ANN ASSIGNMENT - 1

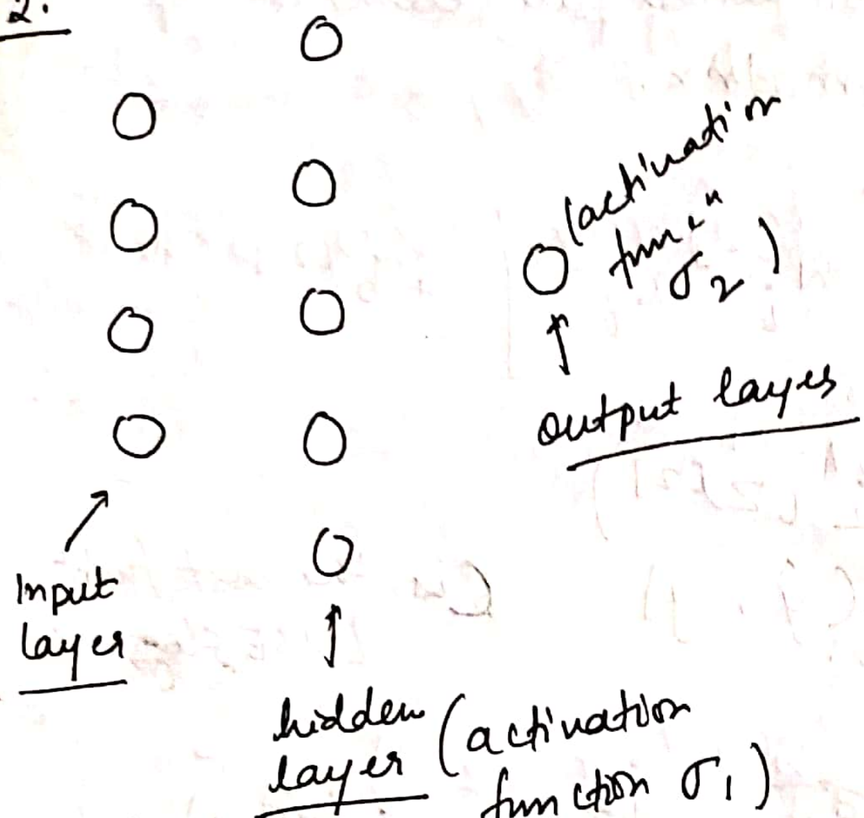
1. Forward Propagation →

Forward propagation refers to the process of calculating the values of all the activation neurons and subsequently, the output of a neural network, for a given set of weights and bias. The term earns its name from the fact that we start with the input values which are feed-forwarded to the next layer and so on, until we reach the last layer.

Backward Propagation:

Backward propagation refers to the process of calculating the gradient of the cost function with respect to a given set of weights and bias. These gradients are then used to optimize the value of weights and bias. The algorithm makes use of chain rule to compute the gradient. The algorithm earns its name from the fact that we compute the gradients layer by layer, starting from the last one and then iterating "backwards".

2.



Nomenclature:

$W^{[1]} \rightarrow 5 \times 4$ weight matrix (joining 1st and 2nd)

$b^{[1]} \rightarrow 5 \times 1$ bias matrix

$W^{[2]} \rightarrow 1 \times 5$ weight matrix joining 2nd & 3rd layer

$b^{[2]} \rightarrow 1 \times 1$ bias matrix

$w_{ij}^{[1]} \rightarrow$ joining jth to ith.

$a^{[1]} \rightarrow 5 \times 1$ activation ~~matrix~~ vector of hidden layer

$x \rightarrow 4 \times 1$ input ~~matrix~~ vector

$\hat{y} \rightarrow 1 \times 1$ output.

$a^{[1]}$ has elements $a_1^{[1]}, a_2^{[1]}, a_3^{[1]}, a_4^{[1]}, a_5^{[1]}$

x has elements x_1, x_2, x_3, x_4.

Forward propagation:

$\cancel{z^{[1]} = \sum_{i=} w_j x_i}$ $\quad \cancel{z_j^{[1]} = \sum_{i=1} w_{ij} \cdot x_i}$

$$z_i^{[1]} = \left[\sum_{j=1}^{4} w_{ij}^{[1]} x_j\right] + b_j^{[1]}$$

$$a_i^{[1]} = \sigma_1 (z_i^{[1]})$$

So, we have obtained all elements of the activation matrix.

Now,

$$z^{[2]} \cancel{y} = \left[\sum_{i=1}^{5} w_i^{[2]} a_i^{[1]}\right] + b^{[2]}$$

$$\hat{y} = \sigma_2(z^{[2]})$$

$$J = \mathcal{C}(\hat{y}, y)$$ $\mathcal{C}$ is the cost funct (MSE, ~~log~~, etc)

<u>Backpropagation:</u>

$$\cancel{\frac{dJ}{dw_i^{[2]}} = \frac{dJ}{d\hat{y}} \frac{d\hat{y}}{dw_i^{[2]}} = c'(\hat{y}, y)}.$$

$$\frac{dJ}{dw_i^{[2]}} = \frac{dJ}{d\hat{y}} \frac{d\hat{y}}{dz^{[2]}} \frac{dz^{[2]}}{dw_i^{[2]}} = \cancel{c'(\hat{y}, y) . \sigma_2'(\hat{y}, y)}$$

$$\boxed{\frac{dJ}{dw_i^{[2]}} = c'(\hat{y}, y) . \sigma_2'(z^{[2]})\, a_i^{[1]}}$$

$$\boxed{\frac{dJ}{db^{[2]}} = \frac{dJ}{d\hat{y}} . \frac{d\hat{y}}{dz^{[2]}} . \frac{dz^{[2]}}{db^{[2]}} = c'(\hat{y}, y) . \sigma_2'(z^{[2]})}$$

Now,

$$x_k \xrightarrow{w_{jk}^{[1]}} z_j \xrightarrow{\sigma_1} a_j^{[1]} \xrightarrow{w_j^{[2]}} z_j^{[2]} \xrightarrow{\sigma_2} \cancel{a}\, \hat{y} \rightarrow J$$

$$\frac{dJ}{dw_{jk}^{(1)}} = \left(\frac{dJ}{d\hat{y}} . \frac{d\hat{y}}{dz^{[2]}}\right)\left(\frac{dz^{[2]}}{da_j^{[1]}}\right)\left(\frac{da_j^{[1]}}{dz_j^{[1]}} \frac{dz_j^{[1]}}{dw_{jk}^{[1]}}\right)$$

$$\boxed{\frac{dJ}{dw_{jk}^{(1)}} = \frac{dJ}{dz^{[2]}} (w_j^{[2]})\, \sigma_1'(z_j^{[1]})\, x_k}$$

$$\boxed{\frac{dJ}{dz^{[2]}} = c'(\hat{y}, y) \cdot \sigma_2'(z^{[2]}) = \frac{dJ}{db^{[2]}}}$$

$$\boxed{\frac{dJ}{db_j^{[1]}} = \frac{dJ}{dz^{[2]}} \cdot w_{0j}^{[2]} \cdot \sigma_1'(z_j^{[1]})}$$

Now, vectorizing them :

$$\boxed{\frac{dJ}{dW^{[2]}} = c'(\hat{y}, y) \cdot \sigma_2'(z^{[2]})(a^{[1]})^T}$$

$$\boxed{\frac{dJ}{db^{[2]}} = c'(\hat{y}, y) \cdot \sigma_2'(z^{[2]}) = \frac{dJ}{dz^{[2]}}}$$

$$\cancel{\frac{dJ}{dW^{[1]}} = (W^{[2]T})} \cdot \frac{dJ}{dz^{[2]}}$$

$$\boxed{\frac{dJ}{dW^{[1]}} = \left(\frac{dJ}{dz^{[2]}}\right)\left((W^{[2]})^T * \sigma_1'(z^{[1]})\right) \times x^T}$$

element wise multiplication

$$\boxed{\frac{dJ}{db^{[1]}} = \left(\frac{dJ}{dz^{[2]}}\right)\left((W^{[2]})^T * \sigma_1'(z^{[1]})\right)}$$

3. (a) Sigmoid:

$$\boxed{\begin{aligned} f(x) &= \frac{1}{1+e^{-x}} \\ f'(x) &= \frac{e^{-x}}{(1+e^{-x})^2} \end{aligned}} = f(x) \times (1 - f(x))$$

(b) ReLU:

$$\boxed{\begin{aligned} f(x) &= \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \\ \Rightarrow f'(x) &= \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \end{aligned}}$$

or $f(x) = max(0, x)$

(c) Leaky ReLU:

$$\boxed{\begin{aligned} f(x) &= \begin{cases} x & x \geq 0 \\ 0.01x & x < 0 \end{cases} \\ f'(x) &= \begin{cases} 1 & x \geq 0 \\ 0.01 & x < 0 \end{cases} \end{aligned}}$$

(d) Tanh:

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

$$\Rightarrow f'(x) = \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$\boxed{= f'(x) = 1 - (f(x))^2}$$

(e) Softmax:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ | \\ x_n \end{bmatrix}$$

$\sigma(\vec{x})$ is also a vector of dimensions

$n \times 1$.

$$\boxed{[\sigma(\vec{x})]_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}}$$

1st case: derivative wrt x_k, $k \neq i$.

$$\therefore \frac{d[\sigma(\vec{x})]_i}{dx_k} = \frac{-e^{x_i} \cdot e^{x_k}}{\left(\sum_{j=1}^{n} e^{x_j}\right)^2} = -[\sigma(\vec{x})]_i [\sigma(\vec{x})]_k$$

2nd case: derivative wrt x_k, $k = i$

$$\therefore \frac{d[\sigma(\vec{x})]_i}{dx_i} = \frac{\left[\left(\sum_{j=1}^{n} e^{x_j}\right) - (e^{x_i})\right](e^{x_i})}{\left(\sum_{j=1}^{n} e^{x_j}\right)^2}$$

$$= [\sigma(\vec{x})]_i \left(1 - [\sigma(\vec{x})]_i\right)$$

$$\therefore \boxed{\frac{d[\sigma(\vec{x})]_i}{dx_k} = \begin{cases} [\sigma(\vec{x})]_i \cdot (1 - \sigma(\vec{x}))_k, & i = k \\ -[\sigma(\vec{x})]_i [\sigma(\vec{x})]_k, & i \neq k \end{cases}}$$

Scanned with CamScanner