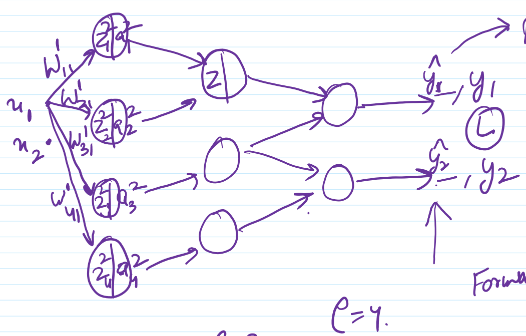
26_Backpropagation_in_CNN_1

14 October 2024 09:29

Simple Neural Networks.



y_1, y_2 → true outputs.

z^ℓ_i / a^ℓ_i

before / after activation.

$$z^2_i = \sum_{j=1}^{2} w_{ij} x_j + b^1_i, \quad i = 1, 2, \ldots 4$$

$$a^2_i = f(z^2_i)$$

$$z^3_i, \quad a^3_i$$

$$z^4_i, \quad a^4_i = \hat{y}_i \qquad i = 1, 2$$

$$L = \frac{1}{2} \sum_{i=1}^{2} (\hat{y}_i - y_i)$$

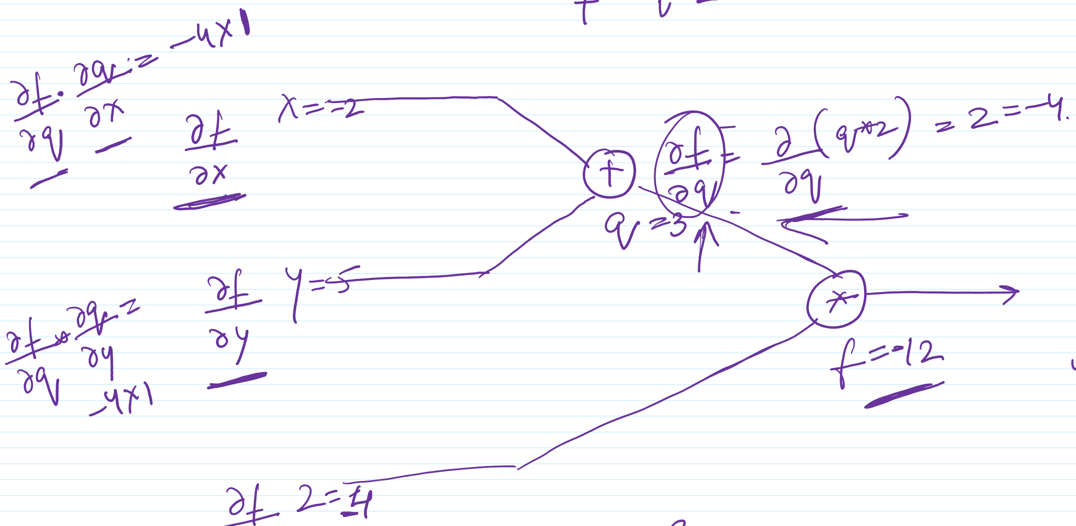
Chain Rule:-

$$f(x, y, z) = (x+y)z$$

Let $q = x+y$

$f = q * z$

$x = -2$
$y = 5$
$z = -4$



"Chain Rule"

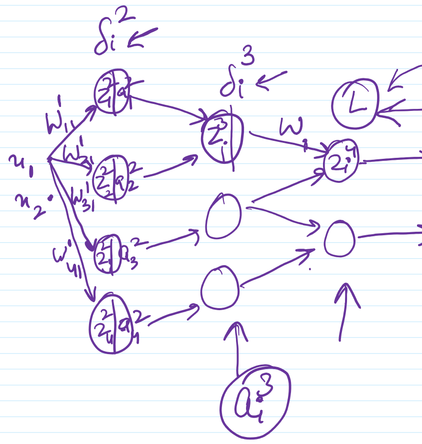
$$\frac{\partial f}{\partial z} = \frac{\partial (q * z)}{\partial z} = q = 3$$

$\partial 2 \quad \partial 2$

$$\frac{\partial L}{\partial w_{ij}^3} = \underbrace{\frac{\partial L}{\partial z_i^4}}_{\delta_i^4} \cdot \frac{\partial z_i^4}{\partial w_{ij}^3}$$

$$\frac{\partial L}{w_{ij}^2} = \underbrace{\frac{\partial L}{\partial z_i^3}}_{\delta_i^3} \cdot \frac{\partial z_i^3}{\partial w_{ij}^2}$$

$$\frac{\partial L}{w_{ij}^1} = \underbrace{\frac{\partial L}{\partial z_i^2}}_{\delta_i^2} \cdot \frac{\partial z_i^2}{\partial w_{ij}^2}$$



$$\boxed{L = \frac{1}{2}\left(\sum_{j=1}^{2}(\hat{y}_i - y_i)\right)^2}$$ ← loss. (MSE)

$$\boxed{\delta_i^4 = \frac{\partial L}{\partial z_i^4} = (\hat{y}_i - y_i) \cdot \frac{\partial \hat{y}_i}{\partial z_i^4}}$$

$$\delta_i^3 = \frac{\partial L}{\partial z_i^3} = \underbrace{\frac{\partial L}{\partial a_i^3}} \cdot \underbrace{\frac{\partial a_i^3}{\partial z_i^3}}_{f'(z_i^3)}$$

$$\frac{\partial L}{\partial a_i^3} = \sum_{j=1}^{2} \frac{\partial L}{\partial z_j^4} \cdot \frac{\partial z_j^4}{\partial a_i^3}$$

# of neurons in $(\ell+1)^{th}$ layer

$$\underbrace{\delta_i^\ell}_{\text{error}} = \sum_{j=1}^{s_{\ell+1}} \left(\delta_j^{\ell+1} \cdot w_{ji}^\ell\right) \cdot f'(z_i^\ell)$$

Backpropagation in CNN:— stride = 2, grayscal image $C=1$, # of con... fil... Output...

a filter $W^{k_1 \times k_2}$, input $X^{N_1 \times N_2}$, output $Y^{M_1 \times M_2}$

$k_1-1 \quad k_2-1$

$\hat{y}_1$, y_1

L

$\hat{y}_2$, y_2

volutional

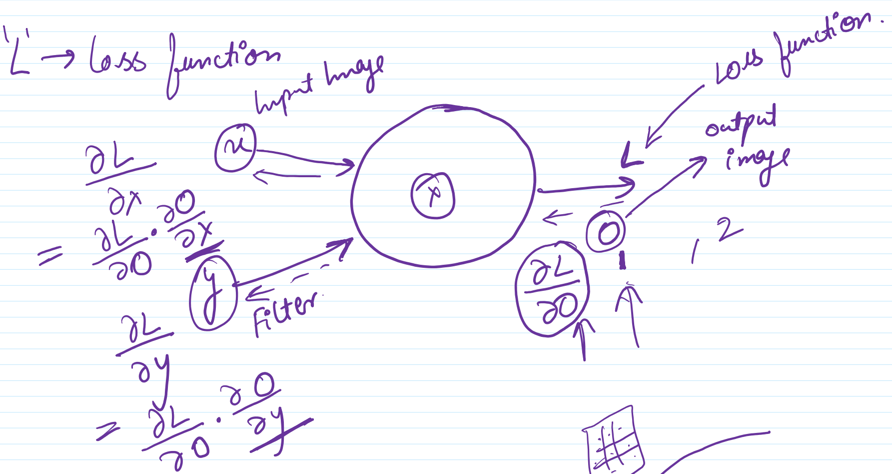
lters are 1

t channel = 1

$$Y[i,j] = \sum_{a=0} \sum_{b=0} X[i-a, j-b]\, W[a,b]$$

⇒ not centering the convolution at a pixel here, but placing the filter at a corner of the window.

'L' → Loss function



$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} * \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix}$$

$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + F_{22}F_{22}$

$O_{12} = X_{21}$ ________________

$O_{21} =$ ________________

$O_{22} =$ ________________

$$\underbrace{\frac{\partial O}{\partial X}, \frac{\partial O}{\partial F}}_{\partial} \longrightarrow L$$

★★ $\frac{\partial L}{\partial F}$ is convolution between input X and Loss Gradient from the next layer $\frac{\partial L}{\partial O}$

★★ $\frac{\partial L}{\partial X}$ is convolution between 180° rotated F and Loss Gradient from the next layer $\frac{\partial L}{\partial O}$

ocal gradients