CS 688: Homework 4

Due: Friday Apr 12 by 10:00 PM

Notes:

- Homework is due **by 10:00 PM on the due date.** Remember that you may not use more than 2 late days on any one homework, and you only have a budget of 5 in total.
- Please keep in mind the collaboration policy as specified in the course syllabus. If you discuss questions with others you **must** write their names on your submission, and if you use any outside resources you **must** reference them. **Do not look at each others' writeups, including code.**
- There are 4 problems on 3 pages in this homework.
- Remember that you will be graded on the quality of your report and writing. We may or may not even look at your code. So please write carefully and clearly.
- For your code, please create a single zip file labeled `yourlastname_hw4.zip` and submit it to the "HW4 Code" assignment on Gradescope. You should provide a readme explaining how to run your code and replicate the results in your report.
- All graphs should have clearly labeled axes.

Problems:

1. (50 points) Implement AdaBoost using decision stumps (depth 1 decision trees) learned using information gain as the weak learners (you may use any library function you wish to implement the learning of decision stumps themselves, but you may not use any libraries for writing the boosting wrapper).

   You will be applying this to the problem of classifying handwritten digits. We have provided a training set and a test set based on data initially made available by Yann LeCun and colleagues in 1990, with normalized handwritten digits, automatically scanned from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations; the images here have been deslanted and size normalized, resulting in 16 x 16 grayscale images. Each line consists of the digit id (0-9) followed by the 256 grayscale values. There are 7291 training observations and 2007 test observations. You will focus on two different problems: classifying a handwritten digit as either "1" or "3" and classifying a handwritten digit as either "3" or "5" – you will need to appropriate preprocess the provided files in order to get to these problems. The training set contains 1005 "1"s, 658 "3"s, and 556 "5"s, while the test set contains 264 "1"s, 166 "3"s, and 160 "5"s.

   Graphically report the training set error and the test set error as a function of the number of weak hypotheses, and summarize and interpret your results. Note that you have plenty of leeway in deciding what to include – you will be graded on how interesting and correct your report is, and on what kinds of insight you provide.

2. (10 points) Consider a regression problem where the labels are noisy, that is $y_i = f(\mathbf{x}_i) + \epsilon$, and, in particular ϵ is a zero-mean variable with variance σ^2. Show that the bias-variance decomposition becomes:

$$\mathbb{E}_{\mathcal{D}}[L(g^{(\mathcal{D})})] = \text{bias} + \text{var} + \sigma^2$$

(where $\text{bias}(x) = (\bar{g}(x) - f(x))^2$ – Bishop calls this squared bias).

3. (15 points) Consider a situation where each x_i is single-dimensional and known to be uniformly distributed on $[-1, 1]$. The true target function is $y = x^2$. However, the hypothesis space $\mathcal{H}$ has only linear functions $h(x) = ax + b$. Imagine that each training set consists of two examples, $(x_1, x_1^2), (x_2, x_2^2)$. Denote the actual hypothesis learned on some training set as $g(x)$.

   (a) What is $\bar{g}(x)$, the "mean hypothesis"?

   (b) What are $\bar{a}, \bar{b}, \bar{a^2}, \bar{b^2}$?

   (c) Use the above to analytically find the bias, variance, and mean squared error in expectation.

4. (25 points) Backpropagation: Consider the neural network in Figure 1. We will work through backpropagation on this network.

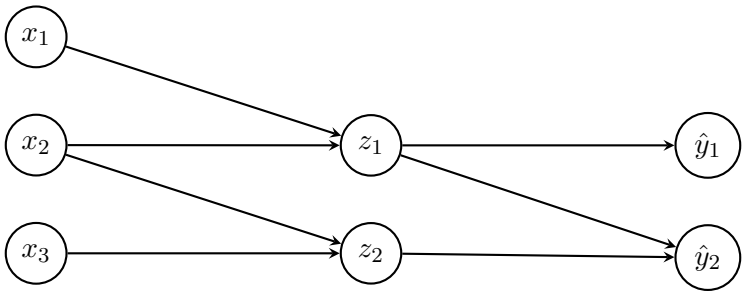

Figure 1: A Directed Acyclic Graph (DAG)

Let $(\mathbf{x}, \mathbf{y})$ be the training example that is considered, where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ and $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$. All the other nodes in the graph are defined as:

$$z_1 = \text{ReLU}(w_{1,1}x_1 + w_{2,1}x_2)$$
$$z_2 = w_{2,2}x_2^2 + w_{3,2}x_3 + b_2$$
$$\hat{y}_1 = \sigma(m_{1,1}z_1^3 + c_1)$$
$$\hat{y}_2 = m_{1,2}\sin(z_1) + m_{2,2}\cos(z_2)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, and $\text{ReLU}(x) = \max(0, x)$. Let $\boldsymbol{\theta}$ be the set of all parameters to be learned in this graph. We have that

$$\boldsymbol{\theta} = \{w_{1,1}, w_{2,1}, w_{2,2}, w_{3,2}, m_{1,1}, m_{1,2}, m_{2,2}, b_2, c_1\}$$

For every set of inputs $\mathbf{x} = (x_1, x_2, x_3)$, we will define objective of the problem as minimizing the loss function

$$J(\boldsymbol{\theta}) = \log\left((y_1 - \hat{y}_1)^2\right) + \log\left((y_2 - \hat{y}_2)^2\right)$$

Assume that you have already gone through the forward pass with inputs $\mathbf{x} = (x_1, x_2, x_3)$ and stored all the relevant values. In the following questions, you will derive the backpropagation algorithm applied to the above DAG. In what follows, $w_{i,j}$ refers to the weight associated with the edge connected

from node x_i to z_j, and similarly $m_{i,j}$ refers to the edge weight from node z_i to y_j.

(a) First, we will derive the gradients with repect to the outputs. What are the expressions for $\frac{\partial J}{\partial \hat{y}_1}$? Write your solution in terms of $\hat{y}_1$.

(b) Now, we will derive the gradients associated with the last layer, ie nodes y_1, y_2. Note that for the full backpropagation algorithm, you would need to calculate the gradients of the loss function with respect to every parameter $(m_{1,1}, m_{1,2}, m_{2,2}, c_1)$ as well as every input of the layer (z_1, z_2), but we are not asking for all of them in this part. For all of the questions in this part, you should use Chain Rule, and write your solution in terms of values from the forward pass, or **gradients with respect to the outputs of this layer,** $\frac{\partial \mathbf{J}}{\partial \hat{\mathbf{y}}_1}$, $\frac{\partial \mathbf{J}}{\partial \hat{\mathbf{y}}_2}$, because you have already calculated these values. In addition, use the sigmoid function $\sigma(x)$ in your answer instead of its explicit form.

i. What is the expression for $\frac{\partial J}{\partial z_1}$?

ii. What is the expression for $\frac{\partial J}{\partial z_2}$?

iii. What is the expression for $\frac{\partial J}{\partial m_{1,1}}$?

iv. What is the expression for $\frac{\partial J}{\partial m_{1,2}}$?

v. What is the expression for $\frac{\partial J}{\partial c_1}$?

(c) Lastly, we will derive the gradients associated with the second layer, ie nodes z_1, z_2. Note that for the full backpropagation algorithm, you need to calculate the gradients of the loss function with respect to every parameter (every $w_{i,j}, b_2$). However, we do not need to calculate the gradients with respect to the inputs of this layer (x_1, x_2, x_3), because they are fixed inputs of the model. For all of the questions in this part, you should use Chain Rule, and write your solution in terms of values from the forward pass or **gradients with respect to the outputs of this layer,** $\frac{\partial \mathbf{J}}{\partial \mathbf{z}_1}$, $\frac{\partial \mathbf{J}}{\partial \mathbf{z}_2}$, because you have already calculated these values.

i. What is the expression for $\frac{\partial J}{\partial w_{2,2}}$?

ii. What is the expression for $\frac{\partial J}{\partial b_2}$?

iii. Recall that $\text{ReLU}(x) = \max(x, 0)$. The ReLU function is not differentiable at $x = 0$, but for backpropagation, we define its derivative as

$$\text{ReLU}'(x) = \begin{cases} 0, x < 0 \\ 1, \text{ otherwise} \end{cases}$$

Now, what is the expression for $\frac{\partial J}{\partial w_{1,1}}$? Explicitly write out the cases.