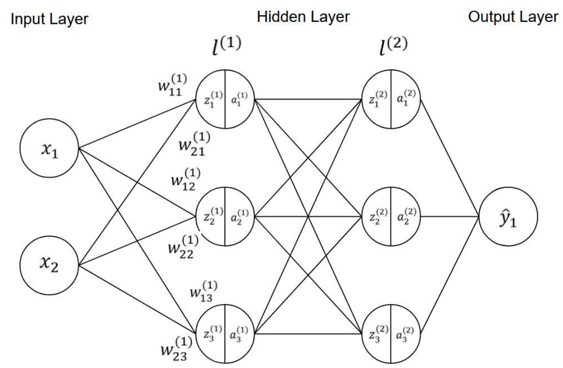
Neuronal Networks

Thursday, February 2, 2023 10:41 AM

Forward Propagation:

In this step we calculate the output of our Neuronal Network:



$z^{(l)} = xw^{(l)} + b^{(l)}$

$a^{(l)} = \sigma\left(z^{(l)}\right)$

Where:
j: j-th layer
i: i-th origin neuron in layer
k: k-th neuron in layer
σ: Activation function
x: Input data
$\hat{y}$: Ouput data
a: Output of hidden layer
w: Weights

Let's calculate $a_i^{(1)}$ in the first hidden layer:

$a_i^{(l)} = \sigma\left(z_i^{(l)}\right)$

$z_1^{(1)} = x_1 w_{11}^{(1)} + x_2 w_{21}^{(1)} + b^{(1)}$

$z_2^{(1)} = x_1 w_{12}^{(1)} + x_2 w_{22}^{(1)} + b^{(1)}$

$z_3^{(1)} = x_1 w_{13}^{(1)} + x_2 w_{23}^{(1)} + b^{(1)}$

$$z_i^{(1)} = xw^{(1)} = [x_1 \quad x_2] \cdot \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} + b^{(1)}$$

$$z_i^{(1)} = \left[x_1 w_{11}^{(1)} + x_2 w_{21}^{(1)} + b^{(1)} \quad x_1 w_{12}^{(1)} + x_2 w_{22}^{(1)} + b^{(1)} \quad x_1 w_{13}^{(1)} + x_2 w_{23}^{(1)} + b^{(1)}\right]$$

$$a_i^{(1)} = \left[\sigma\left(z_1^{(1)}\right) \quad \sigma\left(z_2^{(1)}\right) \quad \sigma\left(z_3^{(1)}\right)\right]$$

Then let's determine the second hidden layer:

$z_i^{(2)} = a_i^{(1)} w^{(2)}$

$$z_i^{(2)} = \left[a_1^{(1)} \quad a_2^{(1)} \quad a_3^{(1)}\right] \cdot \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} \\ w_{31}^{(2)} & w_{32}^{(2)} & w_{33}^{(2)} \end{bmatrix} + b^{(2)}$$

$$z_i^{(2)} = \left[a_1^{(1)} w_{11}^{(1)} + a_2^{(1)} w_{21}^{(1)} + a_3^{(1)} w_{31}^{(1)} + b^{(2)} \quad a_1^{(1)} w_{12}^{(1)} + a_2^{(1)} w_{22}^{(1)} + a_3^{(1)} w_{32}^{(1)} + b^{(2)} \quad a_1^{(1)} w_{13}^{(1)} + a_2^{(1)} w_{23}^{(1)} + a_3^{(1)} w_{33}^{(1)} + b^{(2)}\right]$$

$$a_i^{(2)} = \left[\sigma\left(z_1^{(2)}\right) \quad \sigma\left(z_2^{(2)}\right) \quad \sigma\left(z_3^{(2)}\right)\right]$$

Finally let's calculate the output or prediction:

$\hat{y} = a_i^{(2)} w^{(3)} + b^{(3)}$

$$\hat{y} = \left[a_1^{(2)} \quad a_2^{(2)} \quad a_3^{(2)}\right] \cdot \begin{bmatrix} w_{31}^{(3)} \\ w_{31}^{(3)} \\ w_{31}^{(3)} \end{bmatrix}$$

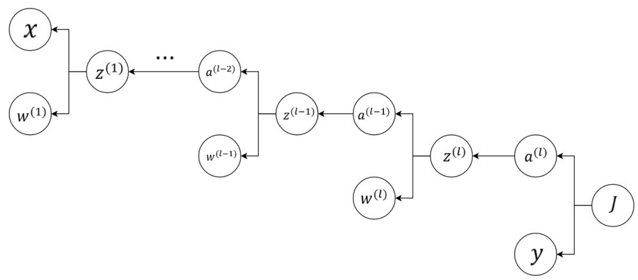
$$\hat{y} = a_1^{(3)} w_{31}^{(3)} + a_2^{(2)} w_{31}^{(3)} + a_3^{(2)} w_{31}^{(3)} + b^{(3)}$$

Back Propagation (MSE) :

$$J = \frac{1}{2m}\sum_{n=1}^{m}(y - \hat{y})^2$$

Gradient Descent Algorithm:

Partial Derivates:



Partial derivate of last layer (One training example):

$$\frac{\partial J_k}{\partial w^{(l)}} = \frac{\partial z^{(l)}}{\partial w^{(l)}}\frac{\partial a^{(l)}}{\partial z^{(l)}}\frac{\partial J_k}{\partial a^{(l)}} = a^{(l-1)}\sigma'\left(z^{(l)}\right)\frac{1}{m}\sum\left(a^{(l)} - y\right)$$

$$\frac{\partial J}{\partial b^{(l)}} = \frac{\partial z^{(l)}}{\partial b^{(l)}}\frac{\partial a^{(l)}}{\partial z^{(l)}}\frac{\partial J}{\partial a^{(l)}} = \sigma'\left(z^{(l)}\right)\frac{1}{m}\sum\left(a^{(l)} - y\right)$$

Partial derivate for multiple training examples:

$$\frac{\partial J}{\partial w^{(l)}} = \frac{1}{m}\sum_{k=1}^{m}\frac{\partial J_k}{\partial w^{(l)}}$$

Partial derivate of the second last layer.

$$\frac{\partial J_k}{\partial w^{(l-1)}} = \frac{\partial z^{(l-1)}}{\partial w^{(l-1)}}\frac{\partial a^{(l-1)}}{\partial z^{(l-1)}}\frac{\partial z^{(l)}}{\partial a^{(l-1)}}\frac{\partial a^{(l)}}{\partial z^{(l)}}\frac{\partial J_k}{\partial a^{(l)}}$$

$$\frac{\partial a^{(l)}}{\partial z^{(l)}}\frac{\partial J_k}{\partial a^{(l)}} = \sigma'\left(z^{(l)}\right)\frac{1}{m}\sum\left(a^{(l)} - y\right)$$

$$\frac{\partial z^{(l-1)}}{\partial w^{(l-1)}}\frac{\partial a^{(l-1)}}{\partial z^{(l-1)}}\frac{\partial z^{(l)}}{\partial a^{(l-1)}} = a^{(l-2)}\sigma'\left(z^{(l-1)}\right)w^{(l)}$$

$$\frac{\partial J_k}{\partial w^{(l-1)}} = a^{(l-2)}\sigma'\left(z^{(l-1)}\right)w^{(l)}\sigma'\left(z^{(l)}\right)\left(\frac{1}{m}\sum\left(a^{(l)} - y\right)\right)$$

$$\frac{\partial J_k}{\partial w^{(l-1)}} = a^{(l-2)}\delta_{l-1}\delta_l$$

Last third layer:

$$\frac{\partial J_k}{\partial w^{(l-2)}} = a^{(l-3)}\delta_{l-2}\delta_{l-1}\delta_l$$

First layer:

$$\frac{\partial J_k}{\partial w^{(1)}} = x\delta_1 \ldots \delta_{l-2}\delta_{l-1}\delta_l$$

First derivate (This case):

$$\frac{\partial J_k}{\partial w^{(1)}} = \frac{\partial z^{(1)}}{\partial w}\frac{\partial a^{(1)}}{\partial z^{(1)}}\frac{\partial z^{(2)}}{\partial a^{(1)}}\frac{\partial a^{(2)}}{\partial z^{(2)}}\frac{\partial z^{(3)}}{\partial a^{(2)}}\frac{\partial a^{(3)}}{\partial z^{(3)}}\frac{\partial J_k}{\partial a^{(3)}}$$

$$\frac{\partial J_k}{\partial w^{(1)}} = x \times \left(w^{(2)}\,\sigma'\left(z^{(1)}\right)\right) \times \left(w^{(3)}\,\sigma'\left(z^{(2)}\right)\right) \times \left(\sigma'\left(z^{(3)}\right)\frac{1}{m}\sum\left(a^{(3)} - y\right)\right)$$

$$\frac{\partial J_k}{\partial w^{(1)}} = x\delta_1\delta_2\delta_3$$

Iterative process (Ignore Input):

First compute error:

$$\delta^{(l)} = \frac{\partial J}{\partial z^{(l)}} = \sigma'\left(z^{(l)}\right)\frac{1}{m}\sum\left(a^{(l)} - y\right)$$

Propagation of errors:

$$\delta^{(l-1)} = \delta^{(l)} w^{(l)} \frac{\partial a^{(l-1)}}{\partial z^{(l-1)}}$$

Cost gradient:

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial w^{(1)}} \\ \frac{\partial J}{\partial b^{(1)}} \\ \vdots \\ \frac{\partial J}{\partial w^{(l)}} \\ \frac{\partial J}{\partial b^{(l)}} \end{bmatrix}$$