CPSC 483/583: Deep Learning on Graph-Structured Data Yale University

Assignment 1

Out: August 31, 2024 Due: September 13, 2024

General Instructions

These questions require thought, but do not require long answers. Please be as concise as possible. You are allowed to take a maximum of 1 late period (see the course website or slides about the definition of a late period).

Submission instructions: You should submit your answers in a *single* PDF file. LaTeX is highly preferred due to the need of formatting equations.

Submitting answers: Prepare answers to your homework in a *single* PDF file. Make sure that the answer to each sub-question is on a *separate page*. The number of the question should be at the top of each page.

Honor Code: When submitting the assignment, you agree to adhere to the Yale Honor Code. Please read carefully to understand what it entails!

Symbol-wise: we use lowercase bold like $\mathbf{x}$ *to indicate a vector and uppercase bold like* $\mathbf{W}$ *to indicate a matrix. Others are scalars. Dimensions will be explicitly clarified if needed.*

1 Sigmoid and Softmax Function

Sigmoid function is used in the logistic regression to predict the label $l \in \{0, 1\}$ given a sample $\mathbf{x} \in \mathbb{R}^d$. The sigmoid function is defined as follows, which maps value in $\mathbb{R}$ to a number in $(0, 1)$.

$$\sigma(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$$

Then the probabilities that label l equals to 0 or 1 are

$$P(l=1|\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^\top\mathbf{x}}} = \sigma(\mathbf{w}^\top\mathbf{x}) \quad \text{and} \quad P(l=0|\mathbf{x}) = 1 - P(l=1|\mathbf{x}),$$

where $\mathbf{w} \in \mathbb{R}^d$ is the vector of weights. Softmax function is a generalization of logistic regression to multiple classes, i.e., $l \in \{0, 1, \cdots, K-1\}$, where K denotes the number of classes. For a vector $\mathbf{z} \in \mathbb{R}^K$, softmax function normalizes it into a probability distribution by

$$Softmax(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{K-1} e^{z_j}} \text{ for } i = 0, 1, \cdots, K-1 \text{ where } \mathbf{z} = (z_0, z_1, \cdots, z_{K-1}) \in \mathbb{R}^K.$$

For an input vector $\mathbf{x} \in \mathbb{R}^d$, Softmax estimates the probability of each label using a weight matrix $\mathbf{W} \in \mathbb{R}^{K\times d}$ by

$$Softmax(\mathbf{W}\mathbf{x}) = \begin{bmatrix} P(l=0|\mathbf{x}, \mathbf{w}_0) \\ P(l=1|\mathbf{x}, \mathbf{w}_1) \\ \cdots \\ P(l=K-1|\mathbf{x}, \mathbf{w}_{K-1}) \end{bmatrix} = \frac{1}{\sum_{i=0}^{K-1} e^{\mathbf{w}_i^\top\mathbf{x}}} \begin{bmatrix} e^{\mathbf{w}_0^\top\mathbf{x}} \\ e^{\mathbf{w}_1^\top\mathbf{x}} \\ \cdots \\ e^{\mathbf{w}_{K-1}^\top\mathbf{x}} \end{bmatrix}.$$

The matrix $\mathbf{W}$ is formed by the weight vectors as $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{K-1}]^\top$. It is easy to verify that the sum of all elements in the output of Softmax function is 1.

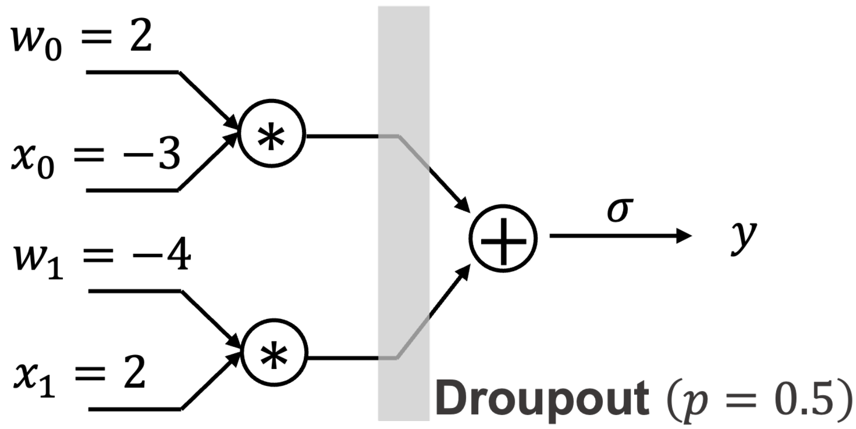
1. Assuming $t = \sigma(s)$, calculate the gradient of Sigmoid function with respect to s and rewrite the gradient as a function of t (i.e., there is no s in the gradient expression).

2. Prove that Softmax function is invariant to a weight shift. Let $\mathbf{W}' = [\mathbf{w}_0 - \mathbf{c}, \mathbf{w}_1 - \mathbf{c}, \cdots, \mathbf{w}_{K-1} - \mathbf{c}]^\top$, where $\mathbf{c}$ is a constant vector that we subtract from each elements in $\mathbf{W}$. Weight-shift invariance implies that $Softmax(\mathbf{W}'\mathbf{x}) = Softmax(\mathbf{W}\mathbf{x})$. (Hint: $e^{x-c} = e^x \cdot e^{-c}$)

3. Prove that when $K = 2$, Softmax-based logistic regression is equivalent to Sigmoid-based logistic regression. (Hint: use the weight-shift invariance property of Softmax function to prove the probabilities of label 0 and 1 estimated by Sigmoid and Softmax are equivalent).

2 Back-Propagation

1. Let's perform back-propagation through a neural network with a sigmoid activation σ. Specifically, we insert a dropout layer before the activation. The computation graph is visualized below.



Therefore, the output $y = \sigma(\delta_0 w_0 x_0 + \delta_1 w_1 x_1)$, where $\delta_0, \delta_1 \sim \text{Bernoulli}(0.5)$. Calculate the expectation of gradients with respect to the input and parameters, i.e., $\mathbb{E}(\frac{\partial y}{\partial w_0})$, $\mathbb{E}(\frac{\partial y}{\partial x_0})$, $\mathbb{E}(\frac{\partial y}{\partial w_1})$, $\mathbb{E}(\frac{\partial y}{\partial x_1})$.

2. In a fully connected layer, let $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^{d_k}$ denotes the input, $\mathbf{W} \in \mathbb{R}^{d_{k+1} \times d_k}$ is the weight matrix corresponding to this fully connected layer and $\mathbf{y} \in \mathbb{R}^{d_{k+1}}$ denotes the output. $\mathcal{L}(\mathbf{y}) \in \mathbb{R}$ is the loss function. Prove that the gradient back-propagation is also in the form of a fully connected layer, where the gradient of $\mathcal{L}$ with respect to $\mathbf{y}$ is the input and the gradient of $\mathcal{L}$ with respect to $\mathbf{x}$ is the output. What is the relationship between the weight matrix of this fully connected layer and $\mathbf{W}$? (Hint: prove that $\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \tilde{\mathbf{W}} \frac{\partial \mathcal{L}}{\partial \mathbf{y}}$ with a certain weight matrix $\tilde{\mathbf{W}}$ and figure out the relationship between $\tilde{\mathbf{W}}$ and $\mathbf{W}$. Also, please treat $\frac{\partial \mathcal{L}}{\partial \mathbf{x}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{y}}$ as row vectors.)

3. We have a two-layer neural network as follows,

$$f(\mathbf{x}) = \sigma(\sigma(\mathbf{x} \cdot \mathbf{W}^{(1)}) \cdot \mathbf{W}^{(2)}),$$

where σ is the sigmoid function, $\mathbf{x} \in \mathbb{R}^{d_1}$ is the input, $\mathbf{W}^{(1)} \in \mathbb{R}^{d_1 \times d_2}$ and $\mathbf{W}^{(2)} \in \mathbb{R}^{d_2 \times 1}$ are weight matrices of the first and second layer, respectively. Show the gradient of the two-layer neural network's output $f(\mathbf{x})$ with respect to parameters $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ (Hint: use the chain rule and results in Q2.2).