13:00 on Gradescope

RANGE: MOOC LECTURES 207-210, 212, 215

(SELECTED PARTS, WITH BACKGROUND FROM ML FOUNDATIONS)

QUESTIONS ARE WELCOMED ON THE NTU COOL FORUM.

We will instruct you on how to use Gradescope to upload your choices and your scanned/printed solutions. For problems marked with (*), please follow the guidelines on the course website and upload your source code to Gradescope as well. 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 400 points. For each problem, there is one correct choice. For most of the problems, if you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get -10 points. That is, the expected value of random guessing is -20 per problem, and if you can eliminate two of the choices accurately, the expected value of random guessing on the remaining three choices would be 0 per problem. For other problems, the TAs will check your solution in terms of the written explanations and/or code. The solution will be given points between $[-20, 20]$ based on how logical your solution is.

Neural Networks

1. (Lecture 212) A fully connected Neural Network has $L = 3$; $d^{(0)} = 4$, $d^{(1)} = 5$, $d^{(2)} = 6$, $d^{(3)} = 1$. If only products of the form $w_{jk}^{(\ell+1)}\delta_k^{(\ell+1)}$ count as operations, without counting anything else, which of the following is the total number of operations required in a single iteration of computing all $\delta_j^{(\ell)}$ for $\ell \in \{1, 2\}$ and $j \in \{1, 2, \ldots, d^{(\ell)}\}$ on one data point in the backward pass, after all $x_i^{(\ell)}$ and $s_i^{(\ell)}$ are computed and stored in the forward pass, and $\delta_1^{(L)}$ has been computed? Choose the correct answer; explain your answer.

[a] 16

[b] 36

[c] 50

[d] 56

[e] 68

2. (Lecture 212) Consider a Neural Network with $d^{(0)} + 1 = 20$ input units, 3 output units, and 50 hidden units (each $x_0^{(\ell)}$ is also counted as a unit). The hidden units can be arranged in any number of layers $\ell = 1, ..., L-1$. That is,

$$\sum_{\ell=1}^{L-1} \left(d^{(\ell)} + 1\right) = 50.$$

Each layer is fully connected to the layer above it. What is the maximum possible number of weights that such a network can have? Choose the correct answer; explain your answer.

[a] 875
[b] 1123
[c] 1130
[d] 1219
[e] 1327

3. (Lecture 212) Multiclass Neural Network of K classes is typically done by having K output neurons in the last layer. For some given example $(\mathbf{x}, y)$, let $s_k^{(L)}$ be the summed input score to the k-th neuron, the joint "softmax" output vector is defined as

$$\mathbf{x}^{(L)} = \left[\frac{\exp(s_1^{(L)})}{\sum_{k=1}^K \exp(s_k^{(L)})}, \frac{\exp(s_2^{(L)})}{\sum_{k=1}^K \exp(s_k^{(L)})}, \ldots, \frac{\exp(s_K^{(L)})}{\sum_{k=1}^K \exp(s_k^{(L)})}\right].$$

It is easy to see that each $x_k^{(L)}$ is between 0 and 1 and the components of the whole vector sum to 1. That is, $\mathbf{x}^{(L)}$, renamed as $\mathbf{q} \equiv \mathbf{x}^{(L)}$ for short, can be viewed as a vector whose k-th component estimates the probability for $\mathbf{x}$ to be in class k.

Define a one-hot-encoded vector of y to be

$$\mathbf{v} = [[\![y = 1]\!], [\![y = 2]\!], \ldots, [\![y = K]\!]].$$

The cross-entropy loss function for the Multiclass Neural Network, much like an extension of the cross-entropy loss function used in logistic regression, is defined as

$$\text{err}(\mathbf{x}, y) = -\sum_{k=1}^K v_k \ln q_k.$$

What is $\frac{\partial \text{err}}{\partial s_k^{(L)}}$, the $\delta_k^{(L)}$ that you'd need for backpropagation? Choose the correct answer; explain your answer.

[a] q_k
[b] $v_k - q_k$
[c] $(v_k - q_k)q_k$
[d] $q_k - v_k$
[e] $(q_k - v_k)q_k$

(*Hint: The problem can be viewed as the Neural Network extension of Problem 10 of Homework 3 in Machine Learning Foundations*)

4. (Lecture 212) Consider a 4-5-1 Neural Network with all hidden layers having a bias input $x_0^{(\ell)} = +1$ and use $\tanh(s)$ as the transformation functions on all neurons (including the output neuron). Consider a single example $\mathbf{x}_n = (1, 0, 0, 0)$ with $y_n = +1$. Use SGD and backpropagation on this single example to update the weights. Set $\eta = 1$ and initialize all the weights in each $\mathbf{w}^{(\ell)}$ to 0. What is the weight $w_{01}^{(1)}$ after 3 updates? Choose the correct answer; explain your answer.

[a] 0
[b] -2
[c] -4
[d] -6
[e] -8

Matrix Factorization

5. (Lecture 215) Consider a matrix factorization model of $\tilde{d} = 1$ solved with alternating least squares. Assume that the $\tilde{d} \times N$ user factor matrix V is initialized to a constant matrix of 2. After step 2.1 of alternating least squares (Page 10 of Lecture 215), what is w_m, the $\tilde{d} \times 1$ movie "vector" for the m-th movie? Choose the correct answer; explain your answer.

[a] the sum of the ratings on the m-th movie

[b] twice the sum of the m-th movie

[c] the average rating of the m-th movie

[d] twice the average rating of the m-th movie

[e] half the average rating of the m-th movie

6. (Lecture 215) The Matrix Factorization Model tries to find the best $\mathbf{w}_m$ and $\mathbf{v}_n$ such that $r_{nm} \approx \mathbf{w}_m^T \mathbf{v}_n$. Sometimes, we can make the model more expressive by introducing bias term. That is, we try to approximate r_{nm} by $\mathbf{w}_m^T \mathbf{v}_n + a_m + b_n$. Then, the per-example error function on Page 14 of Lecture 215 becomes

$$\text{err}(\text{user } n, \text{movie } m, \text{rating } r_{nm}) = (r_{nm} - \mathbf{w}_m^T \mathbf{v}_n - a_m - b_n)^2.$$

Which of the following corresponds to how a_m should be updated when running SGD for this new matrix factorization model with a learning rate $\frac{\eta}{2}$? Choose the correct answer; explain your answer.

[a] $a_m \leftarrow (1 - \eta) a_m - \eta \cdot (r_{nm} - \mathbf{w}_m^T \mathbf{v}_n - b_n)$

[b] $a_m \leftarrow (1 - \eta) a_m + \eta \cdot (r_{nm} - \mathbf{w}_m^T \mathbf{v}_n - b_n)$

[c] $a_m \leftarrow (1 + \eta) a_m - \eta \cdot (r_{nm} - \mathbf{w}_m^T \mathbf{v}_n - b_n)$

[d] $a_m \leftarrow (1 + \eta) a_m + \eta \cdot (r_{nm} - \mathbf{w}_m^T \mathbf{v}_n - b_n)$

[e] $a_m \leftarrow a_m + \eta \cdot (r_{nm} - \mathbf{w}_m^T \mathbf{v}_n - b_n)$

Aggregation

7. (Lecture 207) For a binary classification task, assume that there are 3 binary classifiers g_1, g_2, g_3. If uniform blending is used to blend the three classifiers to get G like Page 7 of Lecture 207, and $E_{\text{out}}(G) = 0.20$. Which of the following is a possible combination of $[E_{\text{out}}(g_1), E_{\text{out}}(g_2), E_{\text{out}}(g_3)]$? Here E_{out} is measured by the 0/1 error. Choose the correct answer; explain your answer.

[a] $[0.04, 0.16, 0.16]$

[b] $[0.04, 0.08, 0.24]$

[c] $[0.06, 0.04, 0.16]$

[d] $[0.16, 0.08, 0.24]$

[e] $[0.04, 0.06, 0.24]$

8. (Lecture 207) For a binary classification task, assume that there are 5 binary classifiers g_1, g_2, …, g_5, and for some $P(\mathbf{x}, y)$, the errors made by the 5 classifiers are independent. That is, the five random variables $[\![y \neq g_1(\mathbf{x})]\!]$, $[\![y \neq g_2(\mathbf{x})]\!]$, …, $[\![y \neq g_5(\mathbf{x})]\!]$ are independent. Assume that $E_{\text{out}}(g_t) = 0.4$ for $t = 1, 2, \ldots, 5$, if uniform blending is used to blend the five classifiers to get G like Page 7 of Lecture 207, what is $E_{\text{out}}(G)$? Choose the closest answer; explain your answer.

[a] 0.68

[b] 0.40

[c] 0.32

[d] 0.08

[e] 0.01

9. (Lectures 207/210) If bootstrapping is used to sample exactly $0.5N$ examples out of N, what is the probability that an example is *not* sampled when N is very large? Choose the closest answer; explain your answer.

[a] 77.9%

[b] 60.7%

[c] 36.8%

[d] 13.5%

[e] 1.8%

10. (Lecture 207) 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 $g_t(\mathbf{x})$.

Assume that the input vectors contain only even integers between (including) $2L$ and $2R$, where $L < R$. Consider the decision stumps $g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}\Big(x_i - \theta\Big)$, where

$$\begin{array}{ll} i & \in \{1, 2, \cdots, d\}, \\ d & \text{is the finite dimensionality of the input space,} \\ s & \in \{-1, +1\}, \\ \theta & \text{is an odd integer between } (2L, 2R). \end{array}$$

Define $\boldsymbol{\phi}_{ds}(\mathbf{x}) = \Big(g_{+1,1,2L+1}(\mathbf{x}), g_{+1,1,2L+3}(\mathbf{x}), \ldots, g_{+1,1,2R-1}(\mathbf{x}), \ldots, g_{-1,d,2R-1}(\mathbf{x})\Big)$. What is $K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}'))$? Choose the correct answer; explain your answer.

[a] $2d(R-L) - \|\mathbf{x} - \mathbf{x}'\|_1$

[b] $2d(R-L)^2 - \|\mathbf{x} - \mathbf{x}'\|_1^2$

[c] $2d(R-L) - \|\mathbf{x} - \mathbf{x}'\|_2$

[d] $2d(R-L)^2 - \|\mathbf{x} - \mathbf{x}'\|_2^2$

[e] none of the other choices

Adaptive Boosting

11. (Lecture 208) Consider applying the AdaBoost algorithm on Page 17 of Lecture 208 to a binary classification data set where 95% of the examples are negative. Because there are so many negative examples, the base algorithm within AdaBoost returns a constant classifier $g_1 = -1$ in the first iteration. Let $u_+^{(2)}$ be the individual example weight of each positive example in the second iteration, and $u_-^{(2)}$ be the example weight of each negative example in the second iteration. What is $\frac{u_+^{(2)}}{u_-^{(2)}}$? Choose the correct answer; explain your answer.

[a] 19

[b] 1/19

[c] 1

[d] 20

[e] 1/20

12. (Lectures 208/211) For the AdaBoost algorithm on Page 17 of Lecture 208, let $U_t = \sum_{n=1}^{N} u_n^{(t)}$. In Lecture 211, it is shown that for any integer $t > 0$, $U_{t+1} = \frac{1}{N} \sum_{n=1}^{N} \exp\left(-y_n \sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x}_n)\right)$, and that $E_{\text{in}}(G_T) \leq U_{T+1}$. Assume that $0 < \epsilon_t \leq \epsilon < \frac{1}{2}$ for each hypothesis g_t, which of the following is correct? Choose the correct answer; explain your answer.

[a] $E_{\text{in}}(G_T) \leq \exp\left(-2T^2(\frac{1}{2} - \epsilon)^2\right)$

[b] $E_{\text{in}}(G_T) \leq \exp\left(-2T\sqrt{T}(\frac{1}{2} - \epsilon)^2\right)$

[c] $E_{\text{in}}(G_T) \leq \exp\left(-4T(\frac{1}{2} - \epsilon)^2\right)$

[d] $E_{\text{in}}(G_T) \leq \exp\left(-2T(\frac{1}{2} - \epsilon)^2\right)$

[e] none of the other choices

(*Hint: It might be helpful to consider checking* $\frac{U_{t+1}}{U_t}$*, and use the fact that*

$$\sqrt{\epsilon(1-\epsilon)} \leq \frac{1}{2}\exp(-2(\frac{1}{2} - \epsilon)^2)$$

for all $0 < \epsilon < \frac{1}{2}$).

Decision Tree

13. (Lecture 209) Impurity functions play an important role in decision tree branching. For binary classification problems, let μ_+ be the fraction of positive examples in a data subset, and $\mu_- = 1 - \mu_+$ be the fraction of negative examples in the data subset. We can normalize each impurity function by dividing it with its maximum value among all $\mu_+ \in [0, 1]$. For instance, the classification error is simply $\min(\mu_+, \mu_-)$ and its maximum value is 0.5. So the normalized classification error is $2\min(\mu_+, \mu_-)$. After normalization, which of the following impurity function is equivalent to the classification error $\min(\mu_+, \mu_-)$? Choose the correct answer; explain your answer.

[a] the Gini index $1 - \mu_+^2 - \mu_-^2$

[b] the squared error (used for branching in classification data sets), which is by definition $\mu_+(1 - (\mu_+ - \mu_-))^2 + \mu_-(-1 - (\mu_+ - \mu_-))^2$

[c] the entropy, which is $-\mu_+ \ln \mu_+ - \mu_- \ln \mu_-$, with $0 \ln 0 \equiv 0$

[d] the closeness, which is $1 - |\mu_+ - \mu_-|$

[e] none of the other choices

Experiments with Decision Tree and Random Forest

In the following questions, you are asked to implement a preliminary random forest algorithm. You need to implement everything by yourself without using any well-implemented packages.

14. (Lecture 209, *) First, let's implement a simple C&RT algorithm without pruning using the Gini index as the impurity measure, as introduced in the class. For the decision stump used in branching, if you are branching with feature i, please sort all the $x_{n,i}$ values to form (at most) $N+1$ segments of equivalent θ, and then pick θ within the median of the segment. If multiple (i, θ) produce the best split, pick the one with the smallest i (and if there is a tie again, pick the one with the smallest θ).

Please run the algorithm on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw6/hw6_train.dat`

and the following file as our test data set for evaluating E_{out}:

`http://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw6/hw6_test.dat`

What is the $E_{\text{out}}(g)$, where g is the unpruned decision tree returned from your C&RT algorithm and E_{out} is evaluated using the 0/1 error? Choose the closest answer; provide your code.

[a] 0.08

[b] 0.13

[c] 0.18

[d] 0.23

[e] 0.28

15. (Lectures 207/210, *) Next, we implement the random forest algorithm by coupling bagging (by sampling with replacement) with $N' = 0.5N$ with your unpruned decision tree in the previous problem. Produce $T = 2000$ trees with bagging. Let $g_1, g_2, \ldots, g_{2000}$ denote the 2000 trees generated. What is $\frac{1}{T}\sum_{t=1}^{T} E_{\text{out}}(g_t)$?, where E_{out} is also evaluated using the 0/1 error? Choose the closest answer; provide your code.

[a] 0.08

[b] 0.13

[c] 0.18

[d] 0.23

[e] 0.28

16. Let $G(\mathbf{x}) = \text{sign}(\sum_{t=1}^{T} g_t(\mathbf{x}))$ be the random forest formed by the trees above. What is $E_{\text{in}}(G)$, where E_{in} is evaluated using the 0/1 error? Choose the closest answer; provide your code.

[a] 0.01

[b] 0.06

[c] 0.11

[d] 0.16

[e] 0.21

17. Following the previous problem, what is $E_{\text{out}}(G)$, where E_{out} is evaluated using the 0/1 error? Choose the closest answer; provide your code.

[a] 0.01

[b] 0.06

[c] 0.11

[d] 0.16

[e] 0.21

18. Following the previous problem, we can calculate $E_{\text{oob}}(G)$ as

$$\frac{1}{N}\sum_{n=1}^{N}\text{err}(y_n, G_n^-(\mathbf{x}_n)),$$

where G_n^- is a random forest that contains all the trees that were *not* trained with $\mathbf{x}_n$. If all trees are trained with $\mathbf{x}_n$, take G_n^- as a constant classifier that always returns -1. Let err be the 0/1 error. What is $E_{\text{oob}}(G)$? Choose the closest answer; provide your code.

[a] 0.02

[b] 0.07

[c] 0.12

[d] 0.17

[e] 0.22

Learning Comes from Feedback

19. Which topic of this class do you like the most? Choose one topic; explain your choice.

[a] support vector machine

[b] matrix factorization

[c] aggregation models: non-boosting ones

[d] aggregation models: AdaBoost and Gradient Boosting

[e] neural networks and deep learning

20. Which topic of this class do you like the least? Choose one topic; explain your choice.

[a] support vector machine

[b] matrix factorization

[c] aggregation models: non-boosting ones

[d] aggregation models: AdaBoost and Gradient Boosting

[e] neural networks and deep learning