Homework 3
600.482/682 Deep Learning
Spring 2022
Ting He

February 21, 2022

Important: Please use the natural logarithm (base e) for all logarithm calculations (log) below. Please show each step of your calculations; points will be deducted if only final results are present.

1. We have presented a matrix back propagation example in class. In this exercise, you will follow the same logic we used in class to derive $\frac{\partial L}{\partial X} = W^T \frac{\partial L}{\partial Y}$.

   (a) Please derive $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} X^T$ (please consider the **general case** and show each step of your computation).

   (b) Suppose the loss function is L2 loss. Given the following initialization of W and X, please calculate the updated W after one iteration. (step size = 0.1)

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

   Hint: L2 loss is defined by $L_2(Y, \hat{Y}) = (y_{11} - \hat{y_{11}})^2 + (y_{12} - \hat{y_{12}})^2 + (y_{21} - \hat{y_{21}})^2 + (y_{22} - \hat{y_{22}})^2$, where $Y = WX$.

A: Given $L = f(W, x) : \mathbb{R}^{m*d} \times \mathbb{R}^{d*n} \mapsto \mathbb{R}^{m*n} \mapsto \mathbb{R}$

We have, $W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1d} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & \dots & w_{md} \end{bmatrix}$, $X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{d1} & x_{d2} & \dots & x_{dn} \end{bmatrix}$

Then

$$Y = WX$$
$$= \begin{bmatrix} w_{11}x_{11}+w_{12}x_{21}+\dots+w_{1d}x_{d1} & w_{11}x_{12}+w_{12}x_{22}+\dots+w_{1d}x_{d2} & \dots & w_{11}x_{1n}+w_{12}x_{2n}+\dots+w_{1d}x_{dn} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1}x_{11}+w_{m2}x_{21}+\dots+w_{md}x_{d1} & w_{m1}x_{11}+w_{m2}x_{22}+\dots+w_{md}x_{d2} & \dots & w_{m1}x_{1n}+w_{m2}x_{2n}+\dots+w_{md}x_{dn} \end{bmatrix}$$

In general form, we have ($i \leq m$ and $x_{j:}$ is all the items in the ith line of matrix)

$$\frac{\partial Y}{\partial w_{ij}} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{j1} & x_{j2} & \dots & x_{jd} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

In this way, we have,

$$\begin{aligned}
\frac{\partial L}{\partial w_{ij}} &= \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial w_{ij}} \\
&= \sum_k \frac{\partial L}{\partial Y_{ik}} X_{jk} \\
&= \sum_k \frac{\partial L}{\partial Y_{ik}} X^T_{kj} \\
&= \frac{\partial L}{\partial Y} X^T
\end{aligned}$$

B. given the initialization of W, X, true label of Y^2, L2 loss function, step size of 0.1, and gradient of W: $\frac{\partial Y}{\partial W} = X$, we can calculate the updated weight matrix by subtraction $step_size*gradient$ by current weight matrix.

$$\begin{aligned}
Y = WX &= \begin{bmatrix} 0.3 & 0.5 \\ -0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1.5 & 1.1 \\ 1.2 & 0 \end{bmatrix} \\
L_2(Y, \hat{Y}) &= \begin{bmatrix} (1.5-0.5)^2 & (1.1-1)^2 \\ (1.2-1)^2 & (0+1.5)^2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0.01 \\ 0.04 & 1.25 \end{bmatrix} \\
\frac{\partial L_2(Y = WX, \hat{Y})}{\partial W} &= \frac{\partial L_2}{\partial Y}\frac{\partial Y}{\partial W} \\
&= 2(Y - \hat{Y})X^T = 2\begin{bmatrix} 1 & 0.1 \\ 0.2 & 1.5 \end{bmatrix} \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0.4 & 6.2 \\ 6 & 4.2 \end{bmatrix} \\
updated_weights &= weights - step_size * gradient \\
&= \begin{bmatrix} 0.3 & 0.5 \\ -0.2 & 0.4 \end{bmatrix} - 0.1\begin{bmatrix} 0.4 & 6.2 \\ 6 & 4.2 \end{bmatrix} \\
&= \begin{bmatrix} 0.26 & -0.12 \\ -0.8 & -0.02 \end{bmatrix}
\end{aligned}$$

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 $y \in [0, 1]$, true label $\hat{y} \in \{0, 1\}$, 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 $L(y, \hat{y}) = -(\hat{y}\log(y) + (1-\hat{y})\log(1-y))$. This network can be represented as: $y = \sigma(w_3 \cdot \sigma(w_2 \cdot \sigma(w_1 \cdot x)))$. 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? *Hint:* Please consider both maximum and minimum extrema.

   (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 $(x = 0.63, \hat{y} = 1)$. Using backpropagation, compute the gradients for each weight. What do you notice about the magnitude?

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

(c) Derive the gradient of the loss function with respect to each of the weights w_1, w_2, w_3.

(d) 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 $(x = 0.63, \hat{y} = 128)$. Using backpropagation, compute the gradients for each weight. What do you notice about the magnitude?

A.

$$\sigma(x) = \frac{1}{1+e^{-x}}$$
$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$$

when $x = 0$, $\sigma'(x)$ reaches its the largest value of 0.25; when $x = -\infty$, $\sigma'(x) = -\infty$; while $x = \infty$, $\sigma'(x) = -\infty$

B.

from question A, we get $\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$ and we can get $L'(y) = -\frac{\hat{y}}{y} + \frac{\hat{y}-1}{1-y}$ easily. During the forward process, we can easily calculate the some key values for middle steps:

$$\begin{aligned}
w_1 x &= 0.16 \\
\sigma(0.16) &= 0.54 \\
w_2\sigma(0.16) &= -0.059 \\
\sigma(-0.059) &= 0.49 \\
w_3 * 0.49 &= 0.38 \\
\sigma(0.38) &= 0.41 \\
L(0.41) &= 0.89
\end{aligned}$$

Then utilize the above middle values and derivatives, we have:

$$\begin{aligned}
\frac{\partial L(y = f(w1, w2, w3, w4, x), \hat{y})}{\partial w_3} &= (\frac{-1}{0.41} + 0)(\frac{e^{-0.38}}{(1+e^{-0.38})^2}) * 0.49 \\
&= \frac{-1}{0.41} * 0.49 \\
&= -0.29 \\
\frac{\partial L(y = f(w1, w2, w3, w4, x), \hat{y})}{\partial w_2} &= (\frac{-1}{0.41} + 0)(\frac{e^{-0.38}}{(1+e^{-0.38})^2}) * 0.78 * (\frac{e^{0.059}}{(1+e^{0.059})^2}) * 0.54 \\
&= -0.062 \\
\frac{\partial L(y = f(w1, w2, w3, w4, x), \hat{y})}{\partial w_1} &= (\frac{-1}{0.41} + 0)(\frac{e^{-0.38}}{(1+e^{-0.38})^2}) * 0.78 * (\frac{e^{0.059}}{(1+e^{0.059})^2}) * (-0.11) * 0.63 \\
&= 0.0079
\end{aligned}$$

I can see from the back-propagation process, gradients are getting smaller and smaller from w1, w2 to w3 since we always multiple the derivatives using a number smaller than 1 and the derivation of sigmoid function is close to 0 when x getting larger or smaller also for cross-entropy loss when x getting larger . I assume when we repeat more iterations, there won't have large changes on weights and easily reach a point of local minimum or paddle point (gradient vanish).

C.

The forward process can be written into following functions:

$$\begin{aligned}
y_1 &= w_1 x \\
y_2 &= \sigma(w_1 x) = \sigma(y_1(w_1, x)) \\
y_3 &= w_2\sigma(w_1 x) = w_2 y_2(w_1, x) \\
y_4 &= \sigma(w_2\sigma(w_1 x)) = \sigma(y_3(w_1, w_2, x)) \\
y_5 &= w_3\sigma(w_2\sigma(w_1 x)) = w_3 y_4(w_1, w_2, x) \\
y &= y_6 = L(y_5, \hat{y}) \\
L(y, \hat{y}) &= (\hat{y} - y)^2
\end{aligned}$$

by calculating their derivatives, we have:

$$\frac{\partial y_6(w_1,w_2,w_3,x)}{\partial w_3} = 2(y-\hat{y}) * [\sigma(w_2\sigma(w_1x))]$$
$$\frac{\partial y_6(w_1,w_2,w_3,x)}{\partial w_2} = 2(y-\hat{y}) * w_3 * \sigma'(w_2\sigma(w_1x))$$
$$\frac{\partial y_6(w_1,w_2,w_3,x)}{\partial w_1} = 2(y-\hat{y}) * w_3 * \sigma'(w_2\sigma(w_1x)) * \sigma'(w_1x)x$$

D.
given $w_1 = 0.25, w_2 = -0.11, w_3 = 0.78, x = 0.63, \hat{y} = 128$

$$\begin{aligned}
y_1 &= w_1x = 0.16 \\
y_2 &= \sigma(w_1x) = \sigma(y_1(w_1,x)) = 0.54 \\
y_3 &= w_2\sigma(w_1x) = w_2y_2(w_1,x) = -0.059 \\
y_4 &= \sigma(w_2\sigma(w_1x)) = \sigma(y_3(w_1,w_2,x)) = 0.49 \\
y_5 &= w_3\sigma(w_2\sigma(w_1x)) = w_3y_4(w_1,w_2,x) = 0.38 \\
y &= y_6 = L(y_5,\hat{y}) = 16287.25 \\
\frac{\partial y_6(w_1,w_2,w_3,x)}{\partial w_3} &= 2(y-\hat{y}) * [\sigma(w_2\sigma(w_1x))] \\
&= -2(128-0.38) = -255.24 \\
\frac{\partial y_6(w_1,w_2,w_3,x)}{\partial w_2} &= 2(y-\hat{y}) * w_3 * \sigma'(w_2\sigma(w_1x)) \\
&= -255.24 * 0.78 * [\frac{e^(-x)}{(1+e^(-x))^2}]_{x=-0.058} \\
&= -26.88 \\
\frac{\partial y_6(w_1,w_2,w_3,x)}{\partial w_1} &= 2(y-\hat{y}) * w_3 * \sigma'(w_2\sigma(w_1x)) * \sigma'(w_1x)x \qquad = -49.77 * (-0.11)[\frac{e^(-x)}{(1+e^(-x))^2}]_{x=-0.16}x \\
&= 0.86
\end{aligned}$$

By keeping all the others the same except the loss function and larger true value of y, the size of gradients for each weight is larger than previous question. Considering that the derivative of L2 loss is $-2(\hat{y}-y)$ which has steeper gradient than