Algorithm MLP

for i in range(epochs):
    for row in xtrain:

forward pass

for each layer l:

for each neuron:

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

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

Back propagation

Layer Outputs $\rightarrow$ {

for each layer, each neuron

0: [neuron1 , neuron2 . . .]

1: [. . ——— - - —— . . .]

:

}

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

Errors →

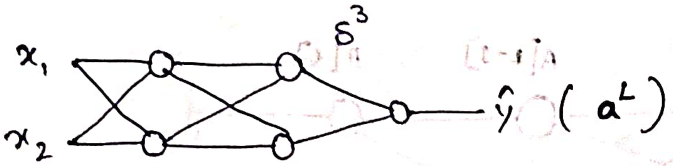
LayerDeltas → {

for each neuron

0 : [δ_1 , δ_2 ...]

1 : [δ_1 , δ_2 ...]

}



$$\underbrace{\delta^3}_{\text{output layer}} = (\underbrace{a_j}_{\hat{y} \text{ of } j \text{ neuron}} - \underbrace{y_j}_{y \text{ actual}}) \cdot \sigma'(a_j)$$

$\frac{\delta L}{\delta \hat{y}} \rightarrow a_j(1-a_j)$

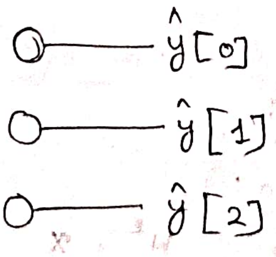
$\underbrace{\delta^2}_{\text{Hidden}} \Rightarrow$

$$\delta^2_1 = (w^3_{11} \cdot \delta^3) \cdot \sigma'(a_j)$$

$$\delta^2_2 = (w^3_{21} \cdot \delta^3) \cdot \sigma'(a_j)$$

So, $\delta^2 = [\, \delta^2_1 , \delta^2_2 \,]$

Output Layer Delta



If output neurons → 3,

$$\delta^L_1 = (\hat{y}[0] - y[0]) \cdot \sigma'(\hat{y}[0])$$

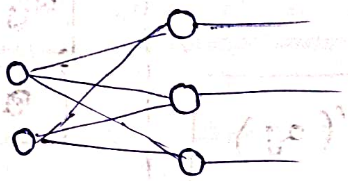
$$\delta^L_2 = (\hat{y}[1] - y[1]) \cdot \sigma'(\hat{y}[1])$$

$$\delta^L_3 = (\hat{y}[2] - y[2]) \cdot \sigma'(\hat{y}[2])$$

L = last layer.

l = hidden layers.

if 2 neurons in L-1 Layer →



$W \rightarrow \quad w^l_{11}, \; w^l_{12}, \; w^l_{13}$

$\qquad\;\; w^l_{21}, \; w^l_{22}, \; w^l_{23}$

$$\delta^l_1 = \left(w^L_{11} \cdot \delta^L_1 + w^L_{12} \cdot \delta^L_2 + w^L_{13} \cdot \delta^L_3\right) \cdot \sigma'(a^l_1)$$

1st neurons output of current Layer

$$\delta^l_2 = w^L_{21} \cdot \delta^L_1 + w^L_{22} \cdot \delta^L_2 + w^L_{23} \cdot \delta^L_3) \cdot \sigma'(a^l_2)$$

$$\delta^l \rightarrow W^{l+1} \cdot \delta^{l+1} \cdot \sigma'(a^l)$$

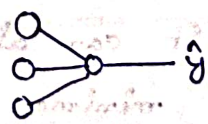
~~for this network~~ if, $\delta^L \rightarrow (3\times1)$,

$$\delta^l \rightarrow \underset{(2\times3)}{w^{l+1}} \cdot \underset{(3\times1)}{\delta^{L+1}} \cdot \underset{(2\times1)}{\sigma'(a^l)}$$

$$\rightarrow (2\times1) \cdot (2\times1)$$

$$\delta^l \rightarrow (2\times1)$$

if $\delta^L \rightarrow (1\times1) \longrightarrow$



$$\rightarrow \delta^l \rightarrow W^{l+1} \cdot \delta^{l+1} \cdot \sigma'(a^l)$$

$$(3\times1) \cdot (1\times1) \qquad (3\times1)$$

$$W \rightarrow \begin{bmatrix} w^L_{11} \\ w^l_{21} \\ w^l_{31} \end{bmatrix}$$

$$\Rightarrow (3\times1) \cdot (3\times1$$

$$\delta^l \Rightarrow (3\times1)$$

So,

for <u>output layer</u>,

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

<u>Weight updation</u>,

$$\boxed{w^l_{ij} = w^l_{ij} - \eta\cdot\delta^l_j\cdot a^{l-1}_j}$$

or

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

vectorized

<u>Bias update</u>

$$\boxed{B^l = B^l - \eta\cdot\delta^l}$$

for <u>hidden layers</u>,

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