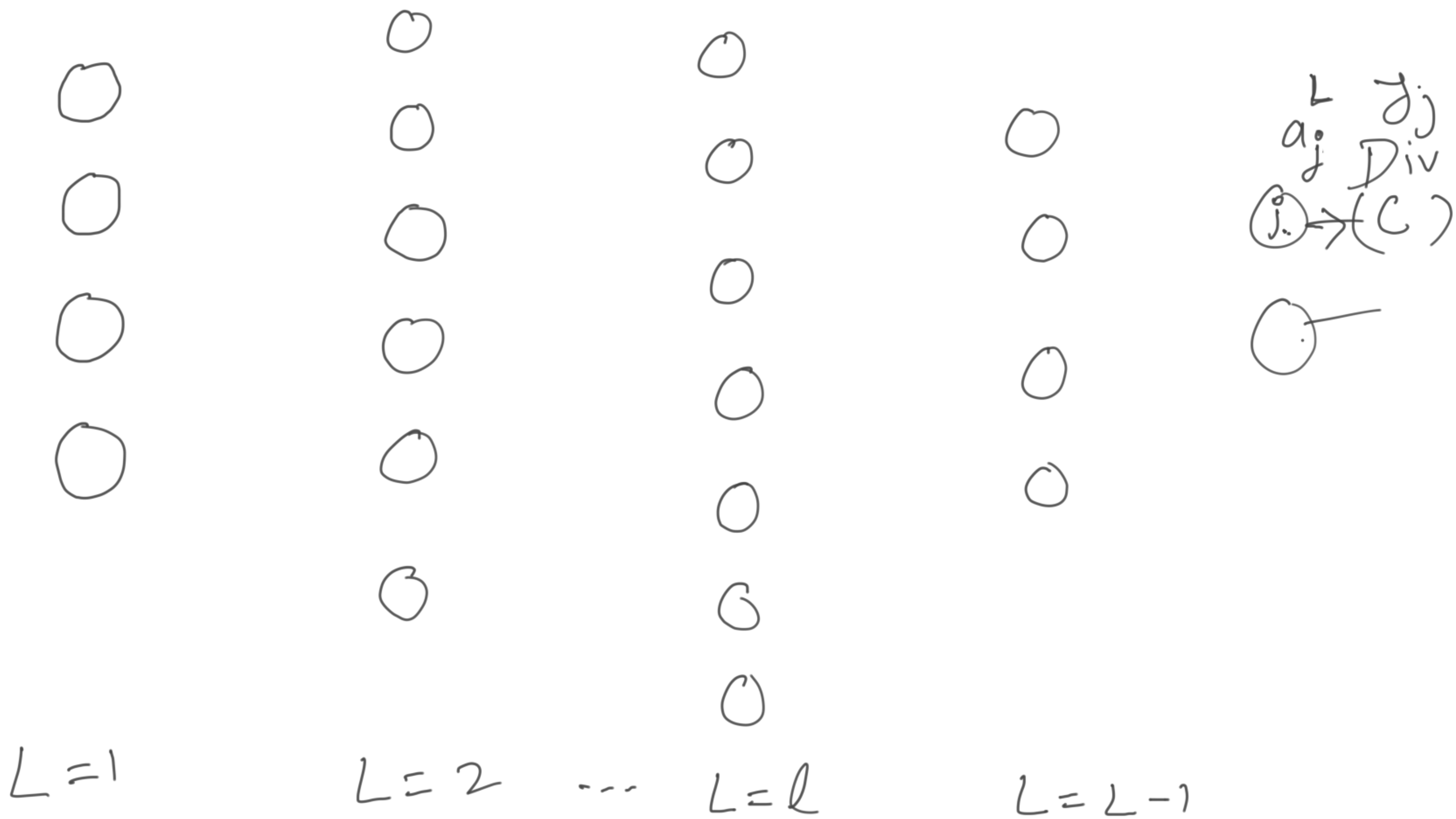
Backpropagation

$$W^* = \underset{W}{\arg\min} \sum_{i=1}^{n} \underline{\underline{Div}}(F(x_i, w), \hat{y}_i)$$

Gradient descent

- Computes gradient at W^0

$w^{t+1} \leftarrow w^t - \eta \nabla Div()$ — Move in $-$ve direction of gradient


L
a_j
y_j
Div
j
C
L=1
L=2
...
L=l
L=L-1

$$C = \frac{1}{2} \| y - a^L \|^2$$

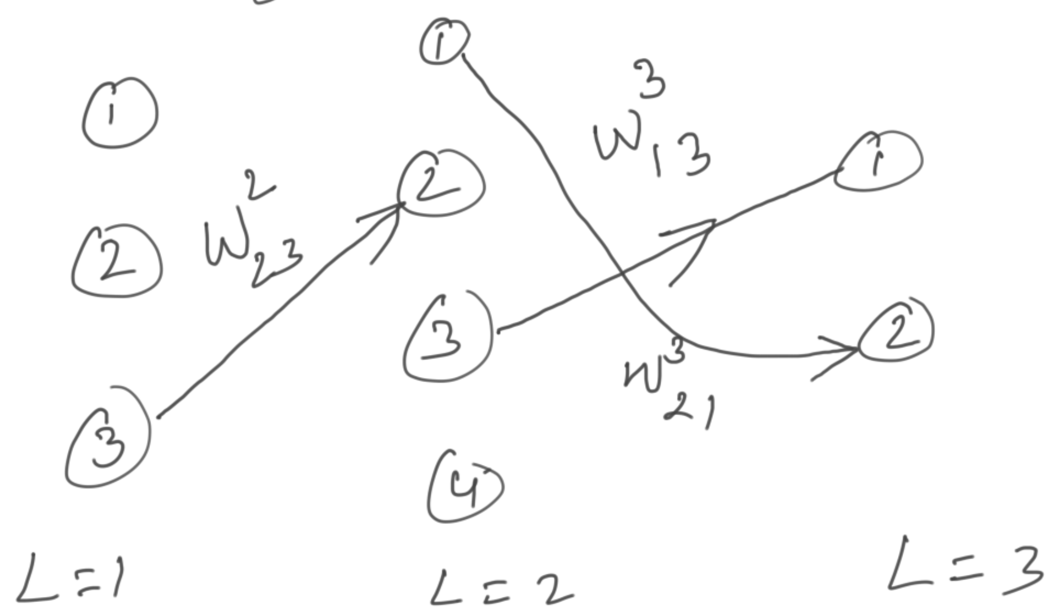
output of the L^{th} layer $= a^L$

$$= \frac{1}{2} \sum_{j=1}^{u} \left(y_j - a_j^L \right)^2$$

output has u neuron.

Goal: We want to compute gradient of C w.r.t. weights.

$w_{jk}^{l} \rightarrow$ incoming weight to j^{th} neuron of l^{th} layer from k^{th} neuron of $(l-1)^{th}$ layer.



$a_j^l \rightarrow$ Activation (output) of j^{th} neuron in l^{th} layer

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

$$= \sigma\left(z_j^l\right)$$

$$\boxed{z_j^l = \sum_k w_{kj}^l a_k^{l-1} + b_j^l}$$

$$\frac{\partial C}{\partial w_{jk}^{l}}, \quad \frac{\partial C}{\partial b_{j}^{l}} \quad l = 1, 2, \ldots, L$$

$$\frac{\partial C}{\partial w_{jk}^{L}} = \frac{\partial}{\partial w_{jk}^{L}}\left(\frac{1}{2}\left(y_j - a_j^L\right)^2\right)$$

$$= \frac{1}{2}\frac{\partial}{\partial a_j^L}\left(y_j - a_j^L\right)^2 \cdot \frac{\partial a_j^L}{\partial w_{jk}^L}$$

$$\left[\frac{\partial}{\partial x} t^2 = \frac{\partial}{\partial t} t^2 \cdot \frac{\partial t}{\partial x}\right]$$

$$= \frac{1}{\cancel{2}} \cancel{2} (y_j - a_j^L) \times (-1) \times \frac{\partial a_j^L}{\partial w_{jk}^L}$$

$$= (a_j^L - y_j) \times \frac{\partial a_j^L}{\partial w_{jk}^L}$$

$$= (a_j^L - y_j) \times \frac{\partial}{\partial w_{jk}} \sigma(z_j^L)$$

$$= \underline{(a_j^L - y_j)\, \sigma'(z_j^L)} \cdot \frac{\partial z_j^L}{\partial w_{jk}^L}$$

$$\frac{\partial C}{\partial w_{jk}^L} = (a_j^L - y_j) \cdot \sigma'(z_j^L) \frac{\partial}{\partial w_{jk}^L} \left(\sum_k w_{jk}^L a_k^{L-1} + b_j \right)$$

$$\left[z_j^L = \sum_k w_{jk}^L a_k^{L-1} + b_j^L \right]$$

$$= \boxed{(a_j^L - y_j)\ \sigma'(z_j^L)}\, a_k^{L-1}$$

$$= \delta_j^L \times a_k^{L-1}$$

$$\frac{\partial C}{\partial w_{jk}^{L-1}} = \frac{\partial}{\partial w_{jk}^{L-1}} \frac{1}{2}(y_j - a_j^L)^2$$

$$= \frac{\frac{\partial}{\partial a_j^L} \frac{1}{2}(y_j - a_j^L)^2}{} \cdot \frac{\partial a_j^L}{\partial w_{jk}^{L-1}}$$

$$= (a_j^L - y_j) \frac{\partial a_j^L}{\partial w_{jk}^{L-1}}$$

$$= (a_j^L - y_j) \frac{\partial}{\partial w_{jk}^{L-1}} \sigma\left(\sum_k w_{jk}^{\ell} a_k^{\ell-1} + b_j^{\ell}\right)$$

$$= \underline{(a_j - y_j)\,\sigma'(z_j^{\ell})}\,\frac{\partial}{\partial w_{jk}^{L-1}}\left(\sum_k w_{jk}^{L} a_k^{L-1} + b_j^{L}\right)$$

$$= \delta_j^{L} \times \frac{\partial}{\partial a_k^{L-1}}\left(\sum_k w_{jk}^{L} a_k^{L-1} + b_j^{L}\right) \cdot \frac{\partial a_k^{L-1}}{\partial w_{jk}^{L-1}}$$

$$= \delta_j^{L} \times w_{jk}^{L} \times \frac{\partial}{\partial w_{jk}^{L-1}}\,\sigma\left(\sum_k w_{jk}^{L-1} a_k^{L-2} + b_j^{L-1}\right)$$

$$\frac{\partial C}{\partial w_{jk}^{\ell-1}} = \delta_j^{L} \times w_{jk}^{L} \times \sigma'(z_j^{\ell-1}) \times a_k^{L-2}$$

$$\frac{\partial C}{\partial w^{L}_{jk}} = \delta^{L}_{j} \times a^{L-1}_{k}$$

$$\frac{\partial C}{\partial w^{L-1}_{jk}} = \delta^{L}_{j} \times w^{L}_{jk} \times \sigma'(z^{L-1}_{j}) \times a^{L-2}_{k}$$

$$\frac{\partial C}{\partial b^{L}_{j}} = \delta^{L}_{j}$$

$$\delta^{l}_{j} = \delta^{l+1}_{j} \cdot w^{l+1}_{jk}$$

$$(l \neq L)$$

$$\frac{\partial C}{\partial w_{jk}^{L-1}} = \delta_j^L \times w_{jk}^L \times \sigma'(z_j^{L-1}) \times a_k^{L-2}$$

$$\frac{\partial C}{\partial w_{jk}^{l}} = \boxed{\delta_j^{l+1} \times w_{jk}^{l+1}} \times \sigma'(z_j^{l}) \times a_k^{l-1}$$

$(l \neq L)$

$$= D_j^l \times \sigma'(z_j^l) \times a_k^{l-1}$$

where $D_j^l = \delta_j^{l+1} \times w_j^{l+1}$

$$\left.\begin{aligned}\frac{\partial C}{\partial w_{jk}^{l}} &= \delta_j^{l} \times a_k^{l-1} \\ \text{where } \delta_j^{l} &= \delta_j^{l+1} \times w_{jk}^{l+1}\end{aligned}\right\} \text{non final layer}$$

$$\frac{\partial C}{\partial w_{jk}^{L}} = (a_j^{L} - y_j)\, \sigma'(z_j^{L}) \;\Big\}\; \text{final layer}$$

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

$$= \sigma\left(w^l a^{l-1} + b^l\right)$$

$$\left[\quad\right]^T_{m\times n}\left[\ \right]_{n\times 1} + \left[\quad\right]_{n\times 1}$$

$l-1$ layer has m neuron

l layer has n neuron

Forward propagation

a_1^3

a_2^3

getting output

Backword pass

$\frac{\partial C}{\partial w}$ $\frac{\partial C}{\partial b}$ Computes gradient

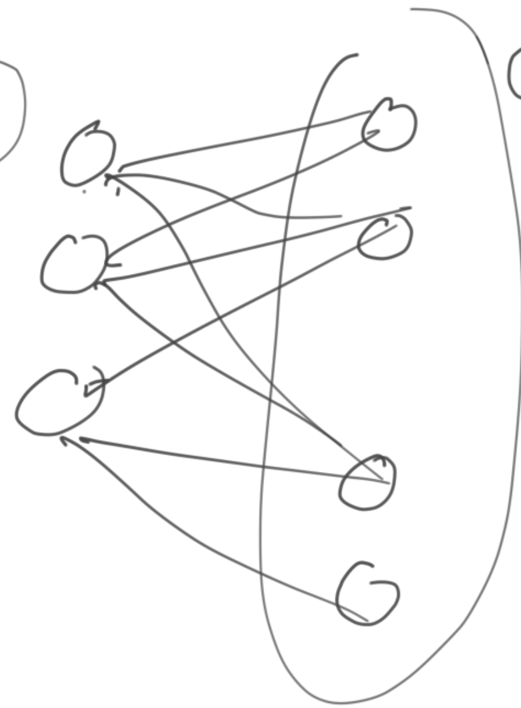
y_1

y_2

ground truth

1. Forward pass — output
2. Backward pass — gradient
3. gradient descent — Updates the weight

$$\frac{\partial(\alpha C_1 + \beta C_2)}{\partial W}$$



C_1 Task 1

C_2 Task 2

$\alpha C_1 + \beta C_2$ = Multitask learning