CIS 419/519: Homework 5

Jiatong Sun

03/13/2020

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: $YuchenSun$, $JunfanPan$
https://stackoverflow.com/questions/50994504/how-to-put-figure-between-items-on-enumerate-list

1 Logical Functions with Neural Nets

a.

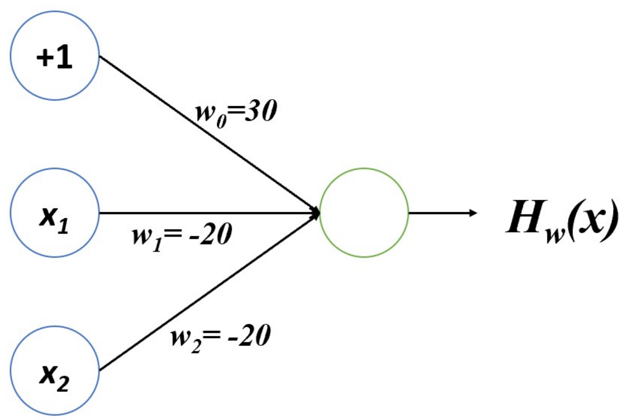

Figure 1: NAND

Truth Table		
x_1	x_2	$H(x)$
0	0	$\sigma(30) = 1$
0	1	$\sigma(10) = 1$
1	0	$\sigma(10) = 1$
1	1	$\sigma(-10) = 0$

Table 1: NAND

b.

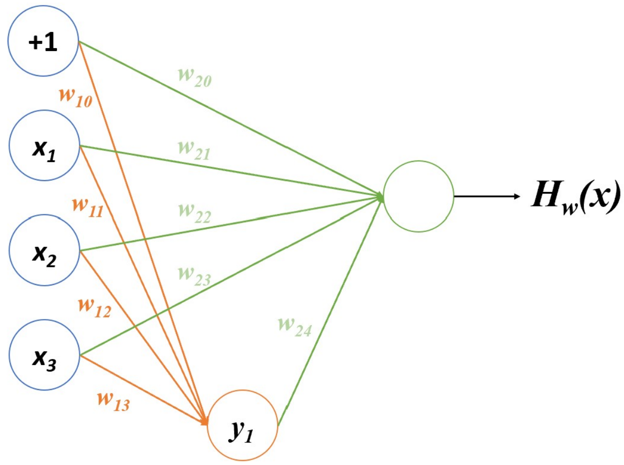

Figure 2: NAND

Truth Table								
w_{10}	w_{11}	w_{12}	w_{13}	w_{20}	w_{21}	w_{22}	w_{23}	w_{24}
-30	20	20	20	-10	20	20	20	-40

Table 2: Weight table

Truth Table				
x_1	x_2	x_3	y_1	$H(x)$
0	0	0	$\sigma(-30) = 0$	$\sigma(-10) = 0$
0	0	1	$\sigma(-10) = 0$	$\sigma(10) = 1$
0	1	0	$\sigma(-10) = 0$	$\sigma(10) = 1$
0	1	1	$\sigma(-10) = 0$	$\sigma(10) = 1$
1	0	0	$\sigma(10) = 1$	$\sigma(-10) = 0$
1	0	1	$\sigma(10) = 1$	$\sigma(-10) = 0$
1	1	0	$\sigma(10) = 1$	$\sigma(-10) = 0$
1	1	1	$\sigma(30) = 1$	$\sigma(10) = 1$

Table 3: Parity

2 Calculating Backprop by Hand

In the following solution, superscript denotes the number of layer and subscript denotes the number's position in its matrix.

$z(\boldsymbol{X})$ denotes the dot product of weights and last layer's output, and $a(\boldsymbol{X})$ denotes the output of current layer, which is after the activation function.

Since the hidden layer uses a sign function and the output layer uses a sigmoid function, the output can be expressed as follows:

$$a^2 = Sigmoid(z^2) = \frac{1}{1+e^{-z^2}} = \frac{1}{1+e^{-\boldsymbol{W^2}a^1}} = \frac{1}{1+e^{-\boldsymbol{W^2} \cdot Sign(\boldsymbol{W^1 X})}}$$

where

$$a^1 = Sign(z_1) = Sign(\boldsymbol{W^1 X}) = Sign(\begin{bmatrix} 0.1 & 0.2 \\ -0.4 & 0.3 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \end{bmatrix}) = Sign(\begin{bmatrix} 1.3 \\ -0.8 \end{bmatrix}) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

and

$$a^2 = \frac{1}{1 + e^{-\boldsymbol{W^2} a^1}} = \frac{1}{1 + e^{-\begin{bmatrix} 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}}} = \frac{1}{1 + e^{0.1}} = 0.475$$

To calculate the output gradient with respect to each of the weights, we can use the chain rule. Every part in the partial derivative chain is calculated and listed in:

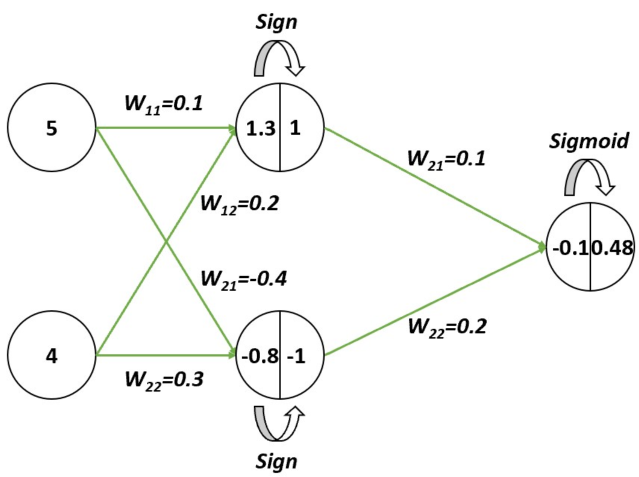

Figure 3: Forward Propagation

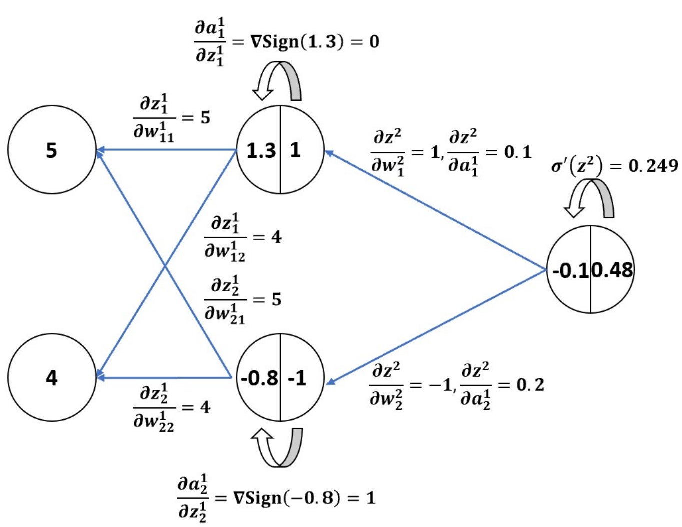

Figure 4: Backprpogation

$$\frac{da^2}{d\boldsymbol{W^2}}: \begin{cases} \frac{da^2}{\partial w^2_{21}} = \frac{\partial a^2}{\partial z^2}\frac{\partial z^2}{\partial w^2_1} = 0.249 \cdot 1 = 0.249 \\ \\ \frac{da^2}{\partial w^2_{21}} = \frac{\partial a^2}{\partial z^2}\frac{\partial z^2}{\partial w^2_2} = 0.249 \cdot (-1) = -0.249 \end{cases}$$

$$\frac{da^2}{d\boldsymbol{W^1}}: \begin{cases} \frac{da^2}{\partial w^1_{11}} = \frac{\partial a^2}{\partial z^2}\frac{\partial z^2}{\partial a^1_1}\frac{\partial a^1_1}{\partial z^1_1}\frac{\partial z^1_1}{\partial w^1_{11}} = 0.249 \cdot 0.1 \cdot 0 \cdot 5 = 0 \\ \\ \frac{da^2}{\partial w^1_{12}} = \frac{\partial a^2}{\partial z^2}\frac{\partial z^2}{\partial a^1_1}\frac{\partial a^1_1}{\partial z^1_1}\frac{\partial z^1_1}{\partial w^1_{12}} = 0.249 \cdot 0.1 \cdot 0 \cdot 4 = 0 \\ \\ \frac{da^2}{\partial w^1_{21}} = \frac{\partial a^2}{\partial z^2}\frac{\partial z^2}{\partial a^1_2}\frac{\partial a^1_2}{\partial z^1_2}\frac{\partial z^1_2}{\partial w^1_{21}} = 0.249 \cdot 0.2 \cdot 1 \cdot 5 = 0.249 \\ \\ \frac{da^2}{\partial w^1_{22}} = \frac{\partial a^2}{\partial z^2}\frac{\partial z^2}{\partial a^1_2}\frac{\partial a^1_2}{\partial z^1_2}\frac{\partial z^1_2}{\partial w^1_{22}} = 0.249 \cdot 0.2 \cdot 1 \cdot 4 = 0.199 \end{cases}$$

So we finally get

$$\frac{da^2}{d\boldsymbol{W^2}} = \begin{bmatrix} 0.249 \\ -0.249 \end{bmatrix}, \quad \frac{da^2}{d\boldsymbol{W^1}} = \begin{bmatrix} 0 & 0 \\ 0.249 & 0.199 \end{bmatrix}$$

3 Neural Nets in SuperTuxKart