Multi-Layer Neural Network

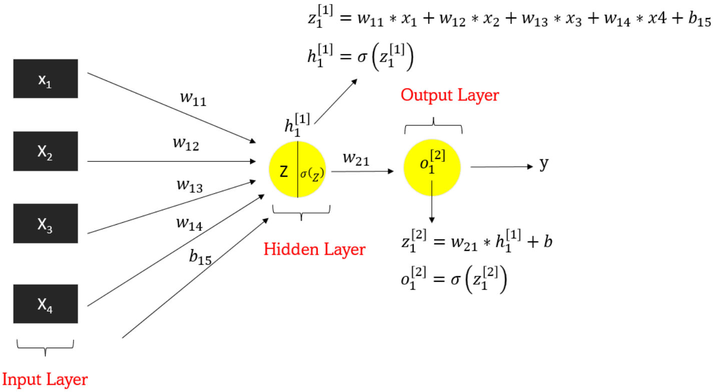
ML002 : Deep Learning Mastery Course

Ayush Singh

Faculty of Artificial Intelligence
Department of Antern Artificial Intelligence

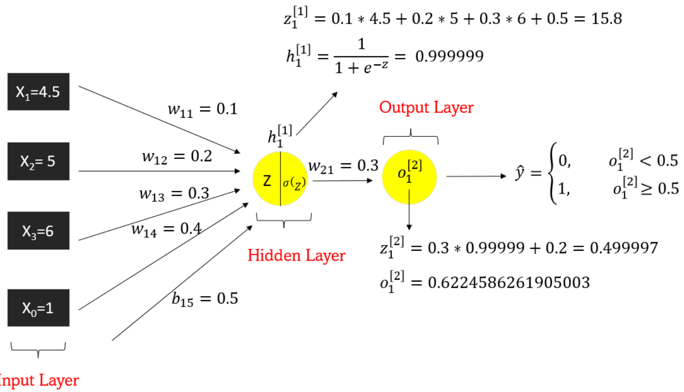
MLP

- Multi-layer Perceptron contain more than one computational layer, In perceptron you were having a input layer and a output layer where all computation was happening in output layer but in multi-layer perceptron, we our input is passed through several layers which is not visible which are called **Hidden Layers**

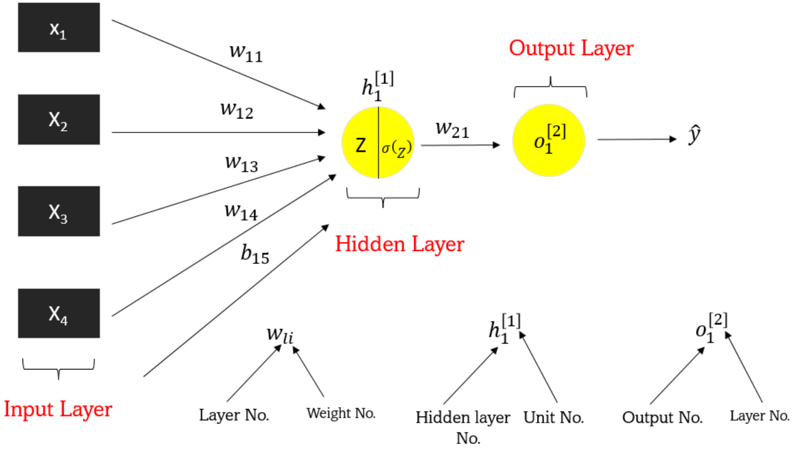


MLP

Example:- Diabetes Prediction System Given Age, Height, BMI



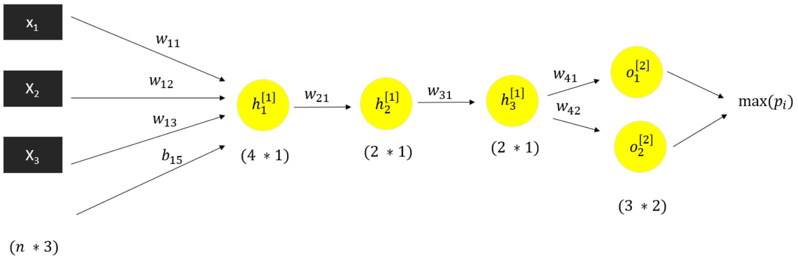
MLP



x_1
x_2
x_3
x_4
Input Layer
w_{11}
w_{12}
w_{13}
w_{14}
b_{15}
$h_1^{[1]}$
Z
$\sigma(z)$
Hidden Layer
w_{21}
Output Layer
$o_1^{[2]}$
$\hat{y}$
w_{li}
Layer No.
Weight No.
$h_1^{[1]}$
Hidden layer No.
Unit No.
$o_1^{[2]}$
Output No.
Layer No.

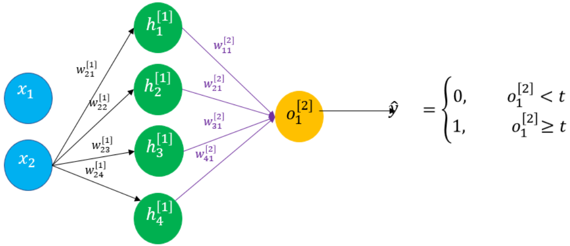
MLP

Example:- Multi-class prediction, Given, Color of Bird, Size of bird, Sound (sweet/non-sweet)



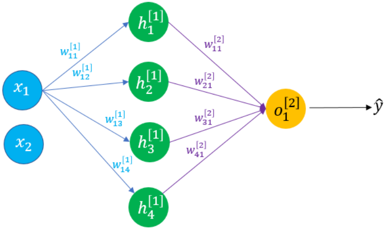
One Hidden Layer Neural Network

One Hidden Layer Neural Network



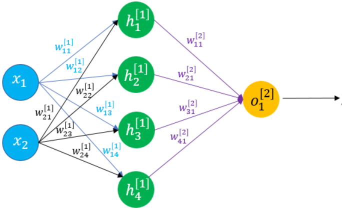
One Hidden Layer Neural Network

One Hidden Layer Neural Network



$$\hat{y} = \begin{cases} 0, & o_1^{[2]} < t \\ 1, & o_1^{[2]} \geq t \end{cases}$$

One Hidden Layer Neural Network



$$\hat{y} = \begin{cases} 0, & o_1^{[2]} < t \\ 1, & o_1^{[2]} \geq t \end{cases}$$

$$z_1^{[1]} = w_{11}^{[1]} * x_1 + w_{21}^{[1]} * x_2 + b_1^{[1]} \qquad h_1^{[1]} = ReLU(z_1^{[1]})$$

$$z_2^{[1]} = w_{12}^{[1]} * x_1 + w_{22}^{[1]} * x_2 + b_2^{[1]} \qquad h_2^{[1]} = ReLU(z_2^{[1]})$$

$$z_3^{[1]} = w_{13}^{[1]} * x_1 + w_{23}^{[1]} * x_2 + b_3^{[1]} \qquad h_2^{[1]} = ReLU(z_2^{[1]})$$

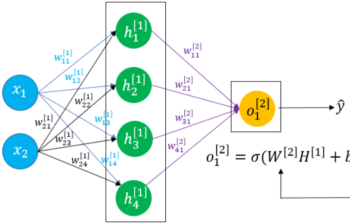
$$z_4^{[1]} = w_{14}^{[1]} * x_1 + w_{24}^{[1]} * x_2 + b_4^{[1]} \qquad h_2^{[1]} = ReLU(z_2^{[1]})$$

$$o_1^{[2]} = \sigma(w_{11}^{[2]} h_1^{[1]} + w_{21}^{[2]} h_2^{[1]} + w_{31}^{[2]} h_3^{[1]} + w_{41}^{[2]} h_4^{[1]} + b_2^{[1]})$$

Writing in Vectorial Format

$$Z^{[1]} = \begin{bmatrix} w_{11}^{[1]} & w_{21}^{[1]} \\ w_{12}^{[1]} & w_{22}^{[1]} \\ w_{13}^{[1]} & w_{23}^{[1]} \\ w_{14}^{[1]} & w_{24}^{[1]} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} \qquad H^{[1]} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \\ \sigma(z_3^{[1]}) \\ \sigma(z_4^{[1]}) \end{bmatrix}$$

$$o_1^{[2]} = \sigma(W^{[2]} H^{[1]} + b^{[2]})$$



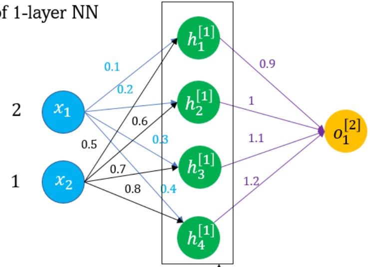
$$o_1^{[2]} = \sigma(W^{[2]} H^{[1]} + b^{[2]})$$

$$W^{[1]} = \begin{bmatrix} w_{11}^{[1]} & w_{21}^{[1]} \\ w_{12}^{[1]} & w_{22}^{[1]} \\ w_{13}^{[1]} & w_{23}^{[1]} \\ w_{14}^{[1]} & w_{24}^{[1]} \end{bmatrix} \qquad W^{[2]} = \begin{bmatrix} w_{11}^{[2]} \\ w_{21}^{[2]} \\ w_{31}^{[2]} \\ w_{41}^{[2]} \end{bmatrix}$$

$$Z^{[1]} = W^{[1]} \vec{x} + b^{[1]}$$

Example:- One Hidden Layer Neural Network

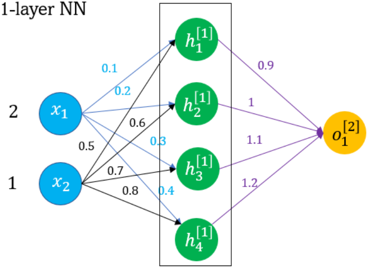
Example:- Worked Example of 1-layer NN



$$Z^{[1]} = \underbrace{\begin{bmatrix} 0.1 & 0.5 \\ 0.2 & 0.6 \\ 0.3 & 0.7 \\ 0.4 & 0.8 \end{bmatrix}}_{(4\,*\,2)} \underbrace{\begin{bmatrix} 2 \\ 1 \end{bmatrix}}_{(2\,*\,1)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{(4\,*\,1)} = \underbrace{\begin{bmatrix} 0.7 \\ 1 \\ 1.3 \\ 1.6 \end{bmatrix}}_{=\,(4\,*\,1)}$$

$$H^{[1]} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \\ \sigma(z_3^{[1]}) \\ \sigma(z_4^{[1]}) \end{bmatrix} = \underbrace{\begin{bmatrix} \sigma(0.7) \\ \sigma(1) \\ \sigma(1.3) \\ \sigma(1.6) \end{bmatrix}}_{\text{Applying Sigmoid Activation Layer}} = \begin{bmatrix} 0.668 \\ 0.731 \\ 0.785 \\ 0.832 \end{bmatrix}$$

Example:- Worked Example of 1-layer NN

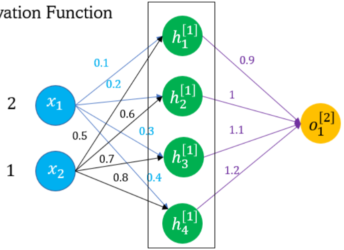


$$o_1^{[2]} = \begin{bmatrix} 0.9 \\ 1 \\ 1.1 \\ 1.2 \end{bmatrix} \begin{bmatrix} 0.668 \\ 0.731 \\ 0.785 \\ 0.832 \end{bmatrix} = (0.9 * 0.668) + (1 * 0.731) + (1.1 * 0.785) + (1.2 * 0.832) = 2.586918464$$

$$o_1^{[2]} = \sigma(2.586918464) = 0.9300149142362439$$

Applying Different Activation Functions

Example:- Applying ReLU Activation Function



Exercise:- Get the output

$$Z^{[1]} = \begin{bmatrix} 0.1 & 0.5 \\ 0.2 & 0.6 \\ 0.3 & 0.7 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 1 \\ 1.3 \\ 1.6 \end{bmatrix}$$

$$(4 * 2) \quad (2 * 1) \quad (4 * 1) = (4 * 1)$$

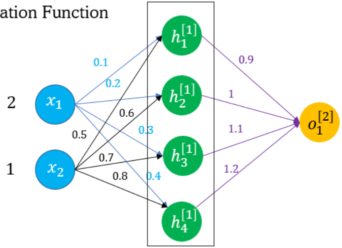
$$H^{[1]} = \underbrace{\begin{bmatrix} \mathrm{ReLU}(z_1^{[1]}) \\ \mathrm{ReLU}(z_2^{[1]}) \\ \mathrm{ReLU}(z_3^{[1]}) \\ \mathrm{ReLU}(z_4^{[1]}) \end{bmatrix} = \begin{bmatrix} \mathrm{ReLU}(0.7) \\ \mathrm{ReLU}(1) \\ \mathrm{ReLU}(1.3) \\ \mathrm{ReLU}(1.6) \end{bmatrix}}_{\text{Applying ReLU Activation Layer}} = \begin{bmatrix} 0.7 \\ 1 \\ 1.3 \\ 1.6 \end{bmatrix}$$

Applying Different Activation Functions

Example:- Applying Tanh Activation Function

Tanh

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



Exercise:- Get the output

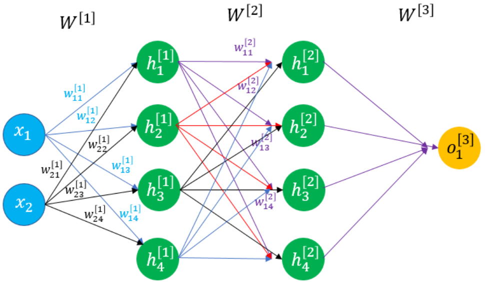
$$Z^{[1]} = \begin{bmatrix} 0.1 & 0.5 \\ 0.2 & 0.6 \\ 0.3 & 0.7 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 1 \\ 1.3 \\ 1.6 \end{bmatrix}$$

$$(4 * 2) \quad (2 * 1) \quad (4 * 1) = (4 * 1)$$

$$H^{[1]} = \begin{bmatrix} \mathrm{Tanh}(z_1^{[1]}) \\ \mathrm{Tanh}(z_2^{[1]}) \\ \mathrm{Tanh}(z_3^{[1]}) \\ \mathrm{Tanh}(z_4^{[1]}) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{Tanh}(0.7) \\ \mathrm{Tanh}(1) \\ \mathrm{Tanh}(1.3) \\ \mathrm{Tanh}(1.6) \end{bmatrix}}_{\text{Applying ReLU Activation Layer}} = \begin{bmatrix} 0.604 \\ 0.761 \\ 0.861 \\ 0.921 \end{bmatrix}$$

Two Layer NN

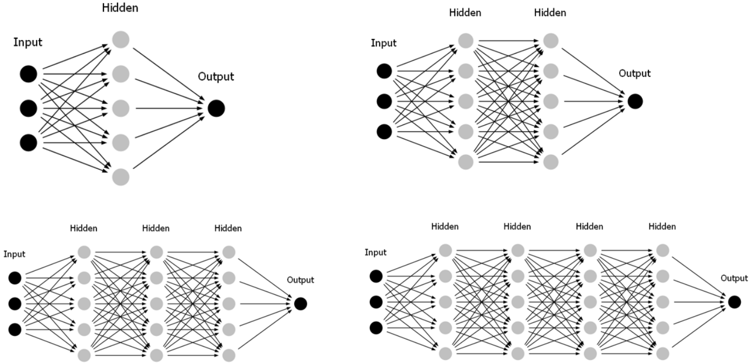
Two Layer Hidden Neural Network



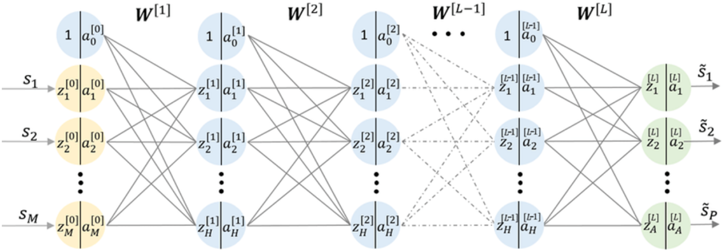
$$W^{[1]} = \begin{bmatrix} w_{11}^{[1]} & w_{21}^{[1]} \\ w_{12}^{[1]} & w_{22}^{[1]} \\ w_{13}^{[1]} & w_{23}^{[1]} \\ w_{14}^{[1]} & w_{24}^{[1]} \end{bmatrix}_{4*2} \quad W^{[2]} = \begin{bmatrix} w_{11}^{[2]} & w_{21}^{[2]} & w_{31}^{[2]} & w_{41}^{[2]} \\ w_{12}^{[2]} & w_{22}^{[2]} & w_{32}^{[2]} & w_{42}^{[2]} \\ w_{13}^{[2]} & w_{23}^{[2]} & w_{33}^{[2]} & w_{43}^{[2]} \\ w_{14}^{[2]} & w_{24}^{[2]} & w_{34}^{[2]} & w_{44}^{[2]} \end{bmatrix}_{4*4} \quad W^{[3]} = \begin{bmatrix} w_{11}^{[3]} \\ w_{21}^{[3]} \\ w_{31}^{[3]} \\ w_{41}^{[3]} \end{bmatrix}_{4*1}$$

Different Layer Neural Network

Different Layer Neural Network

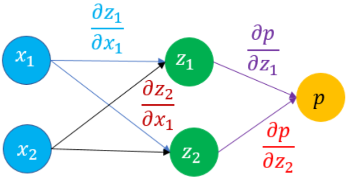


L-Layer Neural Network



Chain Rule

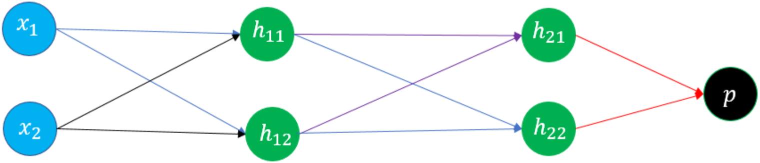
Derivatives of Computation Graph – Chain Rule



$$\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial z_1}\frac{\partial z_1}{\partial x_1} + \frac{\partial p}{\partial z_2}\frac{\partial z_2}{\partial x_1}$$

$$\frac{\partial p}{\partial x_2} = \frac{\partial p}{\partial z_2}\frac{\partial z_2}{\partial x_2} + \frac{\partial p}{\partial z_1}\frac{\partial z_1}{\partial x_2}$$

Chain Rule - Deep Computational Graphs



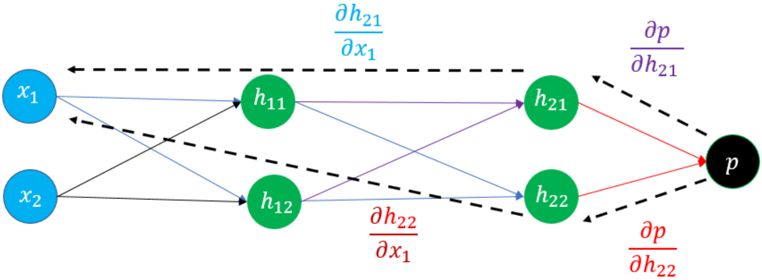
$h_{11} = h_{11}(x_1, x_2)$, $h_{21} = h_{21}(h_{11}, h_{12})$ $p = p(h_{21}, h_{22})$

$h_{12} = h_{12}(x_1, x_2)$, $h_{22} = h_{22}(h_{11}, h12)$

Chain Rule:- $\boxed{\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_{21}} \frac{\partial h_{21}}{\partial x_1} + \frac{\partial p}{\partial h_{22}} \frac{\partial h_{22}}{\partial x_1}}$

Chain Rule - Deep Computational Graphs

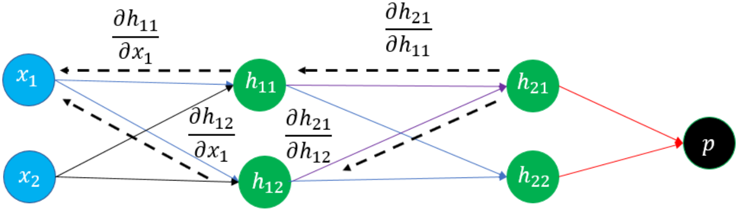
$$\boxed{\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_{21}} \frac{\partial h_{21}}{\partial x_1} + \frac{\partial p}{\partial h_{22}} \frac{\partial h_{22}}{\partial x_1}}$$



Chain Rule - Deep Computational Graphs

$$\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_{21}} \boxed{\frac{\partial h_{21}}{\partial x_1}} + \frac{\partial p}{\partial h_{22}} \frac{\partial h_{22}}{\partial x_1}$$

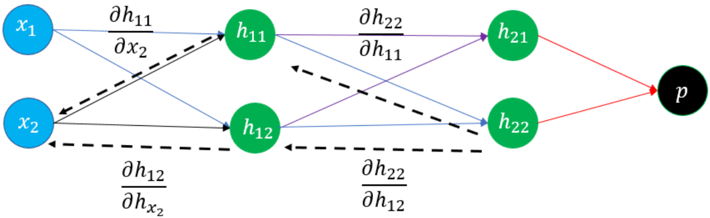
$$\boxed{\frac{\partial h_{21}}{\partial x_1} = \frac{\partial h_{21}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial x_1} + \frac{\partial h_{21}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial x_1}}$$



Chain Rule - Deep Computational Graphs

$$\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_{21}} \frac{\partial h_{21}}{\partial x_1} + \frac{\partial p}{\partial h_{22}} \boxed{\frac{\partial h_{22}}{\partial x_1}}$$

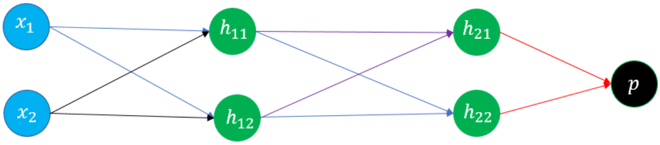
$$\boxed{\frac{\partial h_{22}}{\partial x_1} = \frac{\partial h_{22}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial x_2} + \frac{\partial h_{22}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial h_{x_2}}}$$



Chain Rule - Deep Computational Graphs

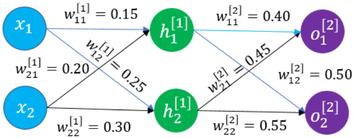
$$\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_{21}}\left(\frac{\partial h_{21}}{\partial h_{11}}.\frac{\partial h_{11}}{\partial x_1} + \frac{\partial h_{21}}{\partial h_{12}}.\frac{\partial h_{12}}{\partial x_1}\right) + \frac{\partial p}{\partial h_{22}}\left(\frac{\partial h_{22}}{\partial h_{11}}.\frac{\partial h_{11}}{\partial x_2} + \frac{\partial h_{22}}{\partial h_{12}}.\frac{\partial h_{12}}{\partial h_{x_2}}\right)$$

$$\frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_{21}}\frac{\partial h_{21}}{\partial h_{11}}\frac{\partial h_{11}}{\partial x_1} + \frac{\partial p}{\partial h_{21}}\frac{\partial h_{21}}{\partial h_{12}}.\frac{\partial h_{12}}{\partial x_1} + \frac{\partial p}{\partial h_{22}}\frac{\partial h_{22}}{\partial h_{11}}.\frac{\partial h_{11}}{\partial x_2} + \frac{\partial p}{\partial h_{22}}\frac{\partial h_{22}}{\partial h_{12}}.\frac{\partial h_{12}}{\partial h_{x_2}}$$



An iteration of Back propagation

Given a Neural Network



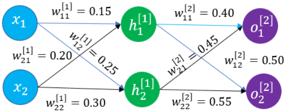
$$X = \begin{bmatrix} 0.05 \\ 0.10 \end{bmatrix}$$

$$W^{[1]} = \begin{bmatrix} 0.15 & 0.25 \\ 0.20 & 0.30 \end{bmatrix} \quad W^{[2]} = \begin{bmatrix} 0.40 & 0.50 \\ 0.45 & 0.55 \end{bmatrix}$$

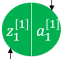
$$b^{[1]} = \begin{bmatrix} 0.35 \\ 0.35 \end{bmatrix} \qquad b^{[2]} = \begin{bmatrix} 0.60 \\ 0.60 \end{bmatrix}$$

An iteration of Back propagation

Forward Propagation:-



$$a_1^{[1]} = \sigma\left(z_1^{[1]}\right) = \frac{1}{1 + e^{-z_1^{[1]}}}$$

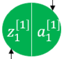


$$z_1^{[1]} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$a_1^{[1]} = \sigma(0.3775) = 0.593269992$$

$$h_1^{[1]} = 0.593269992$$

$$a_1^{[1]} = \sigma\left(z_1^{[1]}\right) = \frac{1}{1 + e^{-z_1^{[1]}}}$$



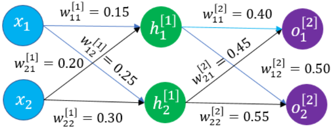
$$z_2^{[1]} = 0.25 * 0.05 + 0.3 * 0.1 + 0.35 * 1 = 0.3775$$

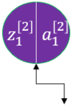
$$a_2^{[1]} = \sigma(0.3775) = 0.5941745$$

$$h_2^{[1]} = 0.5941745$$

An iteration of Back propagation

Forward Propagation:-





$$z_1^{[2]} = 0.4 * 0.593269992 + 0.45 * 0.5941745 + 0.6$$
$$= 1.1046865218$$

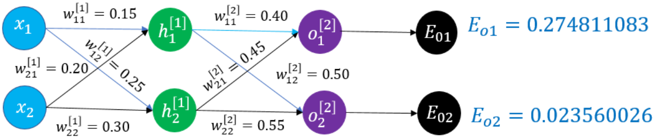
$$a_1^{[2]} = \sigma(1.1046865218) = 0.7511371883290436$$

$$o_1^{[2]} = 0.7511371883290436$$

$$o_2^{[2]} = 0.77298465$$

An iteration of Back propagation

Calculating Error:-



J = 0.298371109

$o_1^{[2]} = 0.01 \quad o_1^{[2]}\_hat = 0.75136507$

$o_2^{[2]} = 0.99 \quad o_1^{[2]}\_hat = 0.77298465$

Backward Propagation

$$\frac{\partial J}{\partial w_{11}^{[2]}} = \boxed{\frac{\partial J}{\partial a_1^{[2]}}} * \boxed{\frac{\partial a_1^{[2]}}{\partial z_1^{[2]}}} * \boxed{\frac{\partial z_1^{[2]}}{\partial w_{11}^{[2]}}}$$

$$z_1^{[2]} = w_{11}^{[2]} * h_1^{[1]} + w_{21}^{[2]} * h_2^{[1]} + b_1^{[2]}$$

$$\frac{\partial z_1^{[2]}}{\partial w_{11}^{[2]}} = 0.593269992$$

$$J = \frac{1}{2}(y - h(x))^2$$

$$\frac{\partial J}{\partial a_1^{[2]}} = 2 * \frac{1}{2}(y - h(x)) * -1 + 0$$

$$\frac{\partial J}{\partial a_1^{[2]}} = -(y - h(x))$$

$$\frac{\partial J}{\partial a_1^{[2]}} = -(0.01 - 0.75136507) = 0.74136507$$

$$a_1^{[2]} = \frac{1}{1 + e^{-z_1^{[2]}}}$$

Already Derived in Intro to NN Chapter

$$\frac{\partial J}{\partial a_1^{[2]}} = \sigma(z_1^{[2]})(1 - \sigma(z_1^{[2]}))$$

$$\frac{\partial J}{\partial a_1^{[2]}} = 0.75136507(1 - 0.75136507)$$

$$\frac{\partial J}{\partial a_1^{[2]}} = 0.186815602$$

An iteration of Back propagation

Delta Rule:- You'll often see this calculation combined in the form of the delta rule

$$\frac{\partial J}{\partial w_{11}^{[2]}} = -\left(y_{o_1^{[2]}} - \hat{y}_{o_1^{[2]}}\right) * \sigma\left(z_1^{[2]}\right)\left(1 - \sigma\left(z_1^{[2]}\right)\right) * a_1^{[1]}$$

$$\delta_{o_1^{[2]}} = -\left(y_{o_1^{[2]}} - \hat{y}_{o_1^{[2]}}\right) * \sigma\left(z_1^{[2]}\right)\left(1 - \sigma\left(z_1^{[2]}\right)\right)$$

$$\frac{\partial J}{\partial w_{11}^{[2]}} = \delta_{o_1^{[2]}} * a_1^{[1]}$$

An iteration of Back propagation

$$\frac{\partial J}{\partial w_{11}^{[2]}} = \frac{\partial J}{\partial a_1^{[2]}} * \frac{\partial a_1^{[2]}}{\partial z_1^{[2]}} * \frac{\partial z_1^{[2]}}{\partial w_{11}^{[2]}}$$

$$\frac{\partial J}{\partial w_{11}^{[2]}} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

$$w_{11}^{[2]*} = w_{11}^{[2]} - \alpha \frac{\partial J}{\partial w_{11}^{[2]}}$$

$$= 0.4 - 0.5 * 0.082167041$$

$$= 0.35891648$$

An iteration of Back propagation

$$\frac{\partial J}{\partial w_{21}^{[2]}} = \frac{\partial J}{\partial a_1^{[2]}} * \frac{\partial a_1^{[2]}}{\partial z_1^{[2]}} * \frac{\partial z_1^{[2]}}{\partial w_{21}^{[2]}}$$

$$\frac{\partial J}{\partial w_{21}^{[2]}} = 0.74136507 * 0.186815602 * 0.5941745 = 0.08229231374$$

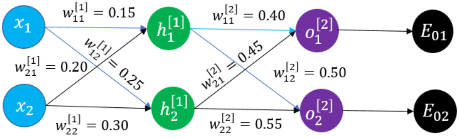
$$w_{21}^{[2]*} = w_{11}^{[2]} - \alpha \frac{\partial J}{\partial w_{21}^{[2]}}$$

$$= 0.45 - 0.5 * 0.08229231374$$

$$= 0.40885384313$$

$$\left.\begin{aligned} w_{12}^{[2]*} &= 0.511301270 \\ w_{22}^{[2]*} &= 0.561370121 \end{aligned}\right\} \text{We can do same thing with other two updates!!!}$$

Updating Values of Parameters of first hidden layer!!!!



$$\frac{\partial J}{\partial w_{11}^{[1]}} = \frac{\partial J}{\partial a_1^{[1]}} * \frac{\partial a_1^{[1]}}{\partial z_1^{[1]}} * \frac{\partial z_1^{[1]}}{\partial w_{11}^{[1]}}$$

$$\frac{\partial J}{\partial a_1^{[1]}} = \frac{\partial E_{o1}}{\partial a_1^{[1]}} + \frac{\partial E_{o2}}{\partial a_1^{[1]}}$$

An iteration of Back propagation

$$\frac{\partial J}{\partial a_1^{[1]}} = \frac{\partial E_{o1}}{\partial a_1^{[1]}} + \frac{\partial E_{o2}}{\partial a_1^{[1]}}$$

$$\frac{\partial E_{o1}}{\partial a_1^{[1]}} = \frac{\partial E_{o1}}{\partial z_1^{[2]}} * \frac{\partial z_1^{[2]}}{\partial a_1^{[1]}}$$

$$\frac{\partial E_{o1}}{\partial z_1^{[2]}} = \frac{\partial E_{o1}}{\partial a_1^{[2]}} * \frac{\partial a_1^{[2]}}{\partial z_1^{[2]}} = 0.74136507 * 0.186815602 = 0.13849562$$

$$z_1^{[2]} = w_{11}^{[2]} * h_1^{[1]} + w_{21}^{[2]} * h_2^{[1]} + b_1^{[2]}$$

$$\frac{\partial z_1^{[2]}}{\partial a_1^{[1]}} = w_{11}^{[2]} = 0.40$$

Plugging them in:

$$\frac{\partial E_{o1}}{\partial a_1^{[1]}} = 0.13849562 * 0.40 = 0.055399425$$

$$\frac{\partial E_{o2}}{\partial a_1^{[1]}} = -0.019049119$$

$$\boxed{\frac{\partial J}{\partial a_1^{[1]}} = 0.055399425 + -0.019049119 = 0.036350306}$$

An iteration of Back propagation

$$\frac{\partial J}{\partial w_{11}^{[1]}} = \frac{\partial J}{\partial a_1^{[1]}} * \frac{\partial a_1^{[1]}}{\partial z_1^{[1]}} * \frac{\partial z_1^{[1]}}{\partial w_{11}^{[1]}}$$

$$a_1^{[1]} = \frac{1}{1 + e^{-z_1^{[1]}}}$$

$$\frac{\partial a_1^{[1]}}{\partial z_1^{[1]}} = \sigma\left(z_1^{[1]}\right)\left(1 - \sigma\left(z_1^{[1]}\right)\right) = 0.593269992 * (1 - 0.593269992) = 0.241300709$$

$$z_1^{[1]} = w_{11}^{[1]} * x_1 + w_{21}^{[1]} * x_2 + b_1^{[1]}$$

$$\frac{\partial z_1^{[1]}}{\partial w_{11}^{[1]}} = x_1 = 0.05$$

$$\frac{\partial J}{\partial w_{11}^{[1]}} = \frac{\partial J}{\partial a_1^{[1]}} * \frac{\partial a_1^{[1]}}{\partial z_1^{[1]}} * \frac{\partial z_1^{[1]}}{\partial w_{11}^{[1]}} \qquad \frac{\partial J}{\partial w_{11}^{[1]}} = 0.036350306 * 0.241300709 * 0.05 = 0.000438568$$

Backpropagation - chain rule

$$h_j = h_j(\overbrace{w_1, w_2, \dots, w_p}^{\text{Input variable}})$$

$$J = J(\underbrace{h_1, h_2, \dots, h_p}_{\text{Intermediate variable}})$$

Output variable

$$\frac{\partial J}{\partial w_j} = \sum_{j=1}^{k} \frac{\partial J}{\partial h_j} \frac{\partial h_j}{\partial w_j}$$