Homework 3
600.482/682 Deep Learning
Spring 2020

February 21, 2020

Due Sun. 03/01/2020 11:59:00pm.
Please submit a latex generated PDF
to Gradescope with entry code 9G83Y7

1. We have talked about backpropagation in class. And here is a supplementary material for calculating the gradient for backpropagation (https://piazza.com/class_profile/get_resource/jxcftju833c25t/k0labsf3cny4qw). Please study this material carefully before you start this exercise. Suppose $P = WX$ and $L = f(P)$ which is a loss function.

   (a) Please show that $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial P} X^T$. Show each step of your derivation.

   (b) Suppose the loss function is L2 loss. L2 loss is defined as $L(y, \hat{y}) = \|y - \hat{y}\|^2$ where y is the groundtruth; $\hat{y}$ is the prediction. Given the following initialization of W and X, please calculate the updated W after one iteration. (step size = 0.1)

$$W = \begin{pmatrix} 0.3 & 0.5 \\ -0.2 & 0.4 \end{pmatrix}, X = (\mathbf{x_1}, \mathbf{x_2}) = \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}, Y = (\mathbf{y_1}, \mathbf{y_2}) = \begin{pmatrix} 0.5 & 1 \\ 1 & -1.5 \end{pmatrix}$$

2. In this exercise, we will explore how vanishing and exploding gradients affect the learning process. Consider a simple, 1-dimensional, 3 layer network with data $x \in \mathbb{R}$, prediction $\hat{y} \in [0, 1]$, true label $y \in \{0, 1\}$, and weights $w_1, w_2, w_3 \in \mathbb{R}$, where weights are initialized randomly via $\sim \mathcal{N}(0, 1)$. We will use the sigmoid activation function σ between all layers, and the cross entropy loss function $L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$. This network can be represented as: $\hat{y} = \sigma(w_3 \cdot \sigma(w_2 \cdot \sigma(w_1 \cdot x)))$. Note that for this problem, we are not including a bias term.

   (a) Compute the derivative for a sigmoid. What are the values of the extrema of this derivative, and when are they reached?

   (b) Consider a random initialization of $w_1 = 0.25, w_2 = -0.11, w_3 = 0.78$, and a sample from the data set $(x = 0.63, y = 1)$. Using backpropagation, compute the gradients for each weight. What have you noticed about the magnitude of the gradient?

   Now consider that we want to switch to a regression task and use a similar network structure as we did above: we remove the final sigmoid activation, so our new network is defined as $\hat{y} = w_3 \cdot \sigma(w_2 \cdot \sigma(w_1 \cdot x))$, where predictions $\hat{y} \in \mathcal{R}$ and targets $y \in \mathcal{R}$; we use the L2 loss function instead of cross entropy: $L(y, \hat{y}) = (y - \hat{y})^2$. Derive the gradient of the loss function with respect to each of the weights w_1, w_2, w_3.

   (c) Consider again the random initialization of $w_1 = 0.25, w_2 = -0.11, w_3 = 0.78$, and a sample from the data set $(x = 0.63, y = 128)$. Using backpropagation, compute the gradients for each weight. What have you noticed about the magnitude of the gradient?