

Algorithm MLP

for i in range(epochs):
for row in x_train:
forward pass

for each layer l:

for each neuron:

$$\text{output} \rightarrow W^l \cdot x + B^l \quad [x \rightarrow \text{single row of } x]$$

$$\text{loss} \rightarrow \text{log loss} \rightarrow -y \log(\hat{y}) - (1-y) \log(1-\hat{y})$$

Back propagation

layer Outputs $\rightarrow \left\{ \begin{array}{l} \text{for each layer, each neuron} \\ 0: [\text{neuron1}, \text{neuron2}, \dots] \\ 1: [\dots] \\ \vdots \end{array} \right\}$

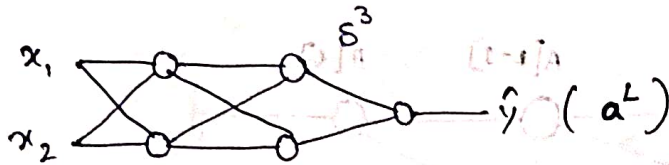
* First init it as all 0 or 1. then on back pass it will be updated.

Errors →

layer Deltas →

for each neuron

$$\left\{ \begin{array}{l} 0: [\delta_1, \delta_2, \dots] \\ 1: [\delta_1, \delta_2, \dots] \end{array} \right\}$$



$$\delta^3 = (a_j - \hat{y}_j) \cdot \sigma'(a_j) \rightarrow \frac{\delta L}{\delta \hat{y}} \rightarrow a_j(1-a_j)$$

Output layer

\hat{y} of j neuron

y_{actual}

$$\delta^2 \Rightarrow \delta_1^2 = (w_{11}^3 \cdot \delta^3) \cdot \sigma'(a_j)$$

$$\delta_2^2 = (w_{21}^3 \cdot \delta^3) \cdot \sigma'(a_j)$$

Hidden

$$\delta^2 = [\delta_1^2, \delta_2^2]$$

Output layer Delta

$$\begin{array}{l} \bigcirc \rightarrow \hat{y}[0] \\ \bigcirc \rightarrow \hat{y}[1] \\ \bigcirc \rightarrow \hat{y}[2] \end{array}$$

If output neurons → 3,

$$\delta_1^L = (\hat{y}[0] - y[0]) \cdot \sigma'(\hat{y}[0])$$

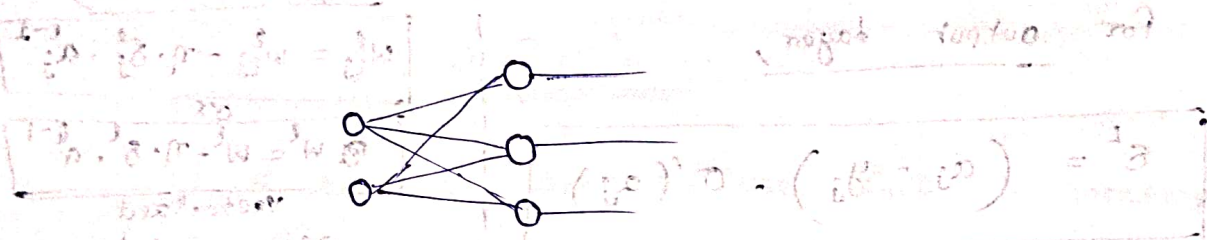
$$\delta_2^L = (\hat{y}[1] - y[1]) \cdot \sigma'(\hat{y}[1])$$

$$\delta_3^L = (\hat{y}[2] - y[2]) \cdot \sigma'(\hat{y}[2])$$

L = last layer

l = hidden layers

2 neurons in L-1 Layer \rightarrow



$$N \rightarrow \begin{bmatrix} w_{11}^L & w_{12}^L & w_{13}^L \\ w_{21}^L & w_{22}^L & w_{23}^L \end{bmatrix}$$

$$\delta_1^L = (w_{11}^L \cdot \delta_1^L + w_{12}^L \cdot \delta_2^L + w_{13}^L \cdot \delta_3^L) \cdot \sigma'(a_1^L)$$

1st neurons output of current layer

$$\delta_2^L = (w_{21}^L \cdot \delta_1^L + w_{22}^L \cdot \delta_2^L + w_{23}^L \cdot \delta_3^L) \cdot \sigma'(a_2^L)$$

$$\delta^L \rightarrow w^{L+1} \cdot \delta^{L+1} \cdot \sigma'(a^L)$$

for this network if, $\delta^L \rightarrow (3 \times 1)$

$$\delta^L \rightarrow \begin{matrix} (2 \times 3) & (3 \times 1) & (2 \times 1) \\ w^{L+1} & \cdot & \delta^{L+1} & \cdot & \sigma'(a^L) \\ \downarrow & & & & \\ (2 \times 1) & \cdot & (2 \times 1) \end{matrix}$$

$$\delta^L \rightarrow (2 \times 1)$$

if $\delta^L \rightarrow (1 \times 1)$

A diagram showing three input nodes on the left connected to a single output node on the right. All three input nodes connect to the single output node.

$$\delta^L \rightarrow \begin{matrix} (3 \times 1) & \cdot & (1 \times 1) & \cdot & \sigma'(a^L) \\ \downarrow & & & & \\ (3 \times 1) & \cdot & (3 \times 1) \end{matrix}$$

$$W \rightarrow \begin{bmatrix} w_{11}^L \\ w_{21}^L \\ w_{31}^L \end{bmatrix}$$

$$\Rightarrow (3 \times 1) \cdot (3 \times 1)$$

$$\delta^L \rightarrow (3 \times 1)$$

So,

for output layers,

$$\delta^L = (a_j - y_j) \cdot \sigma'(a_j)$$

for hidden layers,

$$\delta^l = \left(\underbrace{w^{l+1}}_{\text{dot product}} \cdot \underbrace{\delta^{l+1}}_{\text{normal multiplication}} \right) \cdot \sigma'(a_j)$$

~~Also called error~~

Weight updation,

$$w_{ij}^l = w_{ij}^l - \eta \cdot \delta_j^l \cdot a_j^{l-1}$$

or

$$\odot w^l = w^l - \eta \cdot \delta^l \cdot a^{l-1}$$

vectorized

Bias update

$$b^l = b^l - \eta \cdot \delta^l$$

dot product

normal multiplication