Backpropagation

$y(i)$, $\hat{y}(i)$ each ~~are~~ is vector.

L layers

k_0 nodes in input layer $\Rightarrow \vec{x}(i) = [x_1(i), \ldots, x_{k_0}(i)]^T$

k_r nodes in rth layer

sigmoid activation function $\Rightarrow f(x) = \frac{1}{1+\exp(-ax)}$

$(y(i), x(i)) \Rightarrow N$ training pairs

K_L output neurons $\Rightarrow \vec{\hat{y}}(i) = [\hat{y}_1(i) \ldots \hat{y}_{K_L}(i)]^T$

Cost function, J

$\vec{w}_j^r \Rightarrow$ weight vector of jth neuron of rth layer

$\hookrightarrow$ dimension $(k_{r-1}+1)$ $\hookrightarrow$ threshold included

$\therefore w_j^r = [w_{j0}^r, w_{j1}^r, \ldots, w_{jk_{r-1}}^r]^T$

$w_j^r(\text{new}) = w_j^r(\text{old}) + \Delta w_j^r$

$\Delta w_j^r \hookrightarrow -\mu \frac{\partial J}{\partial w_j^r}$

$v_j^r \Rightarrow \Sigma$ of jth neuron, rth layer (weighted result)

$y_j^r \Rightarrow$ activation(v_j^r)

$J = \sum_{i=1}^{N} \varepsilon(i)$ ~~$\frac{1}{2}$~~

$\varepsilon(i) = \frac{1}{2} \sum_{m=1}^{k_L} e_m^2(i) = \frac{1}{2} \sum_{m=1}^{k_L} (y_m(i) - \hat{y}_m(i))^2,$

$i = 1, 2, \ldots, N$

Scanned by CamScanner

$$v_j^r(i) = \sum_{k=0}^{k_{r-1}} w_{jk}^r \, y_k^{r-1}(i)$$

$y_0^r(i) = +1$ for all layers.

For $r = L$ (output layer) $\Rightarrow$ $y_k^r(i) = \hat{y}_k(i)$
$\quad \hookrightarrow$ for ith example

for $r = 1$,

$$y_k^{r-1}(i) = x_k(i), \quad k = 1, 2, \ldots k_0$$

অর্থাৎ r_0 হলো input layer

$$\frac{\partial \mathcal{E}(i)}{\partial w_j^r} = \frac{\partial \mathcal{E}(i)}{\partial v_j^r(i)} \cdot \frac{\partial v_j^r(i)}{\partial w_j^r}$$

let, $\delta_j^r(i)$

(একটা node এর আর weight এর জন্য এটা same)

$$y^{r-1}(i) = \begin{bmatrix} +1 \\ y_1^{r-1}(i) \\ y_2^{r-1}(i) \\ \vdots \\ y_{k_{r-1}}^{r-1}(i) \end{bmatrix}$$

$$\therefore \Delta w_j^r = -\mu \left(\sum_{i=1}^{N} \delta_j^r(i) \, y^{r-1}(i) \right)$$

we have to compute this. The weight update is for all samples once.

Scanned by CamScanner

$O \to \boxed{\Sigma} \to$ result

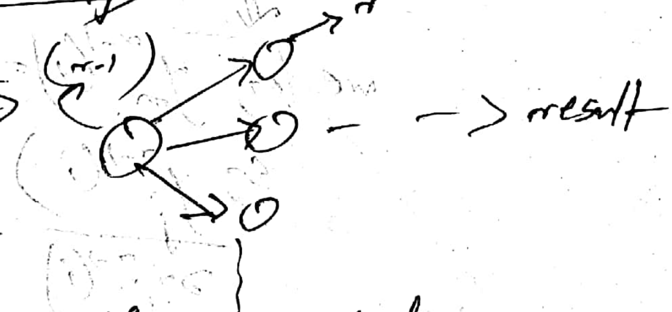
For $r=L$

$$\delta_j^L(i) = \frac{\partial E(i)}{\partial v_j^L(i)} \qquad E(i) \to \frac{1}{2}\sum_{m=1}^{k_L} e_m^2(i)$$

$$\therefore \delta_j^L(i) = \underbrace{e_j(i)}_{\boxed{y_j(i) - \hat{y}_j(i)}} \underbrace{f'(v_j^L(i))}_{\frac{\partial}{\partial v_j^L}(e_j(i))} \;??$$

desired $\leftarrow y_j(i)$, $\hat{y}_j(i) \to$ determined

For $r<L$ $\Longrightarrow$



So, let $(r-1)$ is the present layer.

$$\frac{\partial E(i)}{\partial v_j^{r-1}(i)} = \sum_{k=1}^{k_r} \frac{\partial E(i)}{\partial v_k^r(i)} \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\to \delta_j^{r-1}(i) = \sum_{k=1}^{k_r} \underbrace{\delta_k^r(i)}_{\text{we know this, because coming from last to first.}} \frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)}$$

$$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = \frac{\delta\left[\sum_{m=0}^{k_{r-1}} w_{km}^r \, y_m^{r-1}(i)\right]}{\delta v_j^{r-1}(i)}$$

only $y_j^{r-1}(i)$ is dependent on $v_j^{r-1}(i)$

so, $$\frac{\partial v_k^r(i)}{\partial v_j^{r-1}(i)} = w_{kj}^r \, \underline{\underline{f'(v_j^{r-1}(i))}}$$

$\Downarrow$

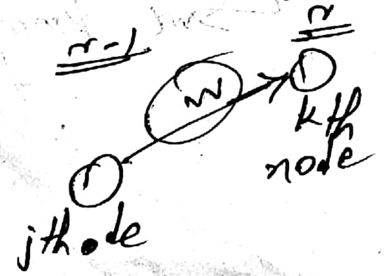
we let, $y_m^{r-1}(i) = f(v_m^{r-1}(i))$

so, $$\frac{y_j^{r-1}(i)}{\partial v_j^{r-1}(i)} = f'(v_j^{r-1}(i))$$

$$\therefore \delta_j^{r-1}(i) = \left[\sum_{k=1}^{k_r} \delta_k^r(i) w_{kj}^r\right] \underline{\underline{f'(v_j^{r-1}(i))}}$$

$\hookrightarrow$ [only $f(v_j^{r-1}(i))$ comes into calculation of all $y_j^r(i)$ of next layer]

w_{kj}^r means



activation function, $f(x) = \frac{1}{1+\exp(-ax)}$

$$f'(x) = af(x)(1-f(x))$$

we shall use this formula in

$f'(v_j^{r-1}(i))$ => this type things

Like this

~~bias there~~ ~~we need to store~~

(1) For each layer, each node, their weights (input excluded)

(2) v_j^r for each node of each layer → (" ")

(3) y_j^r " " " " " " → (input included)

(4) For each input example, each layer, each node, their [$\delta_j^r(i)$ $y^{r-1}(i)$] [input excluded]

(5) when working with each example, store for each layer, each node from $r=2$ to L $\delta_j^r(i)$ for that example.

Scanned by CamScanner