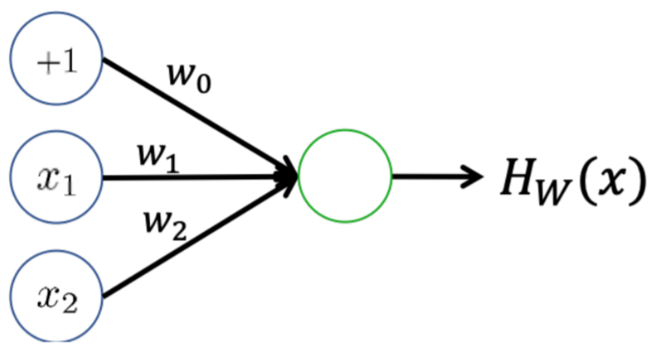
CIS 419/519: Homework 5

{Zhuheng Jiang}

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: {Yongxin Guo, https://medium.com/towards-artificial-intelligence/image-classification-using-deep-learning-pytorch-a-case-study-with-flower-image-data-80a18554df63,
https://stackoverflow.com/questions/55810665/changing-input-dimension-for-alexnet,
https://www.cnblogs.com/charlotte77/p/5629865.html }

1 Logic functions with neural nets

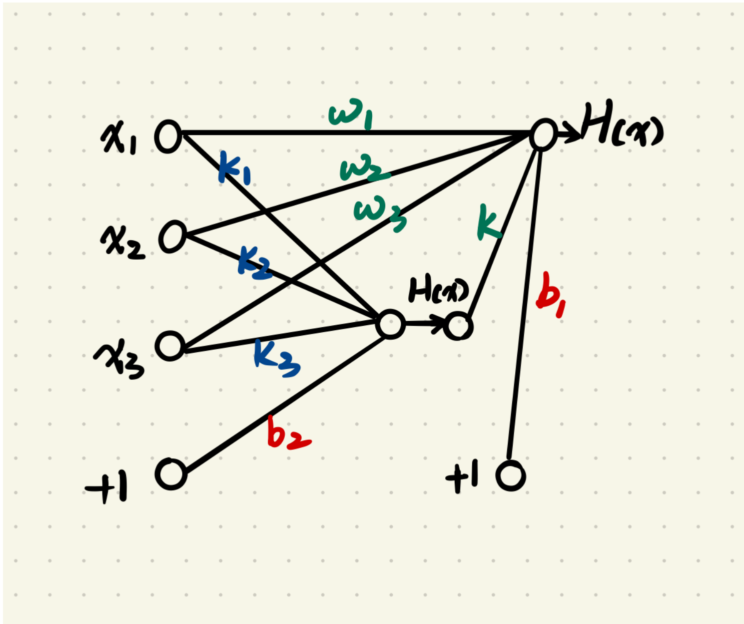
a. The NAND of two binary inputs:
The network is the same with the original network. And the weights are: $\omega_0 = 30, \omega_1 = -20, \omega_2 = -20$.
The truth table is shown below:



Truth Table

x_0	x_1	H(x)
0	0	$\sigma(30) = 1$
0	1	$\sigma(10) = 1$
1	0	$\sigma(10) = 1$
1	1	$\sigma(-10) = 0$

b. The parity of three binary inputs:
The result comes from previous researches and the reference is attached. The neural network is draw below:



And there are many sets of values for weights. I set them to be: $\omega_0 = 30, \omega_1 = \omega_2 = \omega_3 = 1$, $k_1 = k_2 = k_3 = 1$, $k = -2$, $b_1 = -0.5$, $b_2 = -1.5$.

Truth Table

x_1	x_2	x_3	$H(x)$
0	0	0	$\sigma(-0.5) = 0$
0	0	1	$\sigma(0.5) = 1$
0	1	0	$\sigma(0.5) = 1$
1	0	0	$\sigma(0.5) = 1$
1	0	1	$\sigma(-0.5) = 0$
1	1	0	$\sigma(-0.5) = 0$
0	1	1	$\sigma(-0.5) = 0$
1	1	1	$\sigma(0.5) = 1$

Reference: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.5442rep=rep1type=pdf

2 Calculating Backprop by Hand

Firstly implement the forward pass:

$$input_h = W^1 \bullet \mathbf{x} = [1.3, -0.8]^T$$
$$output_h = \text{sign}(\text{ input}_h) = [1, -1]^T$$
$$input_o = W^2 \bullet output_h = -0.1$$
$$output_o = sigmoid(input_o) = 0.4750$$

Then implement the backprop, take W^2 as example, apply the chain rule, w^2 is the element of the W^2:

$$\frac{dL}{dw^2} = \frac{dL}{doutput_o} \frac{doutput_o}{dinput_o} \frac{dinput_o}{dw^2} \tag{1}$$

The loss function and accurate labels are not given, so the first term of the chain cannot be worked out.

$$\frac{dL}{dw^2} = \frac{dL}{doutput_o}[output_o(1 - output_o)](output_h) \tag{2}$$

Since the result of sigmoid function is treated as the loss, so **the gradient with respect to** W^2 is:

$$\frac{dL}{dW^2} = [0.2493, -0.2493] \tag{3}$$

Then the backprop between hidden layer and input layer is computed based on the previous calculations.

$$\frac{dL}{dW^1} = \frac{df_h}{dW^1}\frac{dL}{dW^2} = \frac{dL}{dW^2} * \frac{doutput_h}{dinput_h} * \frac{dinput_h}{dW^1} \tag{4}$$

Where:

$$\frac{doutput_h}{dinput_h} = \nabla sign(output_h) = \nabla sign([1, -1]^T) = [1, 1]^T \tag{5}$$

$$\frac{dinput_h}{dW^1} = \mathbf{x} = \begin{bmatrix} 5 & 4 \\ 5 & 4 \end{bmatrix} \tag{6}$$

After combining and transforming terms, the backprop gradient for W_1 is:

$$\frac{dL}{dW^1} = \begin{bmatrix} 1.2465 & 0.9972 \\ -1.2465 & -0.9972 \end{bmatrix} \tag{7}$$

Here, some assumptions were made for convenience. If the labels or the loss are given, then we can replace the terms with specific numbers.