1 Neural Network

1.1 One Step further of Logistic Regression

Model Description
We use two steps of logistic regression to illustrate a simple neural network. We first introduce the notation we use. The subscript j means the jth component of the input/output vector. The superscript of (i) means the ith sample. The superscript $[k]$ indicates kth layer. So the input vector $X_i^{(j)}$ means the jth component in the ith sample. We write the operation in the first layer using matrix notation

$$z^{[1](i)} = W^{[1]}X^{(i)} + b^{[1](i)}$$

$z^{[1](i)}$ means first layer, ith sample. Namely,

$$\begin{pmatrix} z_1^{[1](1)} & z_1^{[1](2)} & ... & z_1^{[1](m)} \\ ... & ... & ... & ... \\ z_4^{[1](1)} & z_4^{[1](2)} & ... & z_4^{[1](m)} \end{pmatrix} = \begin{pmatrix} w_{11}^{[1]} & w_{12}^{[1]} \\ ... & ... \\ ... & ... \\ w_{21}^{[1]} & w_{22}^{[1]} \end{pmatrix} \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & ... & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & ... & x_2^{(m)} \end{pmatrix} + \begin{pmatrix} b_1^{[1]} & ... & b_1^{[1]} \\ ... & ... & ... \\ b_4^{[1]} & ... & b_4^{[1]} \end{pmatrix}$$

$$a^{[1](i)} = tanh(z^{[1](i)})$$

In matrix notation

$$\begin{pmatrix} a_1^{[1](1)} & a_1^{[1](2)} & ... & a_1^{[1](m)} \\ ... & ... & ... & ... \\ a_4^{[1](1)} & a_4^{[1](2)} & ... & a_4^{[1](m)} \end{pmatrix} = tanh \begin{pmatrix} z_1^{[1](1)} & z_1^{[1](2)} & ... & z_1^{[1](m)} \\ ... & ... & ... & ... \\ z_4^{[1](1)} & z_4^{[1](2)} & ... & z_4^{[1](m)} \end{pmatrix}$$

For the 2nd layer, we have

$$z^{[2](i)} = W^{[2]}X^{(i)} + b^{[2](i)}$$

Namely,

$$\begin{pmatrix} z_1^{[2](1)} & z_1^{2](2)} & ... & z_1^{[2](m)} \end{pmatrix} = \begin{pmatrix} w_1^{[2]} & ... & w_4^{[2]} \end{pmatrix} \begin{pmatrix} a_1^{[1](1)} & a_1^{[1](2)} & ... & a_1^{[1](m)} \\ ... & ... & ... & ... \\ a_4^{[1](1)} & a_4^{[1](2)} & ... & a_4^{[1](m)} \end{pmatrix} + \begin{pmatrix} b_1^{[2]} & ... & b_1^{[2]} \end{pmatrix}$$

$$a^{[2](i)} = sigmoid(z^{[2](i)})$$

In full matrix notation

$$\begin{pmatrix} a_1^{[2](1)} & a_1^{[2](2)} & ... & a_1^{[2](m)} \end{pmatrix} = sigmoid \begin{pmatrix} z_1^{[2](1)} & z_1^{[2](2)} & ... & z_1^{[2](m)} \end{pmatrix}$$

Gradient Calculation We now derive the gradient of the model above.
1) Cost function

$$J = -\frac{1}{m}\sum_i^m [y^{(i)}log(a^{[2](i)})(1 - y^{(i)})log(1 - a^{[2](i)})]$$

(i) denotes sample index [j] denotes layer index The sequence of calculating the gradient is
J - $a^{[2]}_{(i)}$ - $z^{[2]}_{(i)}$ - $W^{[2]}_k b^{[2]}$ - chain rule ...
2)

$$\begin{aligned}\frac{\partial J}{\partial a^{[2]}_{(j)}} &= \frac{\partial}{\partial a^{[2]}_{(j)}}(-\frac{1}{m}\sum_i [y_{(i)}log(a^{[2]}_{(i)}) + (1-y_{(i)})log(1-a^{[2]}_{(i)}]) \text{ only i = j survives} \\ &= \frac{\partial}{\partial a^{[2]}_{(j)}}(-\frac{1}{m}[y_{(j)}log(a^{[2]}_{(j)}) + (1-y_{(j)})log(1-a^{[2]}_{(j)})] \\ &= -\frac{1}{m}[\frac{y^{(j)}}{a^{[2]}_{(j)}} - \frac{(1-y^{(j)})}{(1-a^{[2]}_{(j)})}]\end{aligned}$$

3)

$$\begin{aligned}&\frac{\partial a}{\partial z} \\ =&\frac{\partial}{\partial z}(\frac{e^z}{e^z+1}) \\ =&\frac{1}{(e^z+1)^2}(e^z(e^z+1) - e^z e^z) \\ =&\frac{1}{(e^z+1)^2}e^z \\ =&\frac{e^z}{e^z+1}\frac{1}{e^z+1} \\ =&\frac{e^z}{e^z+1}\frac{e^z+1-e^z}{e^z+1} \\ =&a(z)(1-a(z))\end{aligned}$$

4)

$$\begin{aligned}&\frac{\partial J}{\partial z^{[2]}_j} \\ =&\frac{\partial J}{\partial a^{[2]}_{(j)}}\frac{\partial a^{[2]}_{(j)}}{\partial z^{[2]}_{(j)}} \\ =&-\frac{1}{m}[\frac{y^{(j)}}{a^{[2]}_{(j)}} - \frac{(1-y^{(j)})}{(1-a^{[2]}_{(j)})}]a^{[2]}_{(j)}(1-a^{[2]}_{(j)}) \\ =&-\frac{1}{m}(y_{(j)} - y_{(j)}a^{[2]}_{(j)} - a^{[2]}_{(j)} + y_j a^{[2]}_{(j)}) \\ =&-\frac{1}{m}(y_{(j)} - a^{[2]}_{(j)})\end{aligned}$$

matrix form

$$
\begin{aligned}
&\frac{\partial J}{\partial Z^{[2]}}\\
=&(\frac{\partial J}{\partial z_1^{[2]}}, \frac{\partial J}{\partial z_2^{[2]}}, ..., \frac{\partial J}{\partial z_m^{[2]}})\\
=&-\frac{1}{m}(y_{(1)}a_{(1)}^{[2]}, y_{(2)}a_{(2)}^{[2]}, ..., y_{(m)}a_{(m)}^{[2]})\\
=&\frac{1}{m}(A^{[2]}-Y)
\end{aligned}
$$

5)

$$
\begin{pmatrix} z^{[2](1)} & z^{[2](2)} & ... & z^{[2](m)} \end{pmatrix} = \begin{pmatrix} w_1^{[2]} & w_2^{[2]} & ... & w_4^{[2]} \end{pmatrix} \begin{pmatrix} a_1^{[1](1)} & a_1^{[1](2)} & ... & a_1^{[1](m)} \\ ... & ... & ... & ... \\ a_4^{[2](1)} & a_4^{[1](2)} & ... & a_4^{[1](m)} \end{pmatrix} + (b[2], b[2], ..., b[2])
$$

$$
z_{[2](i)} = \sum_l w_l^{[2]} a_l^{[1](i)} + b^{[2]}
$$

$$
\frac{\partial z_i^{[2]}}{\partial w_k^{[2]}} = a_l^{[1](i)}
$$

$$
\frac{\partial z_i^{[2]}}{\partial b^{[2]}} = 1
$$

6)

$$
\begin{aligned}
&\frac{\partial J}{\partial w_k^{[2]}}\\
=&\sum_{(i)} \frac{\partial J}{\partial z_{(i)}^{[2]}} \frac{\partial z_{(i)}^{[2]}}{\partial w_k^{[2]}} \text{ chain rule in higher dimension}\\
=&\sum_i^m -\frac{1}{m}(y_{(i)} - a_{(i)}^{[2]})a_k^{[1](i)}\\
=&\frac{1}{m}\begin{pmatrix} a_{(1)}^{[2]} - y_{(1)} & a_{(2)}^{[2]} - y_{(2)} & ... & a_{(m)}^{[2]} - y_{(m)} \end{pmatrix} \begin{pmatrix} a_k^{(1)[1]} \\ a_k^{(2)[1]} \\ ... \\ a_k^{(m)[1]} \end{pmatrix}\\
&\text{in matrix form}\\
=&\frac{1}{m}[A^{[2]} - Y](a_k^{[1]})^T
\end{aligned}
$$

$$\begin{pmatrix} \frac{\partial J}{\partial w_k^{[2]}}, & \frac{\partial J}{\partial w_k^{[2]}}, & ... & \frac{\partial J}{\partial w_k^{[2]}} \end{pmatrix}$$

$$\frac{1}{m}\begin{pmatrix} a_{(1)}^{[2]} - y_{(1)} & a_{(2)}^{[2]} - y_{(2)} & ... & a_{(m)}^{[2]} - y_{(m)} \end{pmatrix} \begin{pmatrix} a_1^{(1)[1]} & a_2^{(1)[1]} & ... & a_4^{(1)[1]} \\ a_1^{(2)[1]} & a_2^{(2)[1]} & ... & a_4^{(2)[1]} \\ & ... & ... & \\ a_1^{(m)[1]} & a_2^{(m)[1]} & ... & a_4^{(m)[1]} \end{pmatrix}$$

in matrix form

$$=\frac{1}{m}[A^{[2]} - Y](A^{[1]})^T$$

7)

$$\frac{\partial J}{\partial z_l^{[1](j)}} = \frac{\partial J}{\partial z^{[2](j)}} \frac{\partial z^{[2](j)}}{\partial a_l^{[1](j)}} \frac{\partial a_l^{[1](j)}}{\partial z_l^{[1](j)}}$$

$$=\frac{\partial J}{\partial z^{[2](j)}} \frac{\partial \sum_{l=1}^4 w_l^{[2]} a_l^{[1](j)}}{\partial a_l^{[1](j)}} g^{'}(z_l^{[1](j)})$$

$$=\frac{\partial J}{\partial z^{[2](j)}} w_l^{[2]} g^{'}(z_l^{[1](j)})$$

in matrix form

$$\begin{pmatrix} \frac{\partial J}{\partial z_1^{[1](1)}}, & ... & ... & \frac{\partial J}{\partial z_1^{[1](m)}} \\ ... & ... & ... & \\ \frac{\partial J}{\partial z_4^{[1](1)}}, & ... & ... & \frac{\partial J}{\partial z_4^{[1](m)}} \end{pmatrix}$$

$$= \begin{pmatrix} w_l^{[2]} \frac{\partial J}{\partial z^{[2](1)}}, & ... & ... & w_l^{[2]} \frac{\partial J}{\partial z^{[2](m)}} \\ ... & ... & ... & \\ , & ... & ... & \\ w_4^{[2]} \frac{\partial J}{\partial z^{[2](1)}}, & ... & ... & w_4^{[2]} \frac{\partial J}{\partial z^{[2](m)}} \end{pmatrix}$$

$$\circ \begin{pmatrix} g^{'}(z_l^{[1](1)}), & ... & ... & g^{'}(z_l^{[1](m)}) \\ ... & ... & ... & \\ g^{'}(z_4^{[1](1)}), & ... & ... & g^{'}(z_4^{[1](m)}) \end{pmatrix}$$

2 Example: XNOR

The XNOR operation is well-know in computer science. It has the following truth table.

x1	x2	output
0	0	1
0	1	0
1	0	0
1	1	1

This is a well know problem and it is not linearly separable. By saying not linearly separable we mean we are not able to draw a line on the x_1-x_2 plane to separate output 0s and 1s. So we solve this by a 2-layer neural network. The idea of implementing this neural network comes from the following logic operation in the table below.

x1	x2	a1= x1 & x2	a2=not x1 & not x2	output=a1 OR a2
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

In the neural network implementation, the first layer take the input (x_1, x_2) and generates two output nodes (a_1, a_2). The first node a_1 calculates AND operation using the following expression

$$a_1 = tanh(20x_1 + 20x_2 - 30)$$

The second node a_2 calculates $\bar{x}_1 AND \bar{x}_2$ using the following expression

$$a_2 = tanh(-20x_1 - 20x_2 + 10)$$

Then the second layer calculates the final output node y, and it implements OR operation

$$y = tanh(20a_1 + 20a_2 - 10)$$

$$\begin{pmatrix} z_1^{[1](1)} \\ z_2^{[1](1)} \end{pmatrix} = \begin{pmatrix} 20 & 20 \\ -20 & -20 \end{pmatrix} \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} + \begin{pmatrix} -30 \\ 10 \end{pmatrix}$$

$$\begin{pmatrix} a_1^{[1](1)} \\ a_2^{[1](1)} \end{pmatrix} = sigmoid \begin{pmatrix} z_1^{[1](1)} \\ z_2^{[1](1)} \end{pmatrix}$$

$$z^{[2](1)} = \begin{pmatrix} 20 & 20 \end{pmatrix} \begin{pmatrix} a_1^{[2](1)} \\ a_2^{[2](1)} \end{pmatrix} - 10$$

$$y^{[2](1)} = tanh(z^{[2](1)})$$