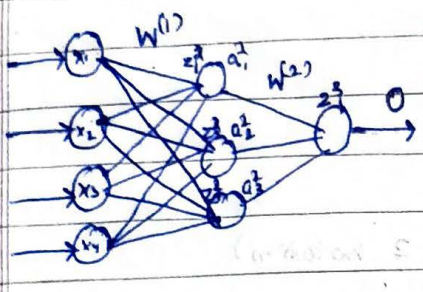
Page No.
Date : / /

Main intuition for extra layer:

→ The layer learns non-linear interactions between the input word vectors.

- **Neural Networks: (Core Understanding)**



Facts

* Neural network is not any real network, just a mathematical abstract idea.

→ In this we take an input vector multiply it with parameter vector and get output

→ Input (is somewhat a hyperparameter); no. of hidden layers are hyper para.
→ parameters or features are learned. (W) — which creates the initial feature
→ there is a loss function at the output of our choice,
→ Each hidden layer has basically 2 parts / function mixed together [adder, non-linearity].

Now we can define notation in 2 ways.

→ either take this → (S) —W— (E) as a row and O —W— O ← this as column

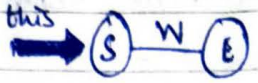
(where S denotes start of a function and E denotes end of a function)

but then function will be written as

$$f(W^T x + b)$$

Page No.
Date : / /

Another way, which is straight forward thus used every where is :-



as column

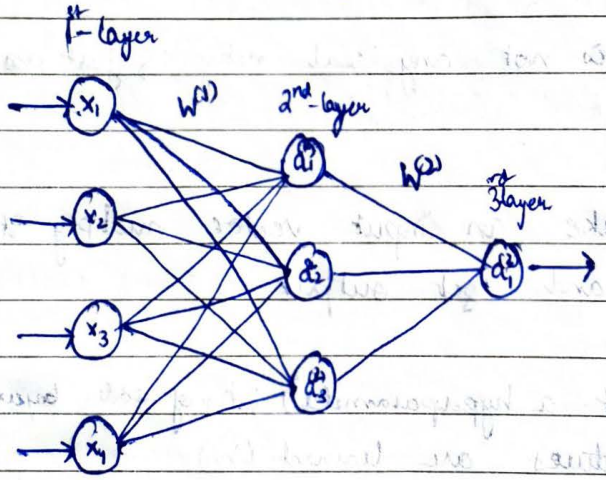
as row.

(where S denotes start of a connection & E denotes end of a connection)

thus, now the functions have a nice look of

$$a = f(Wx + b)$$

Now, in matrix outlay (using 2 notation)



Here vector x is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Here matrix dimension of parameters W are :-

* since end units of 1st hidden layer are $a^{(1)}$ thus row is 3, and start is input layer thus column is 4. $\mathbb{R}^{3\times4}$

$$W^{(1)} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix}_{3\times4}$$

Page No.
Date : / /

Similarly,

End of connection is $a^{(2)}$ hence row is 1, and start of connection is $a^{(1)}$ hence columns is 3.

$$W^{(2)} = \underset{1\times 3}{\begin{bmatrix} w_{11} & w_{12} & w_{13} \end{bmatrix}}$$

Here we have 2 input vector (one is x that we give), other is $a^{(1)}$ that we form,

* Dry Implementation

$$z^{(2)} = \underset{3\times 4}{\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix}} \underset{4\times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}$$

$$z^{(1)} = \underset{3\times 1.}{\begin{bmatrix} (w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{14}x_4) \\ (w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + w_{24}x_4) \\ (w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + w_{34}x_4) \end{bmatrix}}$$

{ where $z^{(1)}$ is pre-activation / added function }

$$a^{(1)} = \underset{3\times 1.}{\begin{bmatrix} f(w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{14}x_4) \\ f(w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + w_{24}x_4) \\ f(w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + w_{34}x_4) \end{bmatrix}}$$

{ where $f(x)$ is a non-linearity func }

Now $a^{(1)}$ becomes input for (2) layer.

NOTE : output of each layer is essentially a vector.

Page No.
Date : / /

Now,

$$z^{(3)} = w^{(2)} a^{(2)}$$

$$z^{(3)} = \underset{1\times 3}{\begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \end{bmatrix}} \underset{3\times 1}{\begin{bmatrix} f(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 + w_{14}^{(1)} x_4) \\ f(w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{23}^{(1)} x_3 + w_{24}^{(1)} x_4) \\ f(w_{31}^{(1)} x_1 + w_{32}^{(1)} x_2 + w_{33}^{(1)} x_3 + w_{34}^{(1)} x_4) \end{bmatrix}}$$

$$z^{(3)} = \underset{1\times 1}{\begin{bmatrix} w_{11}^{(2)} f(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 + w_{14}^{(1)} x_4) + w_{12}^{(2)} f(w_{21} x_1 + w_{22} x_2 + w_{23} x_3 \\ + w_{24} x_4) + w_{13} f(w_{31} x_1 + w_{32} x_2 + w_{33} x_3 + w_{34} x_4) \end{bmatrix}}$$

$$a^{(3)} = \begin{bmatrix} f(w_{11}^{(2)} f(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 + w_{14}^{(1)} x_4) + w_{12}^{(2)} f(w_{21} x_1 + w_{22} x_2 \\ w_{24} x_4) + w_{13} f(w_{31} x_1 + w_{32} x_2 + w_{33} x_3 + w_{34} x_4) \end{bmatrix}$$

Output $(o) = a^{(3)}$

Notes

1. Size (or dimension) of parameter matrix : (no. of units in ending connection layer) (no. of unit in starting connection layer)

$$\mathbb{R}^{(\text{unit in ending layer}) \times (\text{unit in starting layer})}$$

$$\rightarrow \mathbb{R}^{(s_{j+1} \times s_j)}$$

Page No.
Date : / /

Backpropagation in this network

* first of all we will design a cost function for the last output layer

→ let's take it as

$$l = \frac{1}{2}(\hat{y} - y)^2$$ where $\hat{y}$ is the output

we know $\frac{\partial l}{\partial \hat{y}} = \hat{y} - y$

Now, find

$$\frac{\partial l}{\partial w_{ij}^{(1)}} = \frac{\partial l}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_1^{(2)}} \left(\frac{\partial z_1^{(2)}}{\partial a_i^{(1)}} \cdot \frac{\partial a_i^{(1)}}{\partial z_i^{(1)}} \right) \frac{\partial z_i^{(1)}}{\partial w_{ij}^{(1)}}$$

$\delta_i^{(2)}$

$$\frac{\partial \hat{y}}{\partial z_1^{(2)}} = \frac{\partial (a_1^{(2)})}{\partial z_1^{(2)}} = f'(z_1^{(2)})$$

$$\frac{\partial z_i^{(2)}}{\partial a_i^{(1)}} = w_i^{(2)}$$ { important : $\frac{\partial z_i^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} = \delta^{(2)}$ }

$$\frac{\partial a_i^{(1)}}{\partial z_i^{(1)}} = f'(z_i^{(1)})$$ { here $f(z_i) = \sigma(z_i)$ $\Rightarrow f'(z_i) = f(z_i)(1 - f(z_i))$ }

$$\frac{\partial (z_i^{(1)})}{\partial w^{(1)}} = \frac{\partial (w^{(1)} x + b)}{\partial w^{(1)}} = \frac{\partial (w_{i1}^{(1)} x_1 + w_{i2}^{(1)} x_2 + w_{i3}^{(1)} x_3 + w_{i3}^{(1)} x_3 + w_{i4}^{(1)} x_4 + b)}{\partial w^{(1)}}$$

$$\frac{\partial (z_i^{(2)})}{\partial w^{(1)}} = \frac{\partial (\sum_k w_{ki} x_i + b_i)}{\partial w_{ij}^{(1)}}$$

$$\frac{\partial (z_i^{(2)})}{\partial (w_{ij}^{(1)})} = x_j$$ (x_j signifies, we choose j^{th} for w as choosing i it in turn chooses z_i and choosing j takes out the j^{th} element out of z_i

Page No.
Date : / /

Substitute in eqn to get gradient

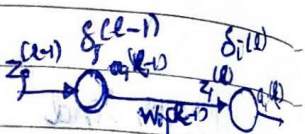
then update gradient wrt:-

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial L}{\partial w_{ij}} \quad (sgd)$$

η : learning rate (hyperparameter)

So in general, from equation we can see that, error received at $a_j^{(l-1)}$

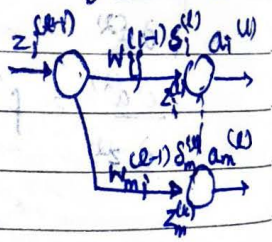
$$\delta_j^{(l-1)} = \delta_i^{(l)} w_{ij}^{(l-1)}$$



(derivative of j^{th} unit in $(L-1)$ layer received at activation layer a_j) = (derivative of j^{th} unit 'connecting' to $L-1$ & L) × (weight connecting)

If there are multiple units in L layer connecting to j^{th} unit of $L-1$ layer then:

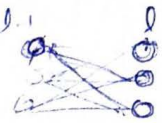
$$\delta_j^{(l-1)} = \sum_i \delta_i^{(l)} w_{ij}^{(l-1)}$$



Now propagating this $z_j^{(l-1)}$,

$$z_j^{(l-1)} = f'(z_j^{(l-1)}) \sum_i \delta_i^{(l)} w_{ij}^{(l-1)}$$

Now this error goes back to other units thus this is denoted as $\delta_j^{(l-1)}$



Page No.
Date : / /

$\Rightarrow \delta_j^{(l-1)} = f'(z_j^{(l-1)}) * \sum_i \delta_i^{(l)} w_{ij}^{(l-1)}$

Obtain a matrix of $\delta_j^{(l-1)}$ that maps every j unit to i^{th}

$\nabla W^{(l)} = \delta^{(l+1)} a^{(l)T}$

$\delta^{(l+1)} \rightarrow S_{l+1} \times 1$
$a^{(l)} \rightarrow 1 \times S_l$
$\nabla W^{(l)} \rightarrow S_{l+1} \times S_l$
Weights, $W^{(l)} \rightarrow S_{l+1} \times S_l$

$\delta^{(l)} = f'(z^{(l)}) \circ \left(W^{(l)T} \cdot \delta^{(l+1)}\right)$

$$\nabla_{W^{(l)}} = \begin{bmatrix} \delta_1^{(l+1)} \\ \vdots \\ \delta_{S(l+1)}^{(l+1)} \end{bmatrix}_{S(l+1)\times 1} \begin{bmatrix} a_1^{(l)} & a_2^{(l)} & \cdots & a_{S(l)}^{(l)} \end{bmatrix}_{1 \times S(l)}$$

$$\Rightarrow \nabla_{W^{(l)}} = \begin{bmatrix} \delta_1^{(l+1)} a_1^{(l)} & \delta_1^{(l+1)} a_2^{(l)} & \cdots & \delta_1^{(l+1)} a_{S(l)}^{(l)} \\ \delta_2^{(l+1)} a_1^{(l)} & \delta_2^{(l+1)} a_2^{(l)} & \cdots & \delta_2^{(l+1)} a_{S(l)}^{(l)} \\ \vdots & & & \vdots \\ \delta_{S(l+1)}^{(l+1)} a_1^{(l)} & \delta_{S(l+1)}^{(l+1)} a_2^{(l)} & \cdots & \delta_{S(l+1)}^{(l+1)} a_{S(l)}^{(l)} \end{bmatrix}_{S(l+1)\times S(l)}$$

where $s(l+1)$ is size of layer $(l+1)$
& $s(l)$ is size of layer (l)

$W \leftarrow W - \alpha(\nabla W)$

$\{ W_{S(l+1)\times S(l)} \leftarrow W_{S(l+1)\times S(l)} - \alpha(\nabla W_{S(l+1)\times S(l)}) \}$

$\delta^{(l)} = f'(z^{(l)}) \circ \left(W^{(l)T} \cdot \delta^{(l+1)}\right)$

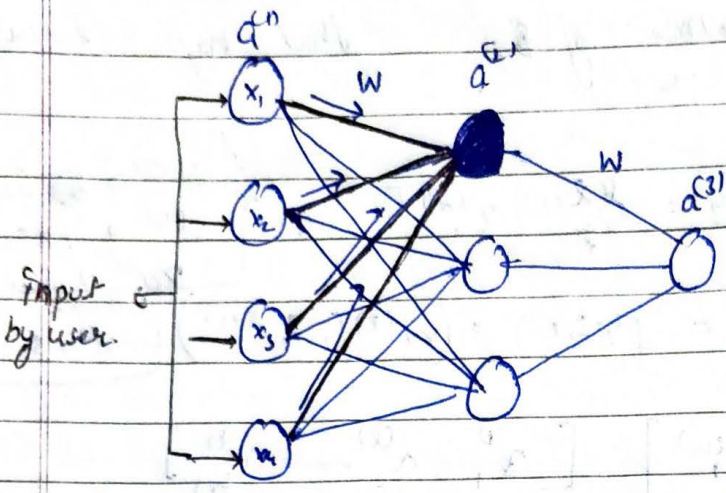
dimension

$W^{(l)T} \rightarrow S(l) \times S(l+1)$
$\delta^{(l+1)} \rightarrow S(l+1) \times 1$

$\delta^{(l)} \rightarrow S(l) \times 1$

Page No.
Date : / /

Visualization of feed-forward propagation



In forward propagation, we select one unit of forward layer,

here $a_1^{(2)}$ (coloured)

take all the weights connecting to that layer from previous layer.

$$W_{1j}^{(1)}$$

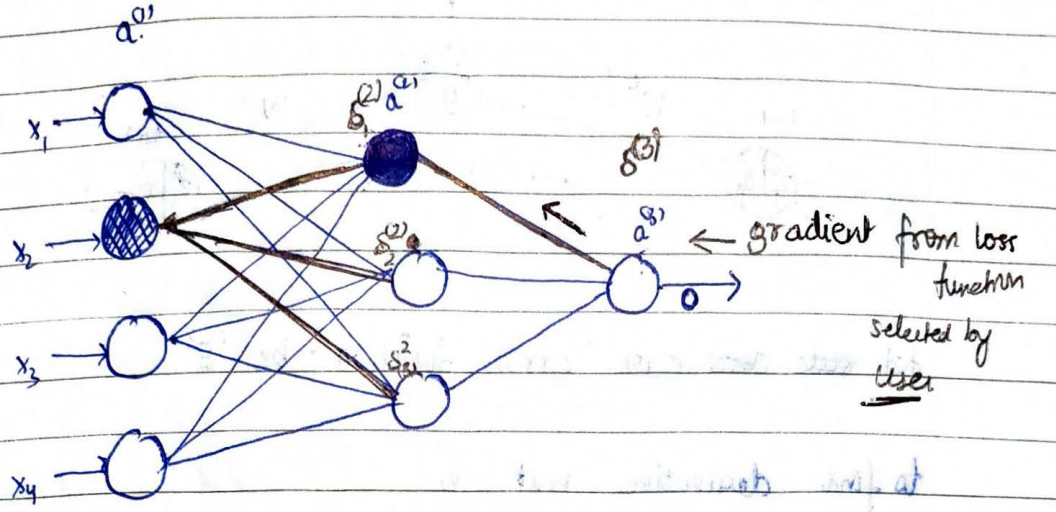
and calculate 'pre-activation' at that layer by:

$$z_1^{(2)} = \sum_j W_{1j} x_j$$

and $a_1^{(2)} = \sigma\left(\sum_j W_{1j} x_j\right)$ {where $\sigma(x)$ is non-linearity}

Page No.
Date : /

Visualization of back-prop



In backward propagation, we go from output to input layer propagating 'δ' (gradients)

$$\delta^{(l)} = \left(W^{(l)^T} \delta^{(l+1)}\right) \odot f'(z^{(l)}) \quad \{l \to L-1 \text{ to } 1\}$$

where, $\delta^{(last)}$ is calculated before by the help of cost function

Here, the selected, $a_1^{(2)}$

$$\delta_1^{(2)} = \delta^{(3)} \cdot w_{11}^{(2)} \cdot f'(z_1^{(2)})$$

$$\nabla_{W^{(l)}} = \delta^{(l+1)} \cdot a^{(l)^T}$$

{ outer product : no dimension in expanding
$(s(l+1) \times 1) \times (1 \times s(l)) \to s(l+1) \times s(l)$ }

Now, propagating ahead let's consider another unit $a_2^{(1)}$

$$\delta_2^{(1)} = \sum_i^{s(2)} \delta_i^{(2)} \cdot w_{i2}^{(1)} \cdot f'(z_2^{(1)})$$

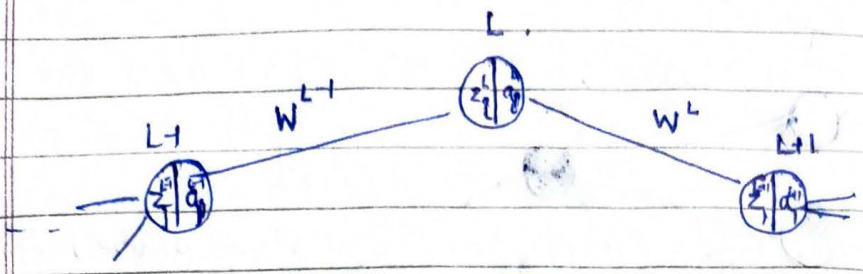
$$\nabla_{W^{(1)}} = \delta^{(2)} \cdot a^{(1)^T}$$

$$\nabla W_{12} = \delta_1^{(2)} x_2$$

↓ just for W_{12}

Page No.
Date : / /

- How to get generalized gradients ?



Let our error function be 'E'.

to find derivative wrt. W^{L-1}

$$\frac{\partial E}{\partial w_{ij}^{L-1}} = \frac{\partial E}{\partial a_i^L} \frac{\partial a_i^L}{\partial w^{L-1}}$$

$$= \left(\frac{\partial E}{\partial a_i^L} \frac{\partial a_i^L}{\partial z_i^L} \right) \frac{\partial z_i^L}{\partial w_{ij}^{L-1}}$$

δ^L → δ^L because every term in L

we know;

$$a_i^L = f(z^L) \quad \{ f(x) \text{ is a non-linearity func} \}$$

$$\Rightarrow \frac{\partial (a_i^L)}{\partial z_i^L} = f'(z_i^L)$$

$$\frac{\partial (z_i^L)}{\partial w_{ij}^{L-1}} = \frac{\partial \left(\sum_j w_{ij} a_j^{L-1} \right)}{\partial w_{ij}^{L-1}} = a_j^{L-1}$$

$$\frac{\partial E}{\partial w_{ij}^{L-1}} = \frac{\partial E}{\partial a_i^L} \cdot f'(z_i^L) \cdot a_j^{L-1}$$

$$\frac{\partial E}{\partial w_{ij}^{L-1}} = \underbrace{\frac{\partial E}{\partial a_i^L} f'(z_i^L)}_{\delta^L} a_j^{L-1} = \delta^L a_j^{L-1}$$

Page No.
Date : / /

$$\delta_i^L = \frac{\partial E}{\partial a_i^L} \cdot f'(z_i^L)$$

$$\frac{\partial E}{\partial a_i^L} = \frac{\partial E}{\partial a_i^{L+1}} \frac{\partial a_i^{L+1}}{\partial a_i^L}$$

$$\frac{\partial E}{\partial a_i^L} = \left(\underbrace{\frac{\partial E}{\partial a_i^{L+1}} \frac{\partial a_i^{L+1}}{\partial z_i^{L+1}}}_{\delta^{L+1}} \right) \left(\frac{\partial z^{L+1}}{\partial a_i^L} \right)$$

$$\delta_i^L = \delta^{L+1} \left(\frac{\partial z^{L+1}}{\partial a_i^L} \right) \left(f'(z_i^L) \right)$$

$$\delta_i^L = \delta^{L+1} \left(\frac{\partial (W^L a^L)}{\partial a_i^L} \right) f'(z_i^L)$$

$$\delta_j^L = \sum_i \delta_i^{L+1} W_{ij}^L \cdot f'(z_i^L)$$

$$\frac{\partial E}{\partial W_{ij}^{(l)}} = \delta_j^{(l+1)} a_i^l$$

where

$$\delta_j^L = \sum_i \delta_i^{L+1} W_{ij}^L f'(z_i^L)$$

↳ can be computed before with forward prop

For vectorization:

$$\nabla_{W^{(l)}} \rightarrow \delta^{(l+1)} a^{(l)T} + \underbrace{\lambda W^{(l)}}_{\text{regularization term}} \quad \{ s(l+1) \times s(l) \rightarrow (s(l+1) \times 1) \times (1 \times s(l)) \}$$

$$\delta^L = \left((W^L)^T \delta^{L+1} \right) \odot \{ f'(z_i^L) \} \quad \{ s(l) \rightarrow (s(l) \times s(l+1)) \times (s(l+1) \times 1) \}$$

Page No.
Date : / /

- Why go through derivations?

1- ~~Acd~~ Actual understanding of math behind most of deep learning

2- Backprop can be an imperfect abstraction by — issues such as vanishing gradient

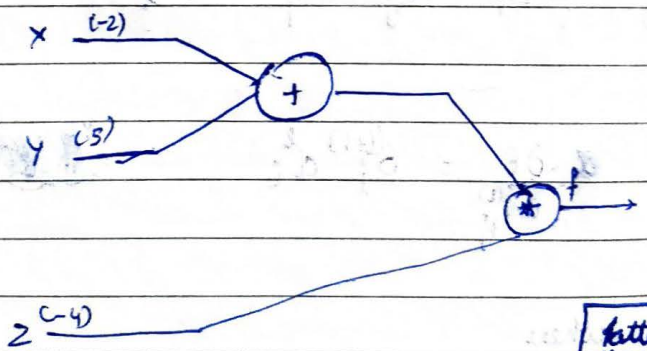
3 Enables you to debug models, think of and implement completely new models that aren't yet supported by any framework.

(Giving general idea)

- Explanation of Backprop: Circuit

Function as circuits

$$f(x,y,z) = (x+y)z \quad ; \quad x=-2, y=5, z=-4$$



$$q = x+y \qquad \frac{\partial q}{\partial x} = 1 \; ; \; \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z \; ; \; \frac{\partial f}{\partial z} = q$$

factoring in backward flow

add gate : gradient distributor

max gate : gradient router

multiply gate : gradient switcher

max gate = max(x,y)

suppose x > y

Page No.
Date : / /

Want : $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

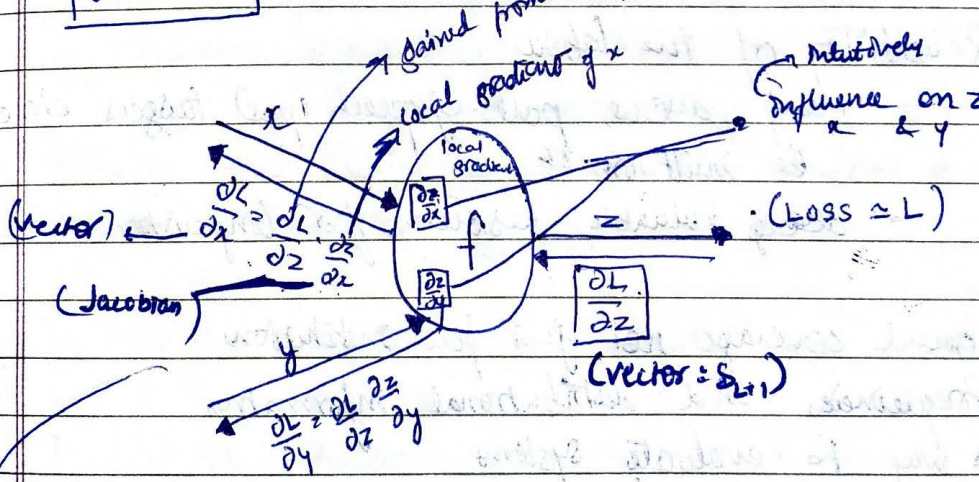
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

- $\boxed{\frac{\partial f}{\partial x} = z}$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

- $\boxed{\frac{\partial f}{\partial y} = z}$

- $\boxed{\frac{\partial f}{\partial z} = q}$



- Another example { In lecture 5 ; similar to circuit ex }

I (compute local gradient / derivative)
II Calculate the derivative with the value obtained in forw. prop
III To completely compute the gradient of function :
(I) * (gradient of the successive layer)