Homework 6 - Written Answer Key

Question 1:

(Question)

Given a neural network with 3 layers, where the *input* layer has 3 neurons, the *hidden* layer has 3 neurons, the *output* layer has 2 neurons, and

$$W^{(1)} = \begin{pmatrix} 4 & 4 & 1 \\ -4 & -4 & 1 \\ 1 & 1 & -5 \end{pmatrix}, \; W^{(2)} = \begin{pmatrix} 4 & 4 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \; b^{(1)} = \begin{pmatrix} -2 \\ 6 \\ 1 \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} -6 \\ 2 \end{pmatrix}$$

(a) Draw the neural network

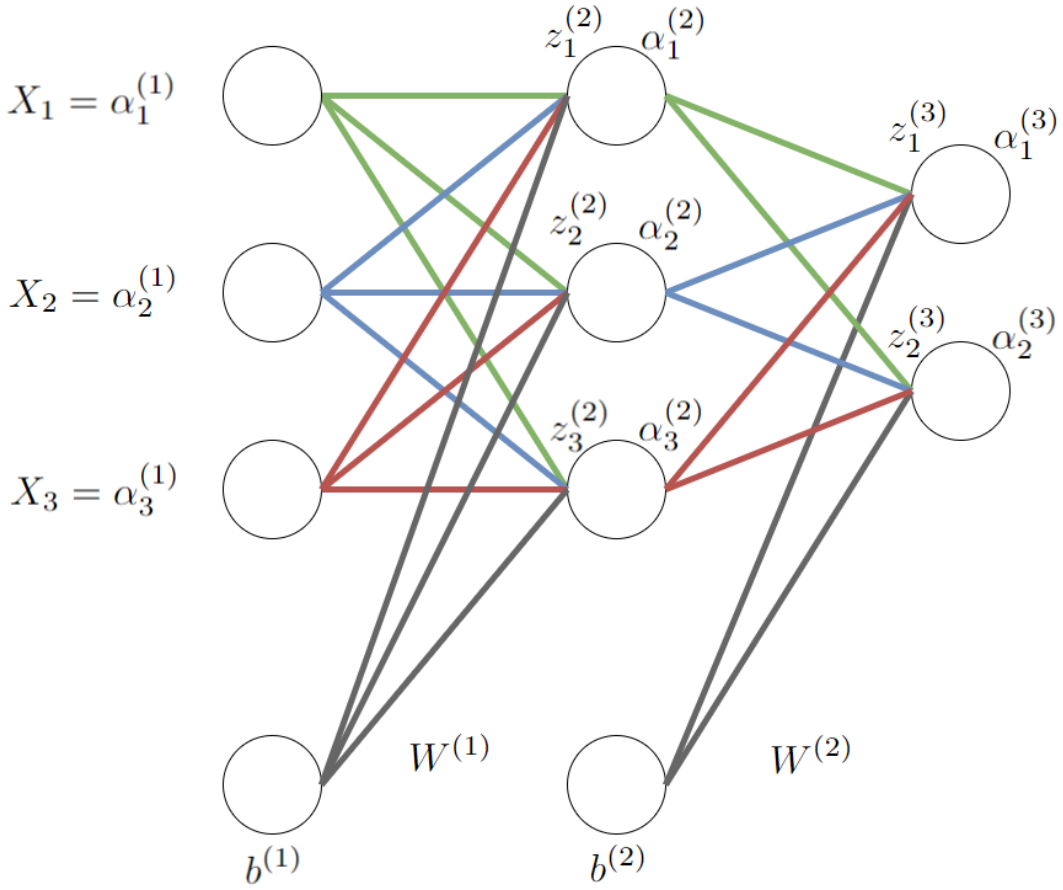
(b) What is $h_{W,b}(\mathbf{x})$ when $\mathbf{x}^T = (1, 2, 3)$

(c) For the example $\mathbf{x}^T = (1, 2, 3)$ and $\mathbf{y}^T = (1, 0)$

- What is $\delta_1^{(3)}$
- What is $\frac{\partial J(W,b;\mathbf{x},y)}{\partial W_{11}^{(2)}}$
- What is $\delta_1^{(2)}$
- What is $\frac{\partial J(W,b;\mathbf{x},y)}{\partial W_{11}^{(1)}}$

(Answer(s))

1.



2.

$$a_1^{(1)} = X_1,\ a_2^{(1)} = X_2,\ a_3^{(1)} = X_3$$

First look at the hidden layer:

$$z_i^{(2)} = (\sum_{j=1}^{3} W_{ij}^{(1)} \alpha_j^{(1)}) + b_j^{(1)}$$

$$\begin{aligned}
z_1^{(2)} &= W_{11}^{(1)}\alpha_1^{(1)} + W_{12}^{(1)}\alpha_2^{(1)} + W_{13}^{(1)}\alpha_3^{(1)} + b_1^{(1)} \\
&= (4)(1) + (4)(2) + (1)(3) + (-2) = 13 \\
z_2^{(2)} &= W_{21}^{(1)}\alpha_1^{(1)} + W_{22}^{(1)}\alpha_2^{(1)} + W_{23}^{(1)}\alpha_3^{(1)} + b_2^{(1)} \\
&= (-4)(1) + (-4)(2) + (1)(3) + (6) = -3 \\
z_3^{(2)} &= W_{31}^{(1)}\alpha_1^{(1)} + W_{32}^{(1)}\alpha_2^{(1)} + W_{33}^{(1)}\alpha_3^{(1)} + b_3^{(1)} \\
&= (1)(1) + (1)(2) + (-5)(3) + (1) = -11
\end{aligned}$$

Then apply the activation function (sigmoid: $f(z) = \frac{1}{1+e^{-z}}$)

$$a_1^{(2)} = f(13) = 0.999998$$

$$a_2^{(2)} = f(-3) = 0.047426$$

$$a_3^{(2)} = f(-11) = 0.000017$$

Now look at the output layer:

$$z_i^{(3)} = (\sum_{j=1}^{3} W_{ij}^{(2)} \alpha_j^{(2)}) + b_j^{(2)}$$

$$z_1^{(3)} = W_{11}^{(2)}\alpha_1^{(2)} + W_{12}^{(2)}\alpha_2^{(2)} + W_{13}^{(2)}\alpha_3^{(2)} + b_1^{(2)}$$
$$= (4)(0.999998) + (4)(0.047426) + (1)(0.000017) + (-6) = -1.810287$$
$$z_2^{(3)} = W_{21}^{(2)}\alpha_1^{(2)} + W_{22}^{(2)}\alpha_2^{(2)} + W_{23}^{(2)}\alpha_3^{(2)} + b_2^{(2)}$$
$$= (1)(0.999998) + (1)(0.047426) + (2)(0.000017) + (2) = 3.047458$$

Lastly, apply the activation function to the final layer

$$a_1^{(3)} = f(-1.810287) = 0.140603$$

$$a_2^{(3)} = f(3.047458) = 0.954673$$

$$h_{W,b}(\mathbf{x}) = \begin{bmatrix} 0.140603 \\ 0.954673 \end{bmatrix}$$

3.

$$f'(z) = f(z)(1 - f(z))$$

The neuron we're looking at is in the output layer, so:

$$\delta_j^{(n_\ell)} = \frac{\partial J}{\partial z_j^{(n_\ell)}} = (f(z_j^{(n_\ell)}) - y)(f'(z_j^{(n_\ell)}))$$

$$\delta_1^{(3)} = ((0.140603) - (1))((0.140603)(1 - (0.140603))) = -0.103844$$

$$\frac{\partial J(W, b; \mathbf{x}, y)}{\partial W_{11}^{(2)}} = \delta_1^{(3)}(\frac{\partial z_1^{(3)}}{W_{11}^{(2)}}) = \delta_1^{(3)}\alpha_1^{(2)}$$

$$\frac{\partial J(W, b; \mathbf{x}, y)}{\partial W_{11}^{(2)}} = (-0.103844)(0.999998) = -0.103844$$

The neuron we're looking at now is in a hidden layer, so:

$$\delta_j^{(\ell)} = \frac{\partial J}{\partial z_j^{(\ell)}} = \sum_{i=1}^{s_{\ell+1}} \delta_i^{(\ell+1)} W_{ij}^{(\ell)} f'(z_j^{(\ell)})$$

$$\delta_1^{(2)} = \delta_1^{(3)} W_{11}^{(2)} f'(z_1^{(2)}) + \delta_2^{(3)} W_{21}^{(2)} f'(z_1^{(2)})$$

$$\delta_2^{(3)} = ((0.954673) - (0))((0.954673)(1 - (0.954673))) = 0.041311$$

$$f'(z_1^{(2)}) = (f(z_1^{(2)}))(1 - f(z_1^{(2)})) = \alpha_1^{(2)}(1 - \alpha_1^{(2)})$$

$$f'(z_1^{(2)}) = (0.999998)(1 - 0.999998) = 0.000002$$

$$\delta_1^{(2)} = (-0.103844)(4)(0.000002) + (0.041311)(1)(0.000002) = -0.000001$$

$$\frac{\partial J(W, b; \mathbf{x}, y)}{\partial W_{11}^{(1)}} = \delta_1^{(2)} \alpha_1^{(1)}$$

$$\frac{\partial J(W, b; \mathbf{x}, y)}{\partial W_{11}^{(1)}} = (-0.000001)(1) = -0.000001$$

Question 2:

(Question)

Given a neural network with 3 layers, where the *input* layer has 2 neurons, the *hidden* layer has 2 neurons, the *output* layer has 1 neuron, and

$$W^{(1)} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \; W^{(2)} = \begin{pmatrix} 1 & 2 \end{pmatrix}, \; b^{(1)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 1 \end{pmatrix}$$

Suppose you had the following training set: $((1,0),1),((0,1),0)$. Perform 1 step of gradient descent where the learning rate is 0.2.

(Answer(s))

1.

$$f(z) = \frac{1}{1+e^{-z}}$$
$$f'(z) = f(z)(1-(f(z)))$$

For the point ((1, 0), 1):
Forward Propogation:

$$\alpha^{(1)} = X = \begin{bmatrix}1\\0\end{bmatrix}$$

$$z^{(2)} = W^{(1)}a^{(1)} + b^{(1)}$$
$$z^{(2)} = \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}1\\-1\end{bmatrix} = \begin{bmatrix}2\\2\end{bmatrix}$$

$$\alpha^{(2)} = f(z^{(2)})$$
$$\alpha^{(2)} = \begin{bmatrix}\frac{1}{1+e^{-2}}\\ \frac{1}{1+e^{-2}}\end{bmatrix} = \begin{bmatrix}0.880797\\0.880797\end{bmatrix}$$

$$z^{(3)} = W^{(2)}\alpha^{(2)} + b^{(2)}$$
$$z^{(3)} = \begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}0.880797\\0.880797\end{bmatrix} + 1 = 3.642391$$

$$\alpha^{(3)} = f(z^{(3)})$$
$$\alpha^{(3)} = \frac{1}{1+e^{-3.642391}} = 0.974479$$

2.

Back Propagation:

$$\delta^{(3)} = (f(z^{(3)}) - y)(f'(z^{(3)}))$$
$$\delta^{(3)} = ((0.974479) - (1))((0.974479)(1-(0.974479))) = -0.000635$$

$$\delta^{(2)} = (W^{(2)})^T\delta^{(3)} \cdot f'(z^{(2)})$$
$$\delta^{(2)} = \begin{bmatrix}1\\2\end{bmatrix}(-0.000635) \cdot \begin{bmatrix}((0.880797)(1-(0.880797)))\\((0.880797)(1-(0.880797)))\end{bmatrix} = \begin{bmatrix}-0.000067\\-0.000133\end{bmatrix}$$

$$\delta^{(1)} = (W^{(1)})^T\delta^{(2)} \cdot f'(z^{(1)})$$
$$\delta^{(1)} = \begin{bmatrix}1 & 3\\2 & 4\end{bmatrix}\begin{bmatrix}-0.000067\\-0.000133\end{bmatrix} \cdot \begin{bmatrix}((1)(1-(1)))\\((0)(1-(0)))\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

3.

Partial Derivatives:

$$\frac{\partial J(W,b;(1,0),1)}{\partial W^{(2)}} = \delta^{(3)}(\alpha^{(2)})^T$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial W^{(2)}} = (-0.000635)\begin{bmatrix}0.880797 & 0.880797\end{bmatrix} = \begin{bmatrix}-0.000559 & -0.000559\end{bmatrix}$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial W^{(1)}} = \delta^{(2)}(\alpha^{(1)})^T$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial W^{(1)}} = \begin{bmatrix}-0.000067\\-0.000133\end{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix} = \begin{bmatrix}-0.000067 & 0\\-0.000133 & 0\end{bmatrix}$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial b^{(2)}} = \delta^{(2)}$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial b^{(2)}} = \begin{bmatrix}-0.000067\\-0.000133\end{bmatrix}$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial b^{(1)}} = \delta^{(3)}$$

$$\frac{\partial J(W,b;(1,0),1)}{\partial b^{(1)}} = \begin{bmatrix}-0.000635\end{bmatrix}$$

4.

For the point ((0, 1), 0):
Forward Propagation:

$$\alpha^{(1)} = X = \begin{bmatrix}0\\1\end{bmatrix}$$

$$z^{(2)} = \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}1\\-1\end{bmatrix} = \begin{bmatrix}3\\3\end{bmatrix}$$

$$\alpha^{(2)} = \begin{bmatrix}\frac{1}{1+e^{-3}}\\\frac{1}{1+e^{-3}}\end{bmatrix} = \begin{bmatrix}0.952574\\0.952574\end{bmatrix}$$

$$z^{(3)} = \begin{bmatrix}1 & 2\end{bmatrix}\begin{bmatrix}0.952574\\0.952574\end{bmatrix} + 1 = 3.857722$$

$$\alpha^{(3)} = \frac{1}{1+e^{-3.857722}} = 0.979321$$

5.

Back Propagation:

$$\delta^{(3)} = ((0.979321) - (0))((0.979321)(1 - (0.979321))) = 0.019833$$

$$\delta^{(2)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} (0.019833) \cdot \begin{bmatrix} ((0.952574)(1 - (0.952574))) \\ ((0.952574)(1 - (0.952574))) \end{bmatrix} = \begin{bmatrix} 0.000896 \\ 0.001792 \end{bmatrix}$$

$$\delta^{(1)} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0.000896 \\ 0.001792 \end{bmatrix} \cdot \begin{bmatrix} ((0)(1 - (0))) \\ ((1)(1 - (1))) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

6.

Partial Derivatives:

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial W^{(2)}} = (0.019833) \begin{bmatrix} 0.952574 & 0.952574 \end{bmatrix} = \begin{bmatrix} 0.018892 & 0.018892 \end{bmatrix}$$

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial W^{(1)}} = \begin{bmatrix} 0.000896 \\ 0.001792 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0.000896 \\ 0 & 0.001792 \end{bmatrix}$$

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial b^{(2)}} = \begin{bmatrix} 0.000896 \\ 0.001792 \end{bmatrix}$$

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial b^{(1)}} = \begin{bmatrix} 0.019833 \end{bmatrix}$$

7.

Now, find the gradients:

$$\Delta W^{(1)} = (\delta^{(2)}(\alpha^{(1)})^T)_{((1,0),1)} + (\delta^{(2)}(\alpha^{(1)})^T)_{((0,1),0)}$$

$$\Delta W^{(1)} = \begin{bmatrix} -0.000067 & 0 \\ -0.000133 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0.000896 \\ 0 & 0.001792 \end{bmatrix} = \begin{bmatrix} -0.000067 & 0.000896 \\ -0.000133 & 0.001792 \end{bmatrix}$$

$$\Delta W^{(2)} = (\delta^{(3)}(\alpha^{(2)})^T)_{((1,0),1)} + (\delta^{(3)}(\alpha^{(2)})^T)_{((0,1),0)}$$

$$\Delta W^{(2)} = \begin{bmatrix} -0.000559 & -0.000559 \end{bmatrix} + \begin{bmatrix} 0.018892 & 0.018892 \end{bmatrix} = \begin{bmatrix} 0.018333 & 0.018333 \end{bmatrix}$$

$$\Delta b^{(1)} = (\delta^{(2)})_{((1,0),1)} + (\delta^{(2)})_{((0,1),0)}$$

$$\Delta b^{(1)} = \begin{bmatrix} 0.000896 \\ 0.001792 \end{bmatrix} + \begin{bmatrix} -0.000067 \\ -0.000133 \end{bmatrix} = \begin{bmatrix} 0.000829 \\ 0.001658 \end{bmatrix}$$

$$\Delta b^{(2)} = (\delta^{(3)})_{((1,0),1)} + (\delta^{(3)})_{((0,1),0)}$$

$$\Delta b^{(2)} = \begin{bmatrix} -0.000635 \end{bmatrix} + \begin{bmatrix} 0.019833 \end{bmatrix} = \begin{bmatrix} 0.019198 \end{bmatrix}$$

8.

And now update the parameters:

$$W^{(1)} = W^{(1)} - \frac{\alpha}{2}(\Delta W^{(1)})$$

$$W^{(1)} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \frac{0.2}{2}\begin{bmatrix} -0.000067 & 0.000896 \\ -0.000133 & 0.001792 \end{bmatrix} = \begin{bmatrix} 1.000007 & 1.999910 \\ 3.000013 & 3.999821 \end{bmatrix}$$

$$W^{(2)} = W^{(2)} - \frac{\alpha}{2}(\Delta W^{(2)})$$

$$W^{(2)} = \begin{bmatrix} 1 & 2 \end{bmatrix} - \frac{0.2}{2}\begin{bmatrix} 0.018333 & 0.018333 \end{bmatrix} = \begin{bmatrix} 0.998167 & 1.998167 \end{bmatrix}$$

$$b^{(1)} = b^{(1)} - \frac{\alpha}{2}(\Delta b^{(1)})$$

$$b^{(1)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} - \frac{0.2}{2}\begin{bmatrix} 0.000829 \\ 0.001658 \end{bmatrix} = \begin{bmatrix} 0.999917 \\ -1.000166 \end{bmatrix}$$

$$b^{(2)} = b^{(2)} - \frac{\alpha}{2}(\Delta b^{(2)})$$

$$b^{(2)} = \begin{bmatrix} 1 \end{bmatrix} - \frac{0.2}{2}\begin{bmatrix} 0.019198 \end{bmatrix} = \begin{bmatrix} 0.998080 \end{bmatrix}$$

Question 3:

(Question)

(Do not turn in) Repeat the previous problem, but now use a RELU activation function for layer 2 instead of a sigmoid activation function for layer 2 of the network. See question 7 below.

(Answer(s))

1.

$$f(z)\frac{1}{1+e^{-z}}$$

$$f'(z) = f(z)(1-(f(z)))$$

$$f^{(2)}(z) = \max(0, z)$$

$$f^{(2)\prime}(z) = \begin{cases} 0, & \text{If } z < 0 \\ 1, & \text{Otherwise} \end{cases}$$

For the point ((1, 0), 1):
Forward Propogation:

$$\alpha^{(1)} = X = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$z^{(2)} = W^{(1)}a^{(1)} + b^{(1)}$$

$$z^{(2)} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\alpha^{(2)} = f(z^{(2)})$$

$$\alpha^{(2)} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$z^{(3)} = W^{(2)}\alpha^{(2)} + b^{(2)}$$

$$z^{(3)} = \begin{bmatrix} 1 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \end{bmatrix} + 1 = 7$$

$$\alpha^{(3)} = f(z^{(3)})$$

$$\alpha^{(3)} = \frac{1}{1+e^{-7}} = 0.999089$$

2.

Back Propagation:

$$\delta^{(3)} = (f(z^{(3)}) - y)(f'(z^{(3)}))$$

$$\delta^{(3)} = ((0.999089) - (1))((0.999089)(1 - (0.999089))) = -0.000001$$

$$\delta^{(2)} = (W^{(2)})^T \delta^{(3)} \cdot f'(z^{(2)})$$

$$\delta^{(2)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} (-0.000001) \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.000001 \\ -0.000002 \end{bmatrix}$$

$$\delta^{(1)} = (W^{(1)})^T \delta^{(2)} \cdot f'(z^{(1)})$$

$$\delta^{(1)} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} -0.000001 \\ -0.000002 \end{bmatrix} \cdot \begin{bmatrix} ((1)(1 - (1))) \\ ((0)(1 - (0))) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

3.

Partial Derivatives:

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial W^{(2)}} = \delta^{(3)} (\alpha^{(2)})^T$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial W^{(2)}} = (-0.000001) \begin{bmatrix} 2 & 2 \end{bmatrix} = \begin{bmatrix} -0.000002 & -0.000002 \end{bmatrix}$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial W^{(1)}} = \delta^{(2)} (\alpha^{(1)})^T$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial W^{(1)}} = \begin{bmatrix} -0.000001 \\ -0.000002 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} -0.000001 & 0 \\ -0.000002 & 0 \end{bmatrix}$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial b^{(2)}} = \delta^{(2)}$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial b^{(2)}} = \begin{bmatrix} -0.000001 \\ -0.000002 \end{bmatrix}$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial b^{(1)}} = \delta^{(3)}$$

$$\frac{\partial J(W, b; (1, 0), 1)}{\partial b^{(1)}} = \begin{bmatrix} -0.000001 \end{bmatrix}$$

4.

For the point ((0, 1), 0):
Forward Propagation:

$$\alpha^{(1)} = X = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$z^{(2)} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$\alpha^{(2)} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$z^{(3)} = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} + 1 = 10$$

$$\alpha^{(3)} = \frac{1}{1 + e^{-10}} = 0.999955$$

5.

Back Propagation:

$$\delta^{(3)} = ((0.999955) - (0))((0.999955)(1 - (0.999955))) = 0.000450$$

$$\delta^{(2)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} (0.000450) \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.000450 \\ 0.000900 \end{bmatrix}$$

$$\delta^{(1)} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0.000450 \\ 0.000900 \end{bmatrix} \cdot \begin{bmatrix} ((0)(1 - (0))) \\ ((1)(1 - (1))) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

6.

Partial Derivatives:

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial W^{(2)}} = (0.000450) \begin{bmatrix} 3 & 3 \end{bmatrix} = \begin{bmatrix} 0.001350 & 0.001350 \end{bmatrix}$$

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial W^{(1)}} = \begin{bmatrix} 0.000450 \\ 0.000900 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0.000450 \\ 0 & 0.000900 \end{bmatrix}$$

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial b^{(2)}} = \begin{bmatrix} 0.000450 \\ 0.000900 \end{bmatrix}$$

$$\frac{\partial J(W, b; (0, 1), 0)}{\partial b^{(1)}} = \begin{bmatrix} 0.000450 \end{bmatrix}$$

7.

Now, find the gradients:

$$\Delta W^{(1)} = (\delta^{(2)}(\alpha^{(1)})^T)_{((1,0),1)} + (\delta^{(2)}(\alpha^{(1)})^T)_{((0,1),0)}$$

$$\Delta W^{(1)} = \begin{bmatrix} -0.000001 & 0 \\ -0.000002 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0.000450 \\ 0 & 0.000900 \end{bmatrix} = \begin{bmatrix} -0.000001 & 0.000450 \\ -0.000002 & 0.000900 \end{bmatrix}$$

$$\Delta W^{(2)} = (\delta^{(3)}(\alpha^{(2)})^T)_{((1,0),1)} + (\delta^{(3)}(\alpha^{(2)})^T)_{((0,1),0)}$$

$$\Delta W^{(2)} = \begin{bmatrix} -0.000002 & -0.000002 \end{bmatrix} + \begin{bmatrix} 0.001350 & 0.001350 \end{bmatrix} = \begin{bmatrix} 0.001348 & 0.001348 \end{bmatrix}$$

$$\Delta b^{(1)} = (\delta^{(2)})_{((1,0),1)} + (\delta^{(2)})_{((0,1),0)}$$

$$\Delta b^{(1)} = \begin{bmatrix} -0.000001 \\ -0.000002 \end{bmatrix} + \begin{bmatrix} 0.000450 \\ 0.000900 \end{bmatrix} = \begin{bmatrix} 0.000449 \\ 0.000898 \end{bmatrix}$$

$$\Delta b^{(2)} = (\delta^{(3)})_{((1,0),1)} + (\delta^{(3)})_{((0,1),0)}$$

$$\Delta b^{(2)} = \begin{bmatrix} -0.000001 \end{bmatrix} + \begin{bmatrix} 0.000450 \end{bmatrix} = \begin{bmatrix} 0.000449 \end{bmatrix}$$

8.

And now update the parameters:

$$W^{(1)} = W^{(1)} - \frac{\alpha}{2}(\Delta W^{(1)})$$

$$W^{(1)} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \frac{0.2}{2} \begin{bmatrix} -0.000001 & 0.000450 \\ -0.000002 & 0.000900 \end{bmatrix} = \begin{bmatrix} 1.000000 & 1.999955 \\ 3.000000 & 3.999910 \end{bmatrix}$$

$$W^{(2)} = W^{(2)} - \frac{\alpha}{2}(\Delta W^{(2)})$$

$$W^{(2)} = \begin{bmatrix} 1 & 2 \end{bmatrix} - \frac{0.2}{2} \begin{bmatrix} 0.001348 & 0.001348 \end{bmatrix} = \begin{bmatrix} 0.999865 & 1.999865 \end{bmatrix}$$

$$b^{(1)} = b^{(1)} - \frac{\alpha}{2}(\Delta b^{(1)})$$

$$b^{(1)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} - \frac{0.2}{2} \begin{bmatrix} 0.000449 \\ 0.000898 \end{bmatrix} = \begin{bmatrix} 0.999955 \\ -1.000090 \end{bmatrix}$$

$$b^{(2)} = b^{(2)} - \frac{\alpha}{2}(\Delta b^{(2)})$$

$$b^{(2)} = \begin{bmatrix} 1 \end{bmatrix} - \frac{0.2}{2} \begin{bmatrix} 0.000449 \end{bmatrix} = \begin{bmatrix} 0.999955 \end{bmatrix}$$

Question 4:

(Question)

Consider training a neural network. Would overfitting be more of a problem when the training set is small or large? Would overfitting be more or less of a problem when the number of parameters to learn (e.g. the number of weights in a neural network) is small or large?

(Answer(s))

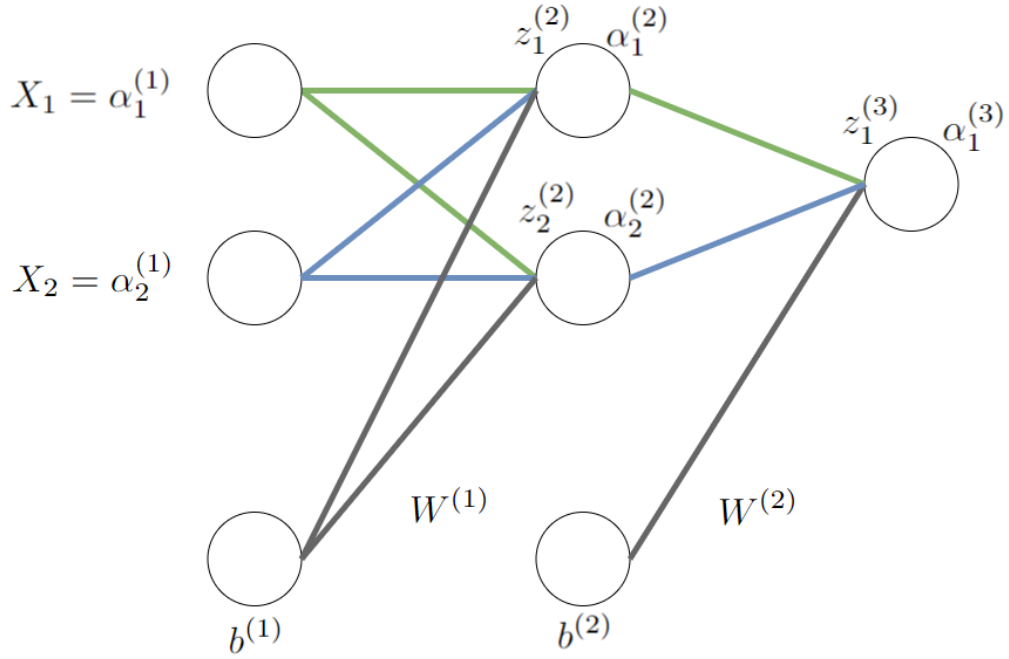
Overfitting is more of a problem when the training set is small. With less samples, the "real" distribution of the data is represented less, so the model will be more prone to fitting noise.
Overfitting is more of an issue when the number of parameters is large. Because there are more weights trying to fit the sample data, it is more likely that noise will be captured as well, leading to overfitting.

Question 5:

(Question)

(Do not turn in) In class we could create a neural network that computed the logical AND function. It is impossible to implement the XOR (exclusive or) function using only a 2 layer neural network. Show it is possible if we have 3 layers where the hidden layer has 2 nodes by assigning values to the weights.

(Answer(s))



Where:

$$f(z) = \frac{1}{1+e^{-z}}$$

$$b^{(1)} = \begin{bmatrix} -25 \\ 75 \end{bmatrix}$$

$$W^{(1)} = \begin{bmatrix} 50 & 50 \\ -50 & -50 \end{bmatrix}$$

$$b^{(2)} = \begin{bmatrix} -75 \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} 50 & 50 \end{bmatrix}$$

Proof:
$X_1 = \alpha_1^{(1)} = 1,\ X_2 = \alpha_2^{(1)} = 1$

$$z_i^{(2)} = (\sum_{j=1}^{2} W_{ij}^{(1)} \alpha_j^{(1)}) + b_j^{(1)}$$

$$z_1^{(2)} = W_{11}^{(1)} \alpha_1^{(1)} + W_{12}^{(1)} \alpha_2^{(1)} + b_1^{(1)}$$
$$z_1^{(2)} = (50)(1) + (50)(1) + (-25) = 75$$

$$z_2^{(2)} = W_{21}^{(1)} \alpha_1^{(1)} + W_{22}^{(1)} \alpha_2^{(1)} + b_2^{(1)}$$
$$z_2^{(2)} = (-50)(1) + (-50)(1) + (75) = -25$$

$$\alpha_1^{(2)} = f(75) \approx 1$$
$$\alpha_2^{(2)} = f(-25) \approx 0$$

$$z_i^{(3)} = (\sum_{j=1}^{2} W_{ij}^{(2)} \alpha_j^{(2)}) + b_j^{(2)}$$

$$z_1^{(3)} = W_{11}^{(2)} \alpha_1^{(2)} + W_{12}^{(2)} \alpha_2^{(2)} + b_1^{(2)}$$
$$z_1^{(3)} = (50)(1) + (50)(0) + (-75) = -25$$

$$\alpha_1^{(3)} = h_{W,b}(\mathbf{x}) = f(-25) \approx 0$$

$X_1 = \alpha_1^{(1)} = 1,\ X_2 = \alpha_2^{(1)} = 0$

$$z_1^{(2)} = (50)(1) + (50)(0) + (-25) = 25$$
$$z_2^{(2)} = (-50)(1) + (-50)(0) + (75) = 25$$

$$\alpha_1^{(2)} = f(25) \approx 1$$
$$\alpha_2^{(2)} = f(25) \approx 1$$

$$z_1^{(3)} = (50)(1) + (50)(1) + (-75) = 25$$

$$\alpha_1^{(3)} = h_{W,b}(\mathbf{x}) = f(25) \approx 1$$

$X_1 = \alpha_1^{(1)} = 0,\ X_2 = \alpha_2^{(1)} = 1$

$$z_1^{(2)} = (50)(0) + (50)(1) + (-25) = 25$$
$$z_2^{(2)} = (-50)(0) + (-50)(1) + (75) = 25$$

$$\alpha_1^{(2)} = f(25) \approx 1$$
$$\alpha_2^{(2)} = f(25) \approx 1$$

$$z_1^{(3)} = (50)(1) + (50)(1) + (-75) = 25$$

$$\alpha_1^{(3)} = h_{W,b}(\mathbf{x}) = f(25) \approx 1$$

$X_1 = \alpha_1^{(1)} = 0,\ X_2 = \alpha_2^{(1)} = 0$

$$z_1^{(2)} = (50)(0) + (50)(0) + (-25) = -25$$
$$z_2^{(2)} = (-50)(0) + (-50)(0) + (75) = 75$$

$$\alpha_1^{(2)} = f(-25) \approx 0$$
$$\alpha_2^{(2)} = f(75) \approx 1$$

$$z_1^{(3)} = (50)(0) + (50)(1) + (-75) = -25$$

$$\alpha_1^{(3)} = h_{W,b}(\mathbf{x}) = f(-25) \approx 0$$

Question 6:

(Question)

(Do not turn in) For each of the following examples determine *if* it could be learned by a neural network with a) no hidden layers, b) 1 hidden layer with two hidden nodes, c) none of the above. Justify your answer by drawing the decision boundaries. (The input layer will have two nodes, and the output layer will have one node.)

(a) + −

(b)

+

−

+ +

(c) − − −

+ −

+ + +

(d)

+

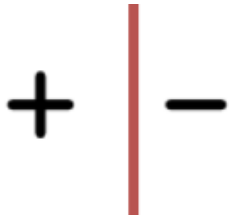
− −

+

(Answer(s))

1.

(a) This example can be learned by a neural network with no hidden layers

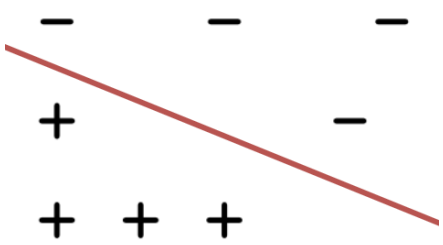


2.

(b) This example can be learned by a neural network with 1 hidden layer with 2 hidden nodes (This hidden layer with 2 hidden nodes can form a linear decision boundary)

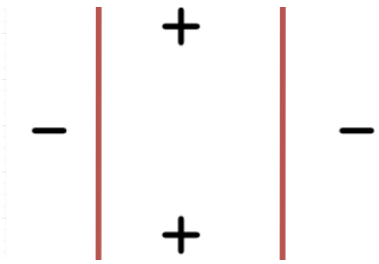
3.

(a) This example can be learned by a neural network with no hidden layers



4.

(b) This example can be learned by a neural network with 1 hidden layer with 2 hidden nodes



Question 7:

(Question)

Modify the neural network implementation we discussed in class to see if you can improve the performance by trying the following:

(a) Add a regularization term to the cost function $\frac{\partial J(W,b)}{\partial W_{ij}^{(\ell)}} = \frac{1}{N}\left[\sum_{i=1}^{N} \frac{\partial J(W,b,\mathbf{x}^{(i)},y^{(i)})}{\partial W_{ij}^{(\ell)}}\right] + \frac{\lambda}{2} W_{ij}^{(\ell)}$ where $\mathbf{x}^{(i)}, y^{(i)}$ are the ith training example. See section 1.2 in `http://adventuresinmachinelearning.com/improve-neural-networks-part-1/`

(b) Try using the *ReLU* activation function, $f(z) = \max(0, z)$. You will notice it is not differentiable at 0, but you can use: $f'(z) = 0$ if $z < 0$ and $f'(z) = 1$ if $z \geq 0$. (You can also try using the *leaky ReLU* activation function.) For more information see `https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning`

(c) Try using the *tanh* activation function, $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. The derivative of *tanh* is $f'(z) = 1 - (f(z))^2$. For more information see `http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/`

(d) Try using the one of the other activation functions

(e) Try changing the number of iterations

(f) Try changing the number of hidden layers

What gave the best performance?

(Answer(s))

Initial Accuracy: 86.648122% (Changes to provided code are shown below)

1.

Accuracy after Regularization: 77.607789%

```
lamb = 0.0005
W[1] += (-alpha * (1.0/N * tri_W[1])) + (lamb / 2 * W[1])
```

2.

Accuracy after ReLU Activation Function: 10.987483%

```
def f(z):
    return np.where(z < 0, 0, z)
def f_deriv(z):
    return np.where(z < 0, 0, 1)
```

3.

Accuracy after tanh Activation Function: 84.422809%

```
def f(z):
    return (np.exp(z) - np.exp(-z)) / (np.exp(z) + np.exp(-z))
def f_deriv(z):
    return 1 - (f(z) ** 2)
```

4.

Accuracy after SoftPlus Activation Function: 5.424200% (overflow encountered)

```
def f(z):
    return np.log(1 + np.exp(z))
def f_deriv(z):
    return 1 / (1 + np.exp(-z))
```

5.

Accuracy after Increasing Iterations to 5000: 91.376912%

```
W, b, avg_cost_func = train_nn(nn_structure, X_train, y_v_train, 5000)
```

6.

Accuracy after Changing Number of Hidden Layers to (64, 7, 7, 10): 6.675939%

```
nn_structure = [64, 7, 7, 10]
```

The modification that gave us the best performance was increasing the number of iterations from 3000 to 5000. Note that this can differ from response to response.