# CIS 419/519: Homework 5

{Yongxin Guo}

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: Jiangzhu Heng, Yihang Xu

## 1 Problem Sets:

### 1.1 Logical Function with Neural Nets

**(a). The NAND of two binary inputs:**

Neural Network Diagram:

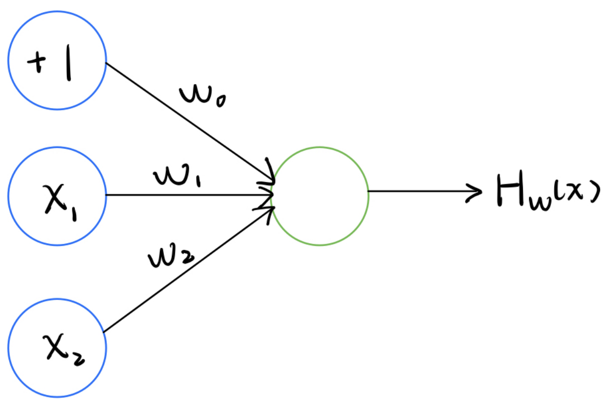


Figure 1: Neural Network Representation of Logical NAND Function

And the weights are tabulated in Table 1:

| | $w_0$ | $w_1$ | $w_2$ |
|---|---|---|---|
| Weight Values | 30 | -20 | -20 |

Table 1: Weight Values for Neural Network Representation of NAND Logical Function

Truth Table:

| $x_1$ | $x_2$ | Logical Function Output | Neural Network Output |
|---|---|---|---|
| 0 | 0 | 1 | $\sigma(30) = 1$ |
| 0 | 1 | 1 | $\sigma(10) = 1$ |
| 1 | 0 | 1 | $\sigma(10) = 1$ |
| 1 | 1 | 0 | $\sigma(-10) = 0$ |

Table 2: Truth Table of NAND Logical Function and Equivalent Neural Network Outputs

**(b). The parity of three binary inputs:**

Neural Network Diagram:

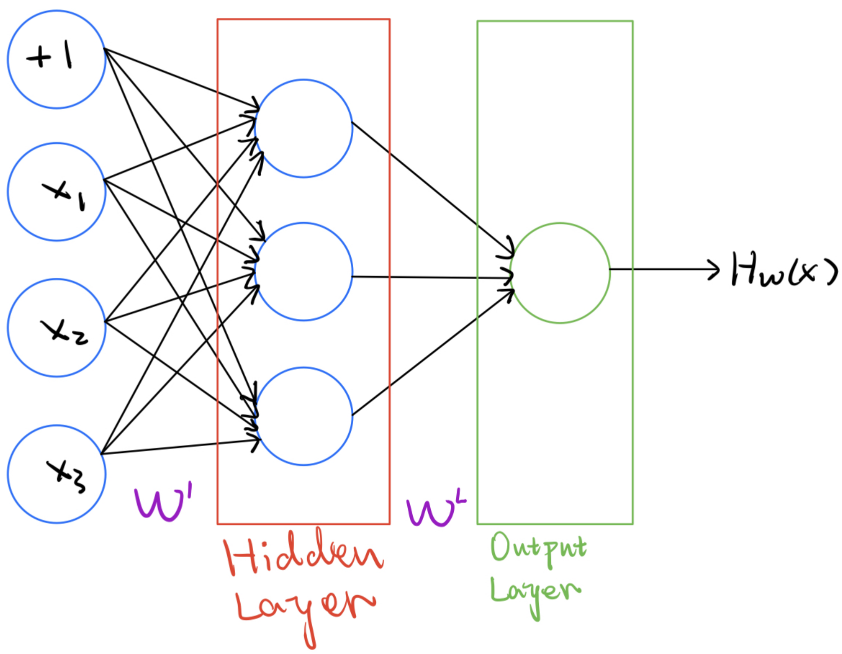


Figure 2: Neural Network Representation of Parity of Three Inputs

And the weights are tabulated in Table 3

| | $\mathbf{W^1_{ij}}$ | $\mathbf{W^L_{ij}}$ |
|---|---|---|
| Weight Values | $\begin{bmatrix} -20 & 30 & 30 & 30 \\ 40 & -30 & -30 & -30 \\ 80 & -30 & -30 & -30 \end{bmatrix}$ | $\begin{bmatrix} 30 & 50 & -60 \end{bmatrix}$ |

Table 3: Weight Values for Neural Network Representation of the Parity of Three Inputs

**Notation:** $W_{ij}$ denotes the weight between $j^{th}$ component of the input vector (the left layer) and the $i_{th}$ neuron of the next layer to the right.
Truth Table:

| $x_1$ | $x_2$ | $x_3$ | Logical Function Output | Neuron 1 | Neuron 2 | Neuron 3 | Neural Network Output |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $\sigma(-20) = 0$ | $\sigma(40) = 1$ | $\sigma(80) = 1$ | $\sigma(-10) = 0$ |
| 1 | 0 | 0 | 1 | $\sigma(10) = 1$ | $\sigma(10) = 1$ | $\sigma(50) = 1$ | $\sigma(20) = 1$ |
| 0 | 1 | 0 | 1 | $\sigma(10) = 1$ | $\sigma(10) = 1$ | $\sigma(50) = 1$ | $\sigma(20) = 1$ |
| 0 | 0 | 1 | 1 | $\sigma(10) = 1$ | $\sigma(10) = 1$ | $\sigma(50) = 1$ | $\sigma(20) = 1$ |
| 1 | 1 | 0 | 0 | $\sigma(40) = 1$ | $\sigma(-20) = 0$ | $\sigma(20) = 1$ | $\sigma(-30) = 0$ |
| 0 | 1 | 1 | 0 | $\sigma(40) = 1$ | $\sigma(-20) = 0$ | $\sigma(20) = 1$ | $\sigma(-30) = 0$ |
| 1 | 0 | 1 | 0 | $\sigma(40) = 1$ | $\sigma(-20) = 0$ | $\sigma(20) = 1$ | $\sigma(-30) = 0$ |
| 1 | 1 | 1 | 1 | $\sigma(70) = 1$ | $\sigma(-50) = 0$ | $\sigma(-10) = 0$ | $\sigma(30) = 1$ |

Table 4: Truth Table of Parity Function of Three Inputs and Equivalent Neural Network Outputs

### 1.2 Calculating Backprop by Hand

Let

$$a_0 = \mathbf{x} = [5, 4]^T \tag{1}$$

$$a_1 = f_1(\mathbf{W^1} a_0) = Sign(\mathbf{W^1} a_0) = Sign([1.3, -0.8]^T) = [1, -1]^T \tag{2}$$

$$H_W(\mathbf{x}) = a_2 = f_2(\mathbf{W^2} a_1) = \sigma(\mathbf{W^2} a_1) \tag{3}$$

And,

$$\mathbf{W^2} = [0.1, 0.2]^T \quad a_1 = [1, -1]^T \tag{4}$$

So,

$$H_W(x) = \sigma(w_1^2 - w_2^2) = \frac{1}{1 + e^{w_1^2 - w_2^2}} \tag{5}$$

Here, the gradient is implemented on the output of the neural network so the loss function $L(H_W(x)) = H_W(x)$. Then the gradient of the output w.r.t $\mathbf{W^2}$ is,

$$\frac{dH_W(x)}{d\mathbf{W^2}} = \frac{df_2(\mathbf{W^2} a_1)}{d\mathbf{W^2}} = \frac{d}{d\mathbf{W^2}}(\frac{1}{1 + e^{w_1^2 - w_2^2}}) = [\frac{d(1 + e^{w_1^2 - w_2^2})^{-1}}{dW_1^2}, \frac{d(1 + e^{w_1^2 - w_2^2})^{-1}}{dW_2^2}] \tag{6}$$

Which has to be evaluated at $\mathbf{W^2} = [0.1, 0.2]^T$. Thus,

$$\frac{dH_W(x)}{d\mathbf{W^2}} = [\frac{e^{0.1}}{(1 + e^{0.1})^2}, \frac{-e^{0.1}}{(1 + e^{0.1})^2}] = [\mathbf{0.2494}, -\mathbf{0.2494}] \tag{7}$$

Next, the gradient of the output w.r.t $\mathbf{W^1}$ is,

$$\frac{dH_W(x)}{d\mathbf{W^1}} = \frac{df_2(\mathbf{W^2} a_1)}{da_1} \times \frac{df_1(\mathbf{W^1} a_0)}{d\mathbf{W^1}} \tag{8}$$

And,

$$\frac{df_2(\mathbf{W^2} a_1)}{da_1} = \frac{df_2(\mathbf{W^2} a_1)}{d(\mathbf{W^2} a_1)} \times \frac{d(\mathbf{W^2} a_1)}{da_1} \tag{9}$$

With,

$$\frac{df_2(\mathbf{W^2} a_1)}{d(\mathbf{W^2} a_1)} = \frac{d}{d(\mathbf{W^2} a_1)}(\frac{1}{1 + e^{-\mathbf{W^2} a_1}}) = \frac{e^{-\mathbf{W^2} a_1}}{(1 + e^{-\mathbf{W^2} a_1})^2} = \frac{e^{0.1}}{(1 + e^{0.1})^2} = 0.2494 \tag{10}$$

Which was evaluated at $\mathbf{W^2} = [0.1, 0.2]^T$ and $a_1 = [1, -1]^T$.

And,

$$\frac{d(\mathbf{W^2} a_1)}{da_1} = \mathbf{W^2} = [0.1, 0.2] \tag{11}$$

So, Plugging the results of Eqn (11) and Eqn (10) into Eqn (9). One can get,

$$\frac{df_2(\mathbf{W^2} a_1)}{da_1} = 0.2494 \times [0.1, 0.2] = [0.02494, 0.04988] \tag{12}$$

Furthermore,

$$\frac{df_1(\mathbf{W^1} a_0)}{d\mathbf{W^1}} = \frac{df_1(\mathbf{W^1} a_0)}{d(\mathbf{W^1} a_0)} \times \frac{d(\mathbf{W^1} a_0)}{d\mathbf{W^1}} = \nabla Sign(\mathbf{W^1} a_0) \times \frac{d(\mathbf{W^1} a_0)}{d\mathbf{W^1}} \tag{13}$$

Since $\mathbf{W^1} a_0$ is a vector and $\mathbf{W^1}$ is a matrix, thus the derivative of a vector w.r.t a matrix is just the derivative of each element of the vector w.r.t each element of the matrix, which results in a tensor as shown in the following

$$\nabla Sign(\mathbf{W^1} a_0) \times \frac{d(\mathbf{W^1} a_0)}{d\mathbf{W^1}} = [\nabla Sign(1.3) \times \frac{d(5W_{11}^1 + 4W_{12}^1)}{d\mathbf{W^1}}, \nabla Sign(-0.8) \times \frac{d(5W_{21}^1 + 4W_{22}^1)}{d\mathbf{W^1}}]^T \tag{14}$$

With $\nabla Sign(1.3) = 0$ and $\nabla Sign(-0.8) = 1$

Thus, plugging everything into Eqn (8), one can get,

$$\frac{dH_W(x)}{d\mathbf{W^1}} = [0.02494, 0.04988][\mathbf{0}_{2\times 2}, \mathbf{g}_{2\times 2}]^T = 0.04988\mathbf{g}_{2\times 2} \tag{15}$$

with $\mathbf{g}_{2\times 2} = \begin{bmatrix} 0 & 0 \\ 5 & 4 \end{bmatrix}$. Thus, the gradient of the output w.r.t $\mathbf{W^1}$ is,

$$\frac{dH_W(x)}{d\mathbf{W^1}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0.2494} & \mathbf{0.1995} \end{bmatrix} \tag{16}$$