Homework 3
600.482/682 Deep Learning
Spring 2019

March 14, 2019

Due 2019 Fri. 03/08 11:59pm.
Please submit a latex generated PDF
to Gradescope with entry code MYRR74

1. We have presented a matrix back propagation example in class (Lecture 6). In this exercise, we hope you to follow the same logic we used in class when concluding $\frac{\partial L}{\partial X} = W^T \frac{\partial L}{\partial Y}$.

   (a) Please use the same example and show your work by deriving $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} X^T$ (please show in detail).
   Ans:

$$W = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}, X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix},$$

$$Y = WX = \begin{pmatrix} w_{11}x_{11} + w_{12}x_{21} & w_{11}x_{12} + w_{12}x_{22} & w_{11}x_{13} + w_{12}x_{23} \\ w_{21}x_{11} + w_{22}x_{21} & w_{21}x_{12} + w_{22}x_{22} & w_{21}x_{13} + w_{22}x_{23} \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \end{pmatrix}$$

   By chain rule, we know that $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial W}$,

$$\frac{\partial L}{\partial Y} = \begin{pmatrix} \frac{\partial L}{\partial y_{11}} & \frac{\partial L}{\partial y_{12}} & \frac{\partial L}{\partial y_{13}} \\ \frac{\partial L}{\partial y_{21}} & \frac{\partial L}{\partial y_{22}} & \frac{\partial L}{\partial y_{23}} \end{pmatrix}, \frac{\partial Y}{\partial w_{11}} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ 0 & 0 & 0 \end{pmatrix}$$

   Therefore,

$$\frac{\partial L}{\partial x_{11}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial w_{11}} = \frac{\partial L}{\partial y_{11}}x_{11} + \frac{\partial L}{\partial y_{12}}x_{12} + \frac{\partial L}{\partial y_{13}}x_{13}$$

   Generalize for all $w_{11}, w_{12}, w_{21}, w_{22}$, we get:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial W} = \begin{pmatrix} \frac{\partial L}{\partial y_{11}}x_{11} + \frac{\partial L}{\partial y_{12}}x_{12} + \frac{\partial L}{\partial y_{13}}x_{13} & \frac{\partial L}{\partial y_{11}}x_{21} + \frac{\partial L}{\partial y_{12}}x_{22} + \frac{\partial L}{\partial y_{13}}x_{23} \\ \frac{\partial L}{\partial y_{21}}x_{11} + \frac{\partial L}{\partial y_{22}}x_{12} + \frac{\partial L}{\partial y_{23}}x_{13} & \frac{\partial L}{\partial y_{21}}x_{21} + \frac{\partial L}{\partial y_{22}}x_{22} + \frac{\partial L}{\partial y_{23}}x_{23} \end{pmatrix},$$

   Thus, we derive that $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} X^T$, where

$$X^T = \begin{pmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \\ x_{13} & x_{23} \end{pmatrix}$$

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

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

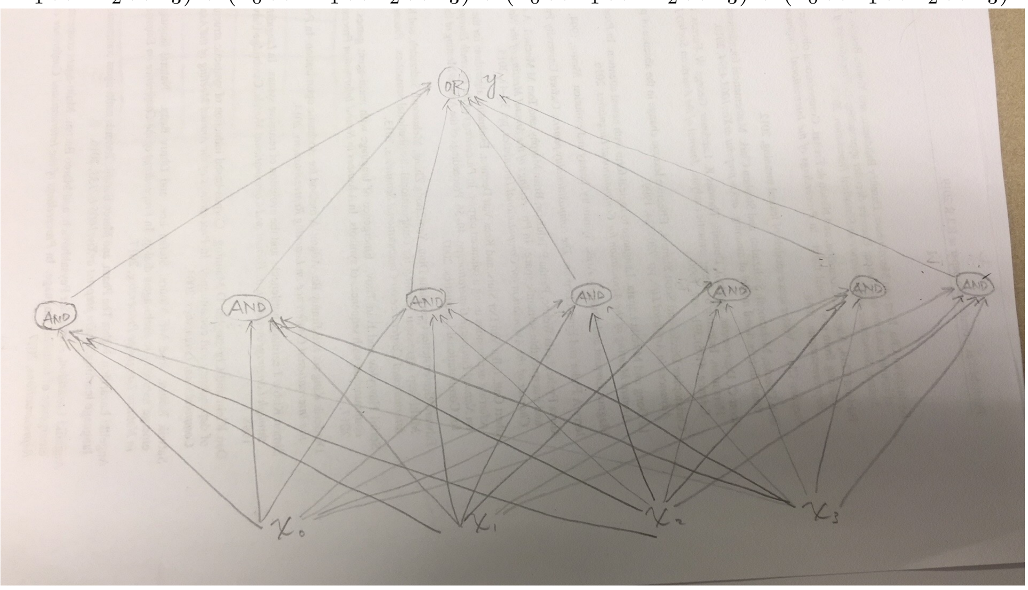
$$\hat{Y} = WX \begin{pmatrix} 1.5 & 1.1 \\ 1.2 & 0 \end{pmatrix},$$

Let $L_2 = ||y - \hat{y}||^2$, then we get $\frac{\partial L}{\partial y} = 2||y - \hat{y}||$. From above we derived that $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} X^T$. Therefore, the update of W would be:

$$W - 0.1 * \frac{\partial L}{\partial Y} X^T = \begin{pmatrix} 0.3 & 0.5 \\ -0.2 & 0.4 \end{pmatrix} - 0.1 * \begin{pmatrix} 3 & 0.2 \\ 0.4 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 0.26 & -0.42 \\ -0.2 & 0.28 \end{pmatrix}$$

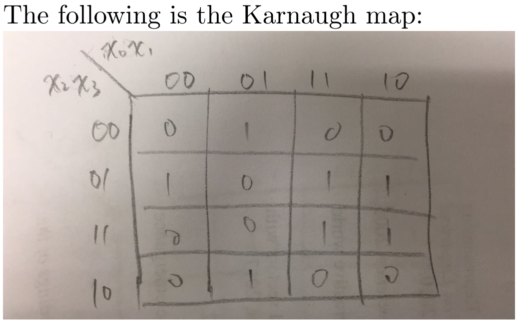
2. In this exercise, we will explore how we can represent arbitrary boolean functions using multi-layer perceptrons (MLP). Truth tables are a useful way to describe such boolean functions since they enumerate all possible combinations of variable values; however, they are not the most compact representation. The **disjunctive normal form** approach is a simple way to specify the boolean function by simply OR-ing together all the rows that have an output of "True"/1. Another approach is the **Karnaugh map**, which requires finding groups of adjacent "True"/1's in the table, with the conditions that the size of each group being a power of 2 and all "True"/1's belonging to some group (examples can be found here). This approach yields a more reduced form of the function compared to disjunctive normal form. All the following problems will be using the truth table defined in (Figure 1).

   (a) Express an equation for y in disjunctive normal form, and draw the corresponding MLP. How many total hidden units and edges are required?
   Ans:
   $y = (\neg x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_0 \wedge x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_0 \wedge x_1 \wedge x_2 \wedge \neg x_3) \vee (x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_0 \wedge \neg x_1 \wedge x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge \neg x_2 \wedge x_3) \vee (x_0 \wedge x_1 \wedge x_2 \wedge x_3)$



   There are 7 hidden units, 4*7=28 edges from input to hidden units, and 7 edges from hidden units to output gate, in total 35 edges required.

   (b) Using the Karnaugh map approach, find a reduced form for y, and draw the corresponding MLP. How many hidden units and edges are required?
   Ans:
   The following is the Karnaugh map:

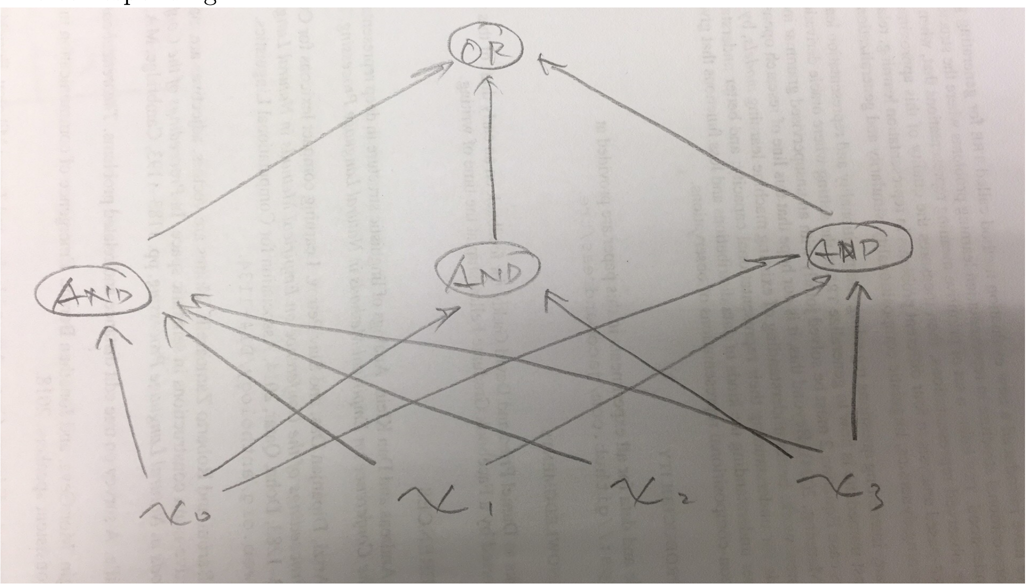


x_2x_3 \ x_0x_1	00	01	11	10
00	0	1	0	0
01	1	0	1	1
11	0	0	1	1
10	0	1	0	0

$\mathbf{x_0}$	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{y}$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Figure 1: Truth table for Problem 2

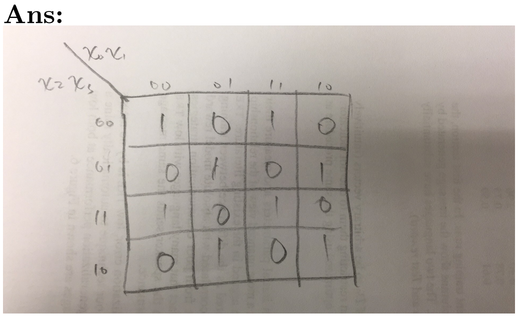
reduced form for y: $y = (\neg x_0 \wedge \neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_0 \wedge x_3) \vee (\neg x_0 \wedge x_1 \wedge \neg x_3)$
The corresponding MLP:



3 hidden units are required. 2+3+4 = 9 edges required from input to hidden units, and 3 more required from hidden units to output gate, therefore in total 12 edges required.

(c) What is the largest irreducable Karnaugh map in this case of 4 variables? Please derive this map. How many neurons in a DNF (one-hidden-layer) MLP for your derived Boolean function?

Ans:



This is the largest irreducable Karnaugh map since there can be no more grouping.

There would be 8 neurons in a DNF MLP for the derived Boolean function.

(d) What is the worst case bound on the number of hidden units and edges for n boolean variables $x_0, x_1, \ldots, x_{n-1}$? Use big-O notation.

Ans:

for n boolean variables, there would be $O(2^{n-1}) = O(2^n)$ hidden units, and $O((n+1)2^{n-1}) = O(n2^n)$ number of edges.

3. 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 sigmoid. what are the values of the extrema of this derivative, and when are they reached?

$$\frac{\partial \sigma}{\partial z} = \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right) = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1+e^{-z}-1}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} - \left(\frac{1}{1+e^{-z}}\right)^2 = \sigma - \sigma^2$$
$$= \sigma(1-\sigma)$$

   The values of this derivative range from $(0, 1/4]$. The maximum $1/4$ is reached when $z = 0$, making $\sigma = 1/2$. However, the derivative will approach to minimum 0 really fast, e.g. $z = \pm 10$, but will never reach.

   (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 do you notice about the magnitude?
   for each weight, the gradient should be:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w_i}$$

   Therefore we first consider:

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

   we first set

$$\begin{aligned} f_1 &= w_1 \cdot x = 0.1575, \\ f_2 &= w_2 \cdot \sigma(w_1 \cdot x) = w_2 \cdot \sigma(f_1) = -0.0593, \\ f_3 &= w_3 \cdot \sigma(w_2 \cdot \sigma(w_1 \cdot x)) = w_3 \cdot \sigma(f_2) = 0.3784 \end{aligned}$$

   Then for expanded w_1,

$$\frac{\partial \hat{y}}{\partial w_1} = \frac{\partial \sigma(f_3)}{\partial f_3} \cdot \frac{\partial f_3}{\partial f_2} \cdot \frac{\partial f_2}{\partial f_1} \cdot \frac{\partial f_1}{\partial w_1}$$

   For w_1, we get:

$$\begin{aligned} \frac{\partial \hat{y}}{\partial w_1} &= [\sigma(f_3) \cdot (1-\sigma(f_3))] \cdot [w_3 \cdot \sigma(f_2) \cdot (1-\sigma(f_2))] \cdot [w_2 \cdot \sigma(f_1) \cdot (1-\sigma(f_1))] \cdot x \\ &= [0.24126] \cdot [0.78 \cdot 0.24978] \cdot [-0.11 \cdot 0.2485] \cdot 0.63 = -0.00081 \end{aligned}$$

   For w_2, we get:

$$\begin{aligned} \frac{\partial \hat{y}}{\partial w_2} &= [\sigma(f_3) \cdot (1-\sigma(f_3))] \cdot [w_3 \cdot \sigma(f_2) \cdot (1-\sigma(f_2))] \cdot [\sigma(w_1 \cdot x)] \\ &= [0.24126] \cdot [0.78 \cdot 0.24978] \cdot [0.5393] = 0.02535 \end{aligned}$$

   For w_3, we get:

$$\begin{aligned} \frac{\partial \hat{y}}{\partial w_3} &= [\sigma(f_3) \cdot (1-\sigma(f_3))] \cdot [\sigma(w_2 \cdot \sigma(w_1 \cdot x))] \\ &= [0.24126] \cdot [0.4852] = 0.1171 \end{aligned}$$

   Then $\hat{y} = \sigma(f_3) = 0.5935$, and we get $\frac{\partial L}{\partial \hat{y}} = -\frac{1}{\hat{y}} + \frac{1-1}{1-\hat{y}} = -1.6849$.
   Thus, $\frac{\partial L}{\partial w_1} = 0.001364$, $\frac{\partial L}{\partial w_2} = -0.04271$, $\frac{\partial L}{\partial w_3} = -0.19723$.
   We noticed that the magnitudes are very small, and significantly decrease as we back-propagate earlier weights.

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 $\hat{y} = w_3 \cdot \sigma(w_2 \cdot \sigma(w_1 \cdot x))$, where predictions $\hat{y} \in \mathcal{R}$ and targets $y \in \mathcal{R}$. We will also 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 do you notice about the magnitude?

for each weight, the gradient should be:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w_i}$$

Therefore we first consider:

$$\frac{\partial L}{\partial \hat{y}} = -2(y - \hat{y})$$

we first set

$$\begin{aligned} f_1 = w_1 \cdot x = 0.1575, \\ f_2 = w_2 \cdot \sigma(w_1 \cdot x) = w_2 \cdot \sigma(f_1) = -0.0593, \\ f_3 = w_3 \cdot \sigma(w_2 \cdot \sigma(w_1 \cdot x)) = w_3 \cdot \sigma(f_2) = 0.3784 \end{aligned}$$

Then for expanded w_1,

$$\frac{\partial \hat{y}}{\partial w_1} = w_3 \cdot \frac{\partial \sigma(f_2)}{\partial f_2} \cdot \frac{\partial f_2}{\partial f_1} \cdot \frac{\partial f_1}{\partial w_1}$$

For w_1, we get:

$$\begin{aligned} \frac{\partial \hat{y}}{\partial w_1} &= [w_3 \cdot \sigma(f_2) \cdot (1 - \sigma(f_2))] \cdot [w_2 \cdot \sigma(f_1) \cdot (1 - \sigma(f_1))] \cdot x \\ &= [0.78 \cdot 0.24978] \cdot [-0.11 \cdot 0.2485] \cdot 0.63 = -0.003354 \end{aligned}$$

For w_2, we get:

$$\begin{aligned} \frac{\partial \hat{y}}{\partial w_2} &= [\sigma(f_3) \cdot (1 - \sigma(f_3))] \cdot [w_3 \cdot \sigma(f_2) \cdot (1 - \sigma(f_2))] \cdot [\sigma(w_1 \cdot x)] \\ &= [0.78 \cdot 0.24978] \cdot [0.5393] = 0.1051 \end{aligned}$$

For w_3, we get:

$$\begin{aligned} \frac{\partial \hat{y}}{\partial w_3} &= [\sigma(f_3) \cdot (1 - \sigma(f_3))] \cdot [\sigma(w_2 \cdot \sigma(w_1 \cdot x))] \\ &= [0.4852] \end{aligned}$$

Then $\hat{y} = f_3 = 0.3784$, and we get $\frac{\partial L}{\partial \hat{y}} = -2(y - \hat{y}) = -255.2431$.
Thus, $\frac{\partial L}{\partial w_1} = 0.8562$, $\frac{\partial L}{\partial w_2} = -26.8184$, $\frac{\partial L}{\partial w_3} = -123.8373$.
The magnitude still decrease a lot as backpropagate into earlier layers, but the magnitude are not smaller than 0 anymore.