Neural Networks - II

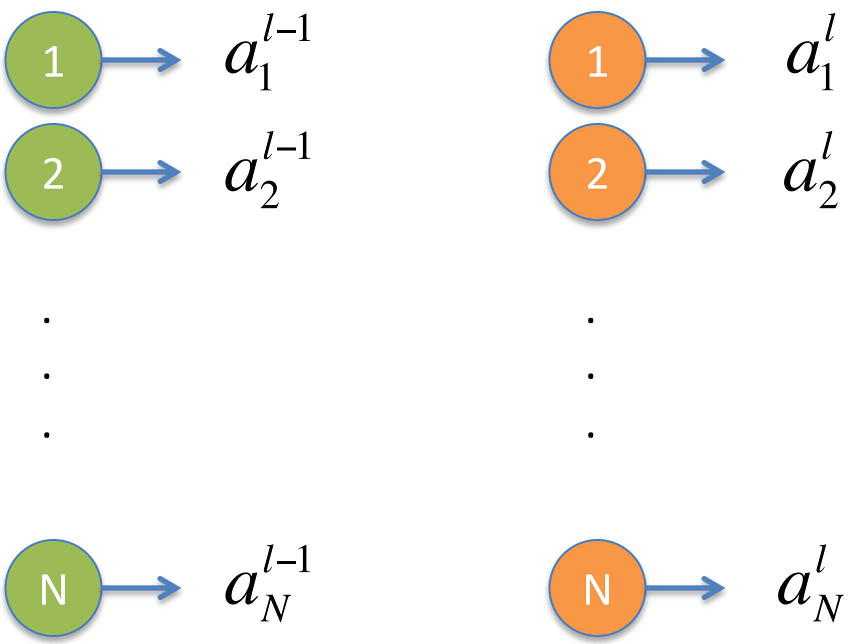
Thang Vu

21.11.2025

Outline

- Neuron
- Neural network
- Computing the output
- Training the network
- Backpropagation

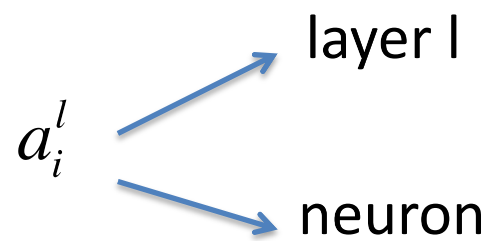
Notation



Layer l-1
N_{l-1} nodes

Layer l
N_l nodes

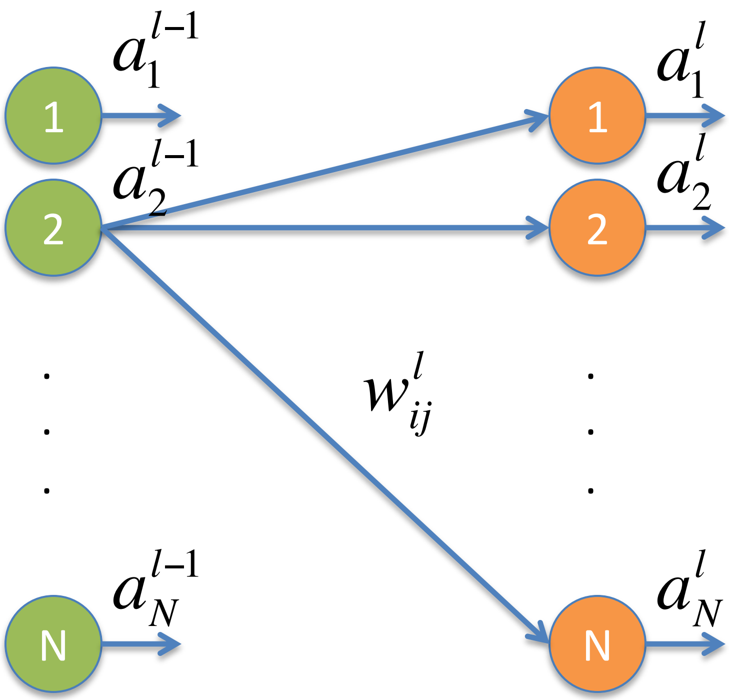
- Output of a neuron:



- Output of one layer:

a^l is a vector

Notation



Layer l-1

N_{l-1} nodes

Layer l

N_l nodes

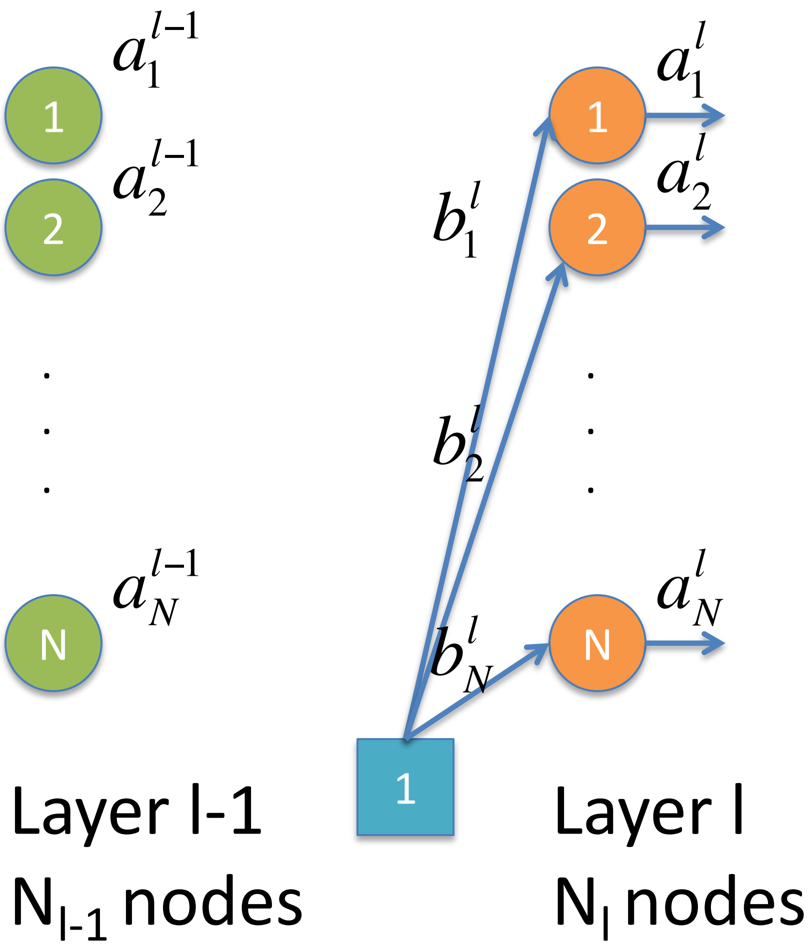
- **Weights:**

w_{ij}^l → layer *l – 1* to layer *l*

→ from neuron *j* (layer *l -1*) to neuron *i* (layer *l*)

$$W^l = \underbrace{\begin{bmatrix} w_{11}^l & w_{12}^l & \cdots \\ w_{21}^l & w_{22}^l & \cdots \\ \vdots & & \end{bmatrix}}_{N_{l-1} \text{ columns}} \quad N_l \text{ rows}$$

Notation



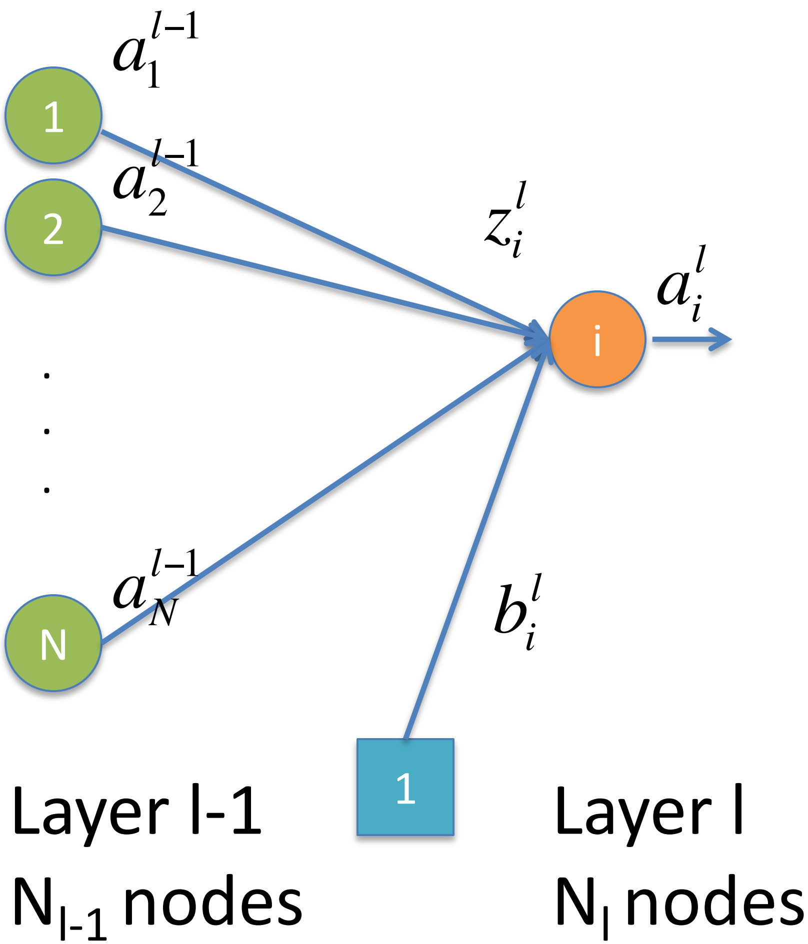
- Biases:

b_i^l → layer l

b_i^l → neuron i

$$b^l = \begin{bmatrix} b_1^l \\ b_2^l \\ \vdots \end{bmatrix}$$ Bias for all the neurons in layer l

Notation



z_i^l : input of the activation function for neuron i at layer l

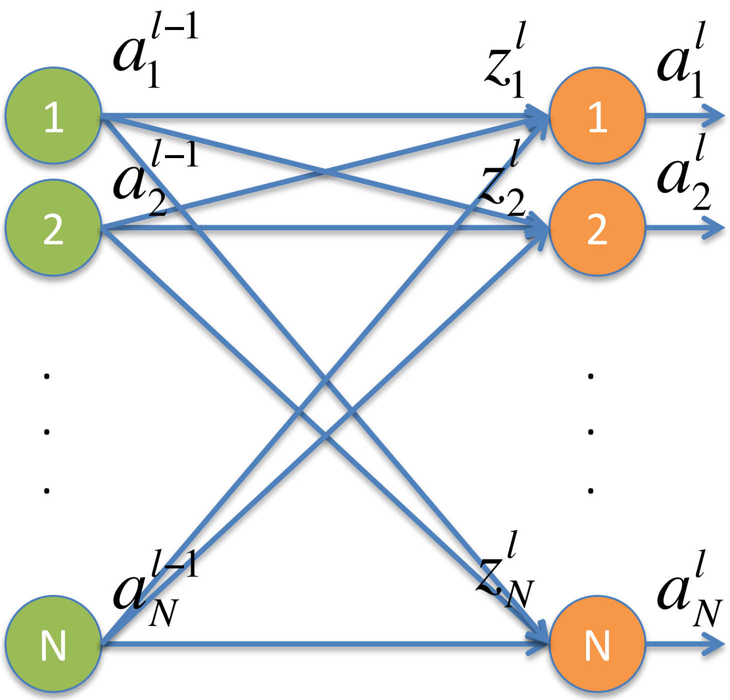
z^l : input of the activation function of all the neurons in layer l

$$z_i^l = w_{i1}^l a_1^{l-1} + w_{i2}^l a_2^{l-1} + \ldots + b_i^l$$

or in another form

$$z_i^l = \sum_{j=1}^{N_{l-1}} w_{ij}^l a_j^{l-1} + b_i^l$$

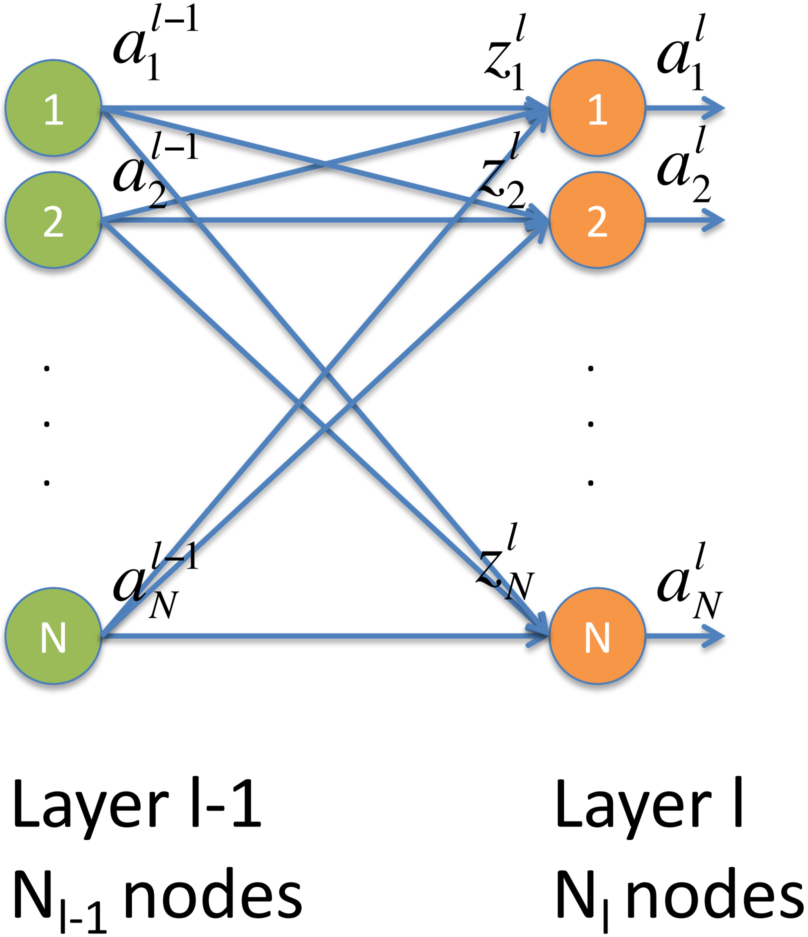
Relations between layer outputs



Layer l-1
N_{l-1} nodes

Layer l
N_l nodes

Relations between layer outputs



$$z_1^l = w_{11}^l a_1^{l-1} + w_{12}^l a_2^{l-1} + \ldots + b_1^l$$

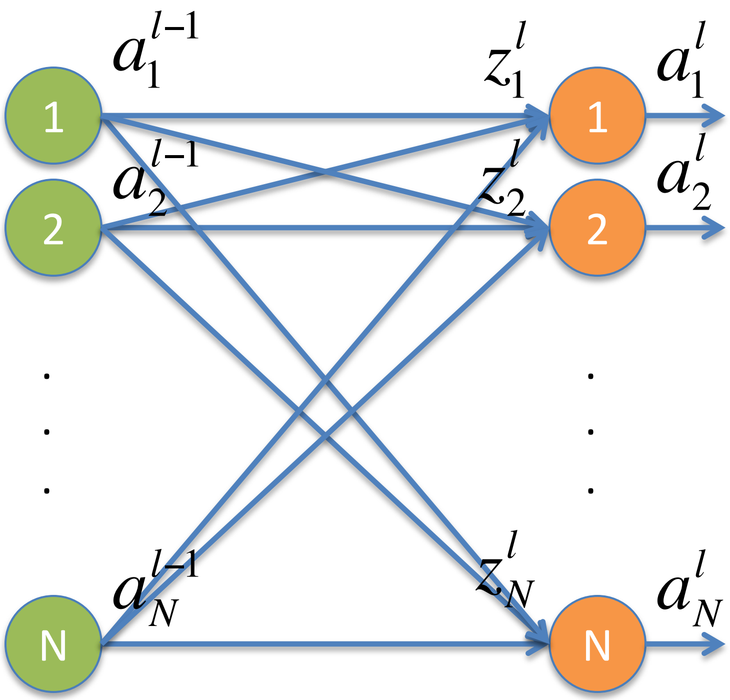
$$z_2^l = w_{21}^l a_1^{l-1} + w_{22}^l a_2^{l-1} + \ldots + b_2^l$$

.....

$$z_N^l = w_{N1}^l a_1^{l-1} + w_{N2}^l a_2^{l-1} + \ldots + b_N^l$$

$$z^l = W^l a^{l-1} + b^l$$

Relations between layer outputs



$$a_1^l = \sigma(z_1^l)$$

$$a_2^l = \sigma(z_2^l)$$

....

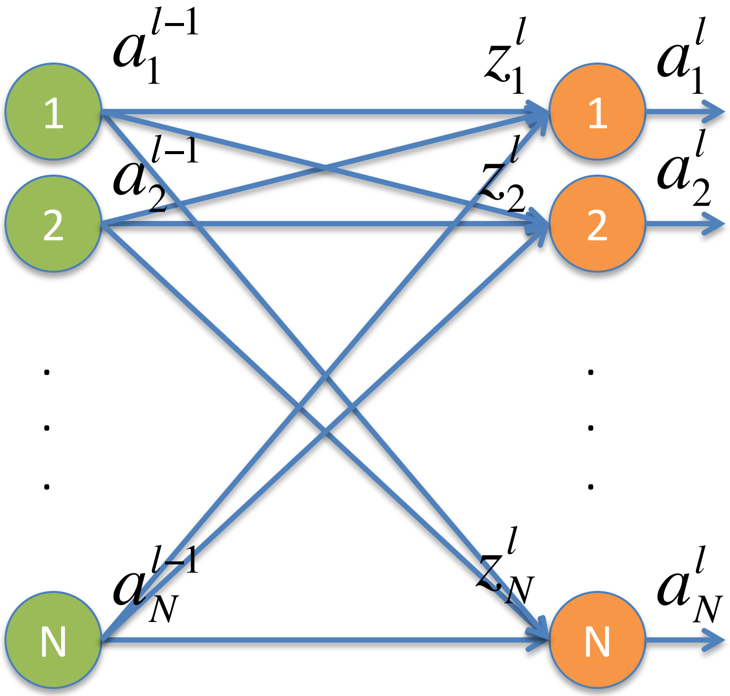
$$a_N^l = \sigma(z_N^l)$$

$$a^l = \sigma(z^l)$$

Layer l-1
N_{l-1} nodes

Layer l
N_l nodes

Relations between layer outputs



Layer l-1

N_{l-1} nodes

Layer l

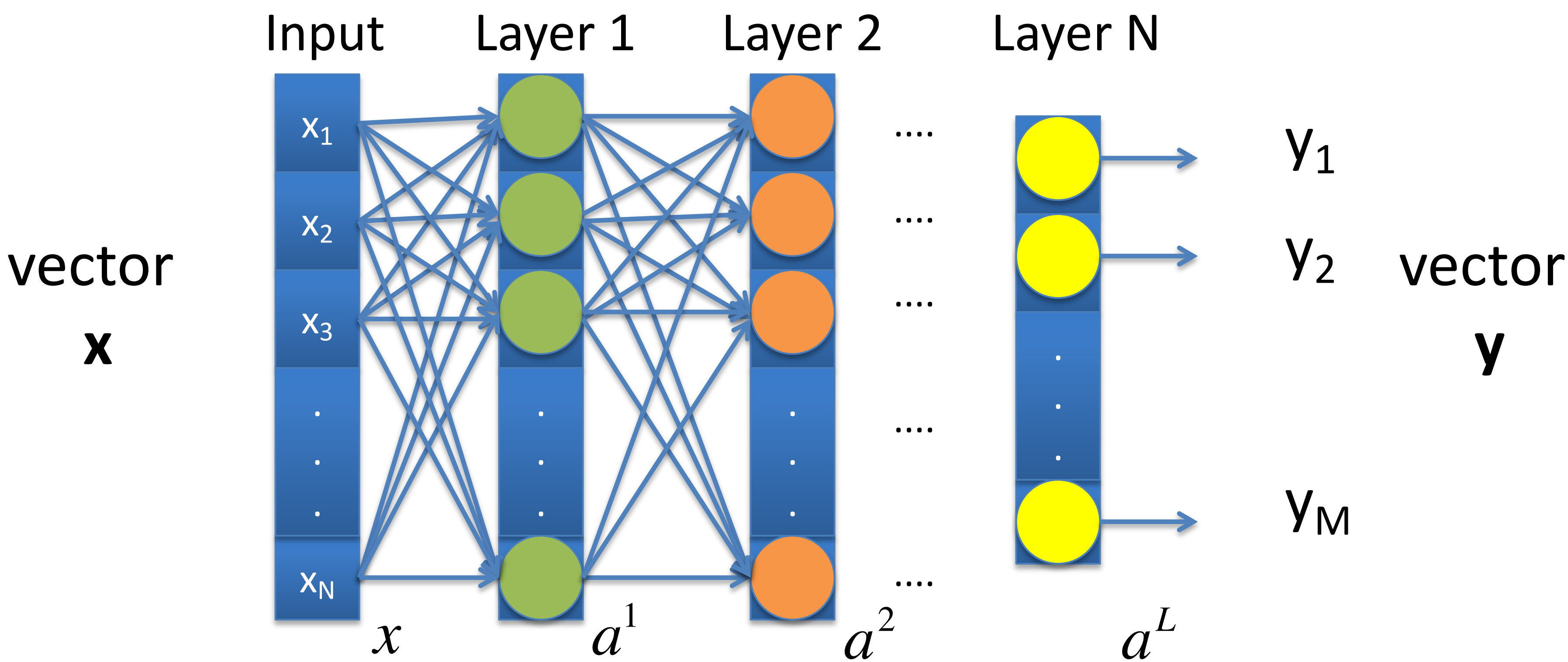
N_l nodes

$$z^l = W^l a^{l-1} + b^l$$

$$a^l = \sigma(z^l)$$

$$a^l = \sigma(W^l a^{l-1} + b^l)$$

Computation of the final output

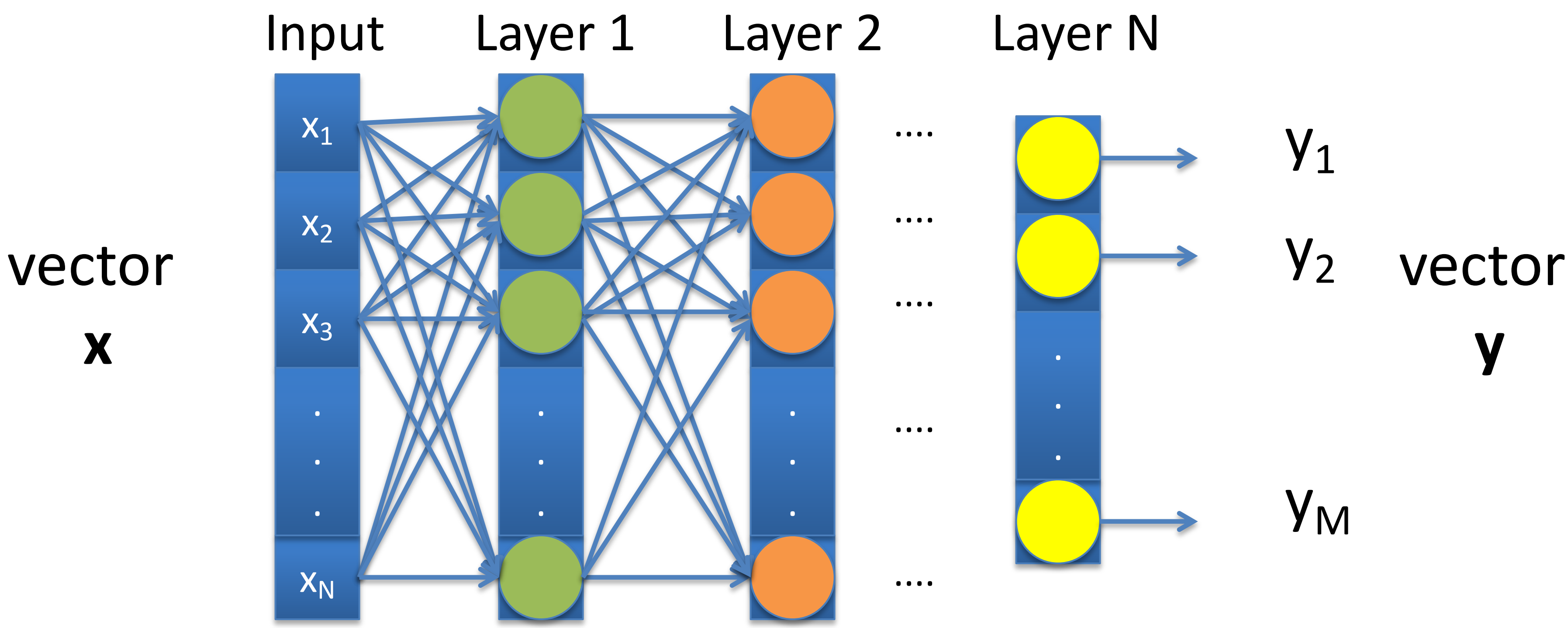


$$a^1 = \sigma(W^1 x + b^1)$$

$$a^L = \sigma(W^L a^{L-1} + b^L)$$

$$a^2 = \sigma(W^2 a^1 + b^2)$$

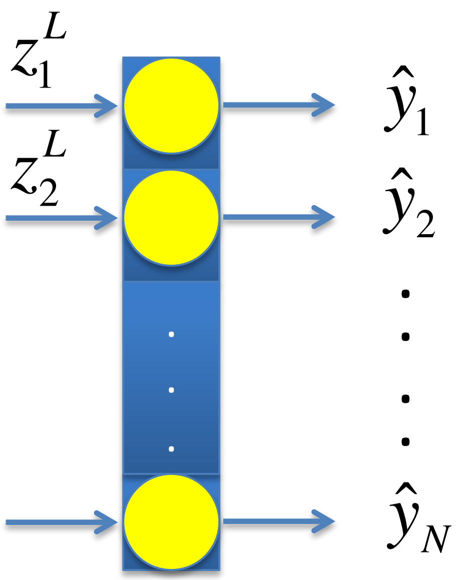
Computation of the final output



$$y = f(x) = \sigma(W^L \ldots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \ldots + b^L)$$

Softmax function

Ouput layer L



- Outputs are probabilities

$$0 < y_i < 1$$

$$\sum_i y_i = 1$$

- Sofmax function:

$$y_i = \frac{e^{z_i^L}}{\sum_j e^{z_j^L}}$$

$$\frac{\delta y_i}{\delta z_j} = y_j(1 - y_j) \qquad i = j$$

$$= -y_i y_j \qquad i \neq j$$

Train a Neural Network

Searching for the best function

- Best function = best parameters

$$y = f(x) = \sigma(W^L \ldots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \ldots + b^L)$$

- The function itself depends on the parameter set

$$f(x) = f(x, \theta)$$

$$\theta = \{W^1, b^1, W^2, b^2, \ldots, W^L, b^L\}$$

- Seach the best function f*

→ Search the best parameter set θ*

Empirical risk minimization

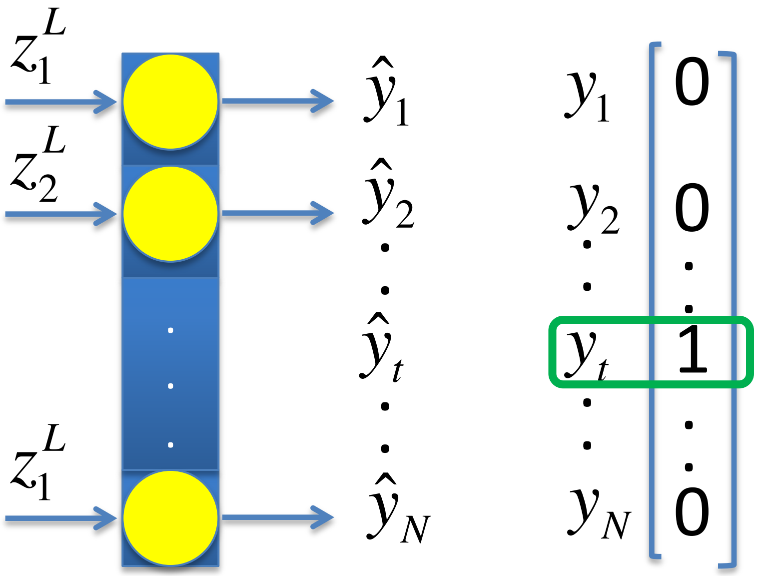
- Given a finite set of training data
- Empirical risk = average loss on this training data

$$C(\theta) = \frac{1}{|D|} \sum_{(x,y)} c(f(x), y)$$

$$= \frac{1}{|D|} \sum_{(x,y)} c(\theta)$$

Loss function

Ouput layer L



- Sofmax function:

$$y_i = \frac{e^{z_i^L}}{\sum_j e^{z_j^L}}$$

- Mean Squared Error (MSE)

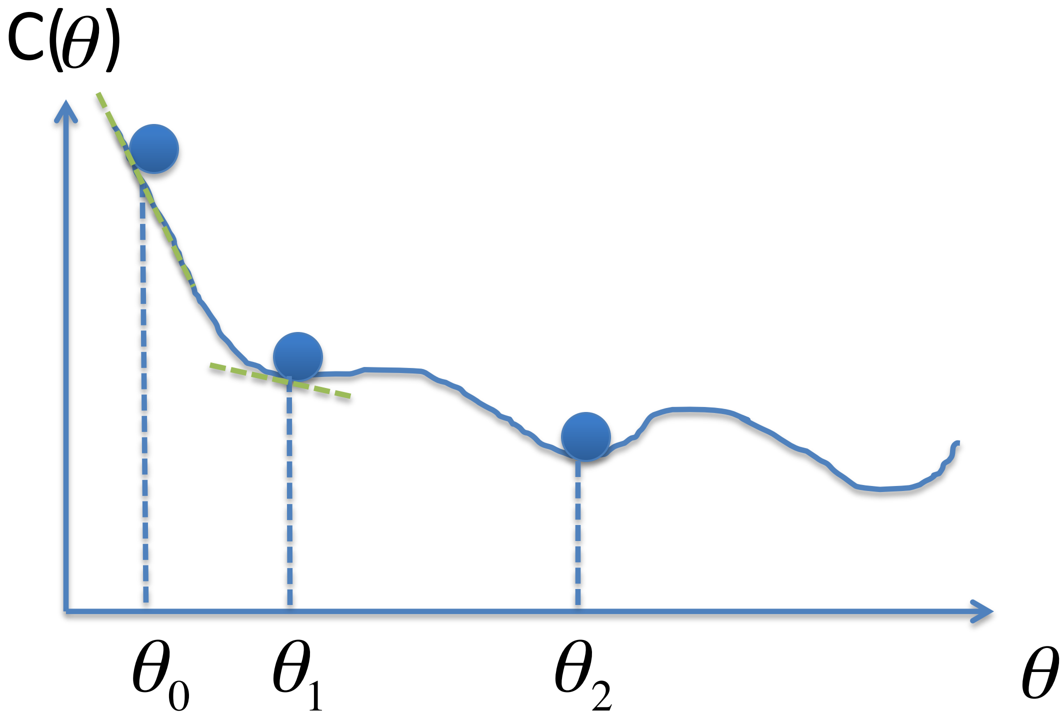
$$c(\theta) = C_x(\theta) = \sum_i (\hat{y}_i - y_i)^2$$

- Cross Entropy (CE):

$$c(\theta) = C_x(\theta) = -\log \hat{y}_t$$

Gradient descent

- If θ has only one variable



- Randomly start at θ_0
- Compute $dC(\theta_0)/d\theta$

$$\theta_1 \leftarrow \theta_0 - \eta dC(\theta_0)/d\theta$$

- Compute $dC(\theta_1)/d\theta$

$$\theta_2 \leftarrow \theta_1 - \eta dC(\theta_1)/d\theta$$

- ……

Gradient descent for Neural Network

- We will do the same thing as presented before
- Starting parameters θ_0

$$\longrightarrow \theta_1 \longrightarrow \theta_2 \longrightarrow \theta_3 \;\ldots\ldots\ldots\ldots$$

- However,

$$\theta = \{W^1, b^1, W^2, b^2, \ldots, W^L, b^L\}$$

- i.e. millions of parameters ☹

Backpropagation

Based on the chain rules:

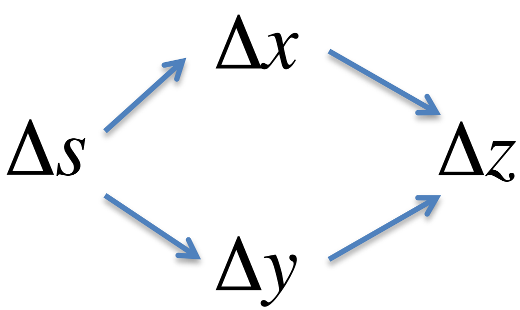
- Case 1: $y = g(x)$

  $z = h(y)$

  $\Delta x \rightarrow \Delta y \rightarrow \Delta z$

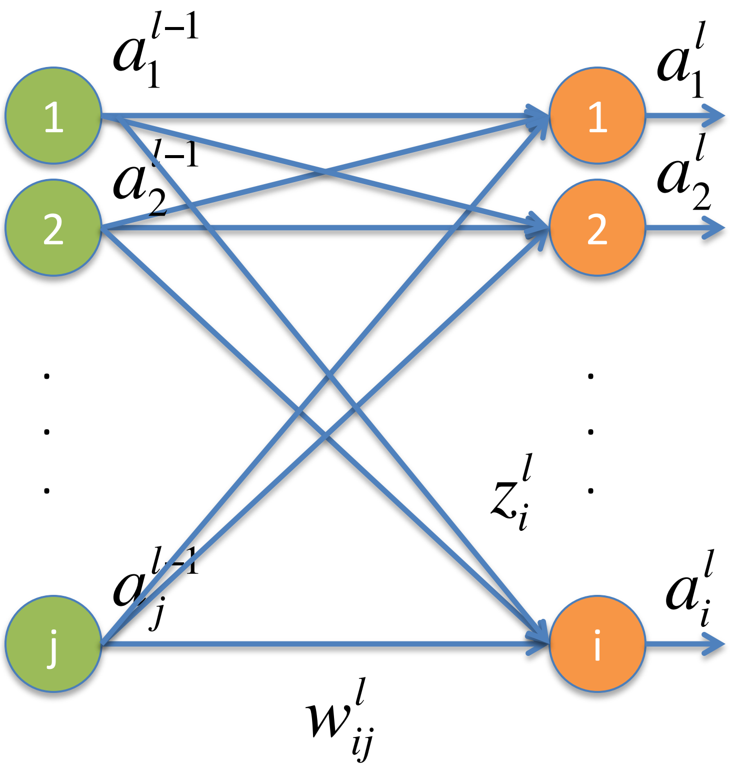
  $$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

- Case 2: $x = g(s) \quad y = h(s) \quad z = k(x, y)$



$$\frac{\partial z}{\partial s} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial s}$$

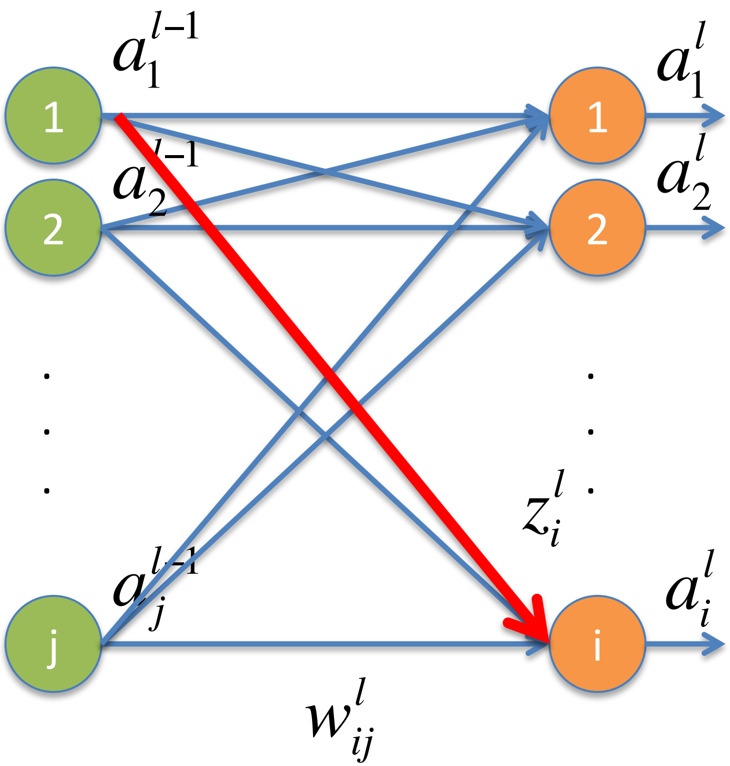
Compute $\frac{\partial C}{\partial w_{ij}^l}$



$$\Delta w_{ij}^l \rightarrow \Delta z_i^l \ldots \rightarrow \Delta C$$

$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C}{\partial z_i^l}$$

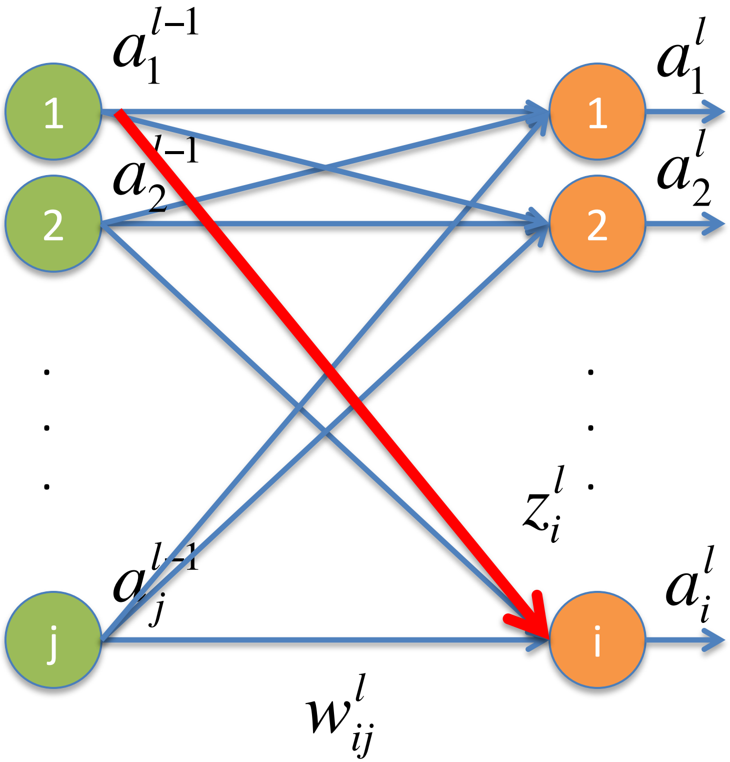
Compute $\frac{\partial C}{\partial w_{ij}^l}$



$$\Delta w_{ij}^l \rightarrow \Delta z_i^l \ldots \rightarrow \Delta C$$

$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C}{\partial z_i^l}$$

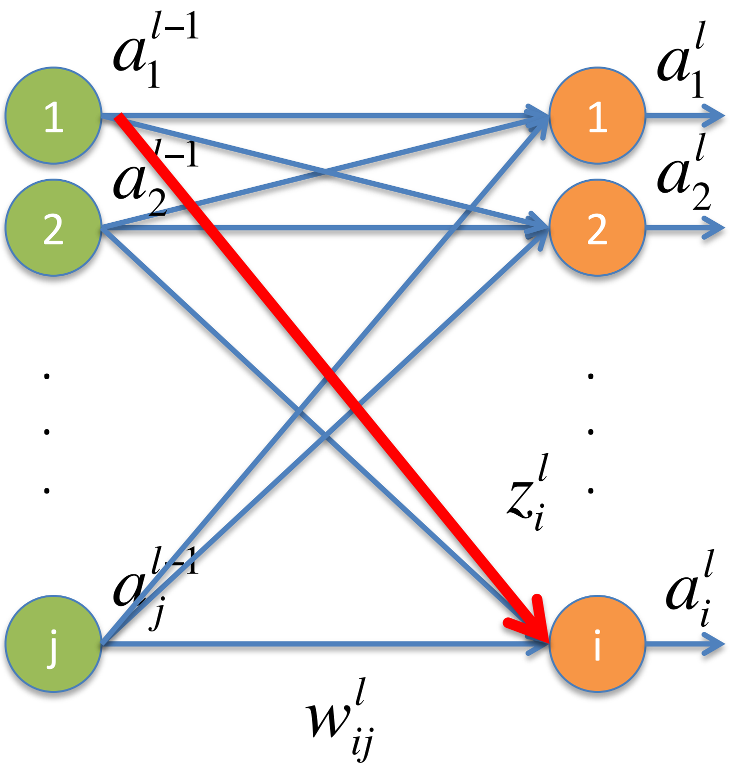
Compute - $\frac{\partial C}{\partial w_{ij}^l}$ First term



$$\Delta w_{ij}^l \rightarrow \Delta z_i^l \ldots \rightarrow \Delta C$$

$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C}{\partial z_i^l}$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$ - First term



- If l > 1:

$$z_i^l = \sum_j w_{ij}^l a_j^{l-1} + b_i^l$$

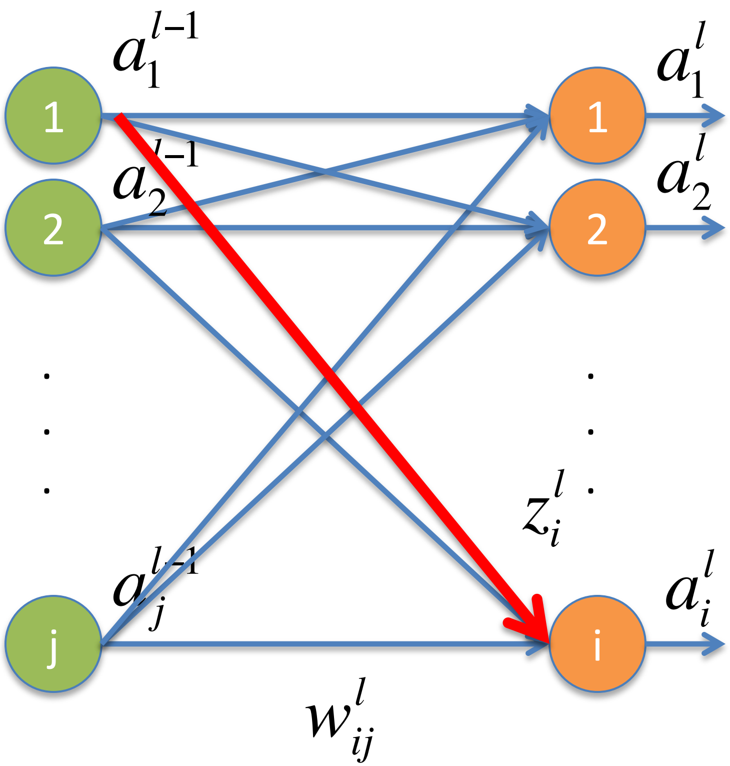
$$\frac{\partial z_i^l}{\partial w_{ij}^l} = a_j^{l-1}$$

- If l = 1:

$$z_i^l = \sum_j w_{ij}^1 x_j + b_i^1$$

$$\frac{\partial z_i^1}{\partial w_{ij}^1} = x_j$$

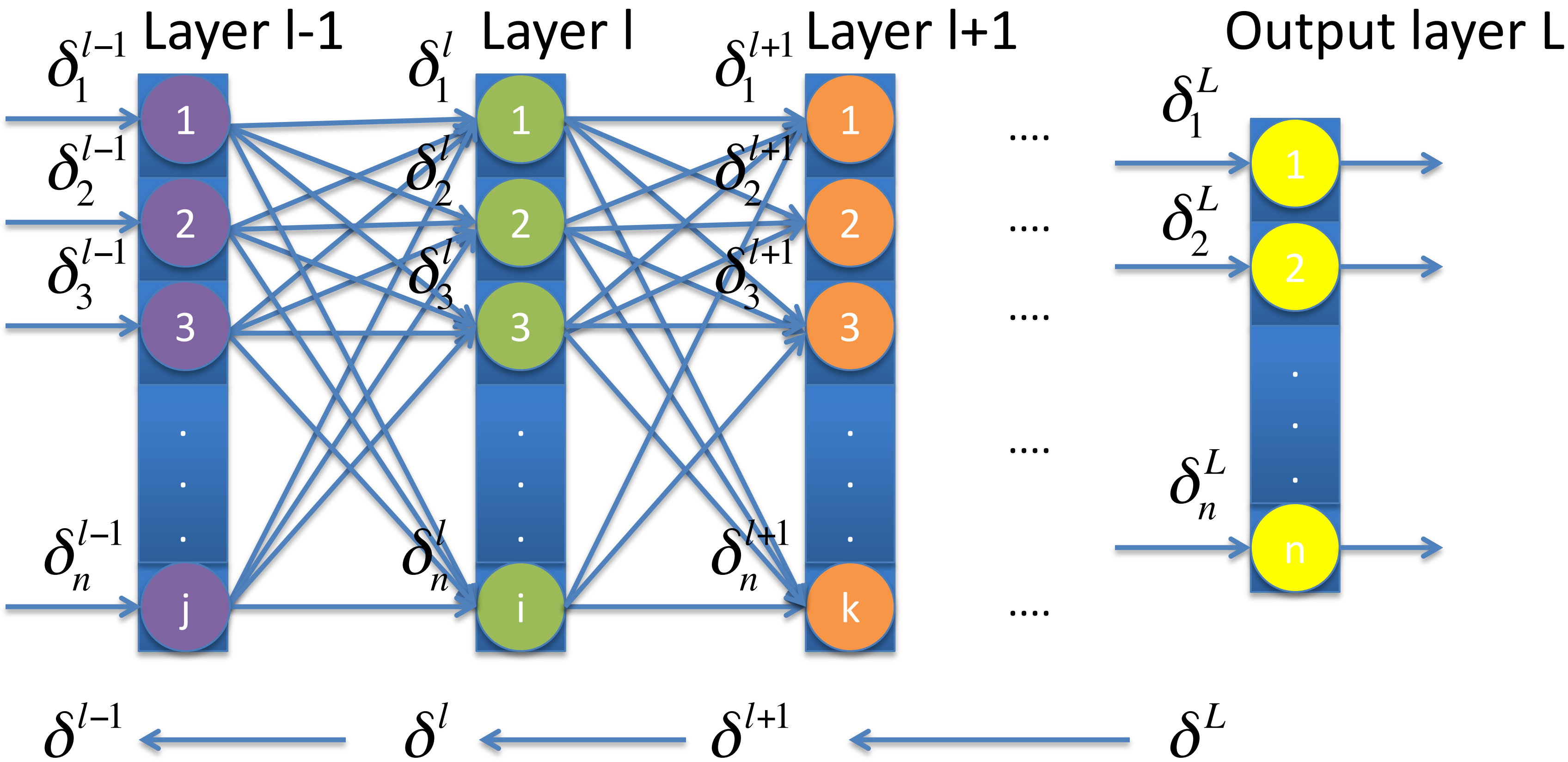
Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



$$\Delta w_{ij}^l \rightarrow \Delta z_i^l \ldots \rightarrow \Delta C$$

$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C}{\partial z_i^l}$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



Compute $\frac{\partial C}{\partial w_{ij}^{l}}$ - Second term

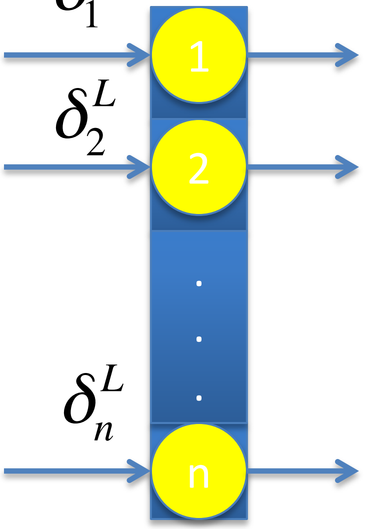
$$\Delta z_n^L \rightarrow \Delta a_n^L = \Delta y_n \rightarrow \Delta C$$

$$\delta_n^L = \frac{\partial C}{\partial z_n^L} = \frac{\partial a_n^L}{\partial z_n^L} \frac{\partial C}{\partial a_n^L}$$

$$\sigma'(z_n^L)$$

Depending on
The loss function

Output layer L



δ^L

$$\delta^L = \sigma'(z^L) \nabla C(y)$$ Elementwise multiplication

Second term – Last layer

- If the last layer uses softmax and cross-entropy loss is used

$$c(\theta) = C_x(\theta) = -\log \hat{y}_t$$

$$\delta_i^L = \frac{\partial a_i^L}{\partial z_i^L} \frac{\partial C}{\partial a_i^L}$$

$$\frac{\partial C}{\partial a_i^L} = -\frac{1}{\hat{y}_t}$$

Ground truth

$$\begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_t \\ \vdots \\ y_N \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Second term – Last layer

- If the last layer uses softmax and cross-entropy loss is used

$$\delta_i^L = \frac{\partial a_i^L}{\partial z_i^L} \frac{\partial C}{\partial a_i^L}$$

$$y_i = \frac{e^{z_i^L}}{\sum_j e^{z_j^L}}$$

$$\frac{\delta y_i}{\delta z_j} = y_j(1 - y_j) \quad i = j$$
$$= -y_i y_j \quad i \neq j$$

Ground truth

$$\begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_t \\ \vdots \\ y_N \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$\frac{\partial y_t^L}{\partial z_i^L} = y_t(1 - y_t) \text{ if i = t}$$
$$= -y_t y_i \text{ if i} \neq \text{t}$$

Second term – Last layer

- If the last layer uses softmax and cross-entropy loss is used

$$\delta_i^L = \frac{\partial a_i^L}{\partial z_i^L}\frac{\partial C}{\partial a_i^L}$$

$$\frac{\partial y_t^L}{\partial z_i^L} = y_t(1 - y_t) \text{ if i = t}$$
$$= -y_t y_i \text{ if i} \neq \text{t}$$

$$\frac{\partial C}{\partial a_i^L} = -\frac{1}{\hat{y}_t}$$

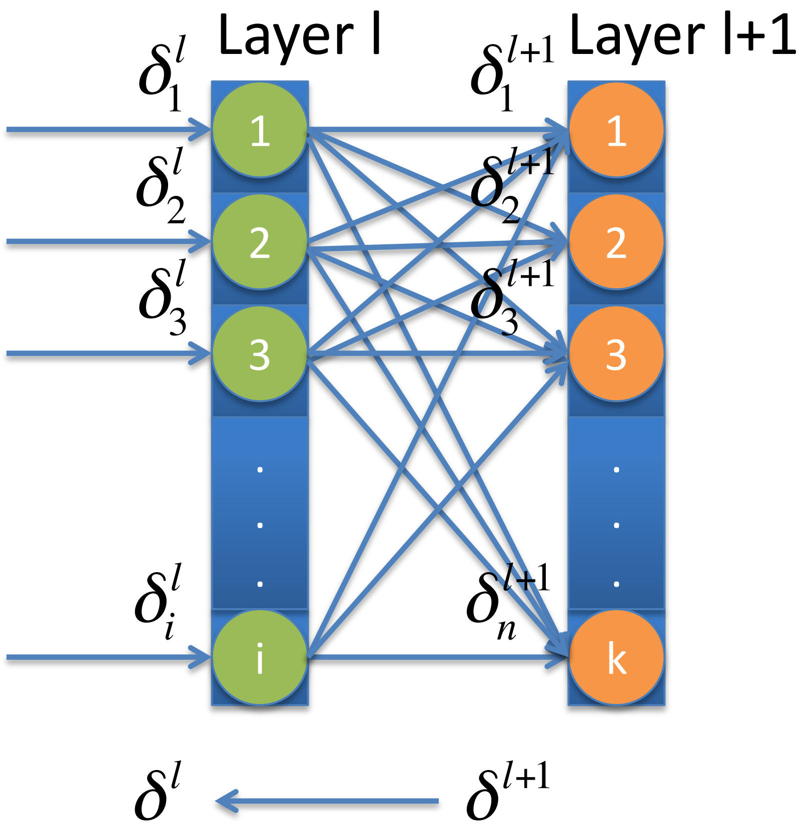
Ground truth

$$\begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_t \\ \vdots \\ y_N \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$\delta_i^L = \hat{y}_t(1 - \hat{y}_t)\left(-\frac{1}{\hat{y}_t}\right) = \hat{y}_t - 1 \quad \text{if i = t}$$

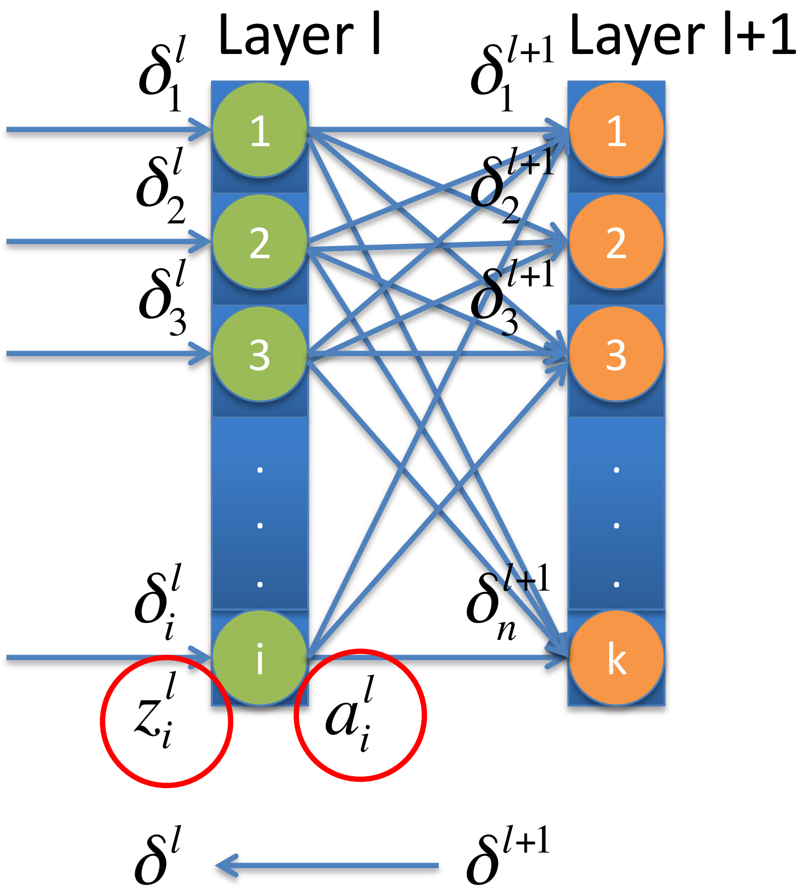
$$= -\hat{y}_t\hat{y}_i\left(-\frac{1}{\hat{y}_t}\right) = \hat{y}_i \qquad \text{if i} \neq \text{t}$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



$$\delta_i^l = \frac{\partial C}{\partial z_i^l}$$

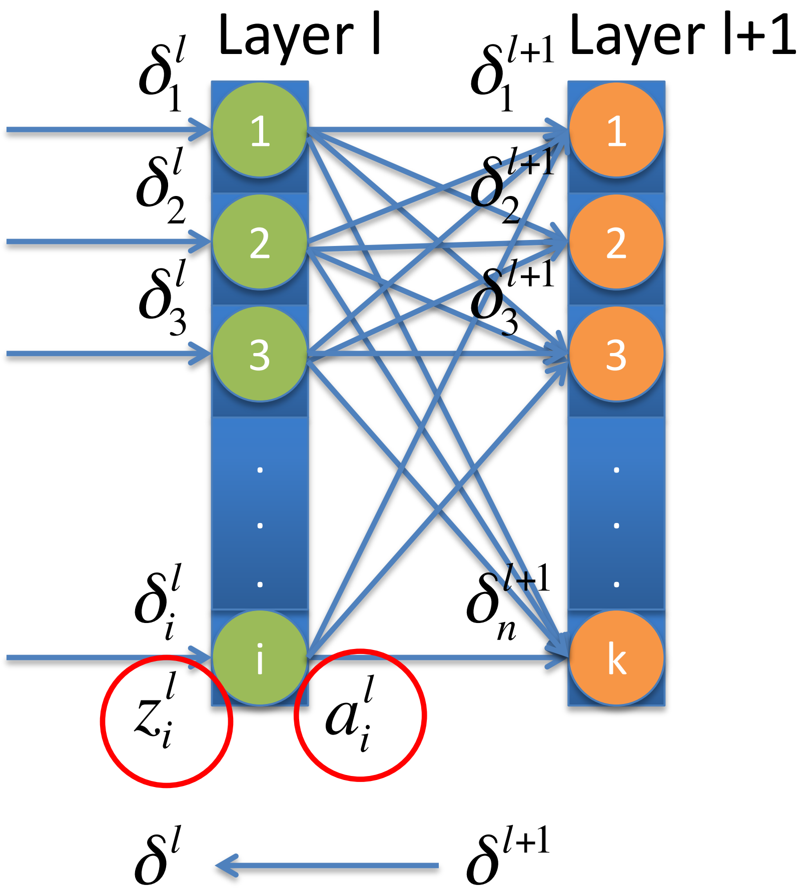
Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



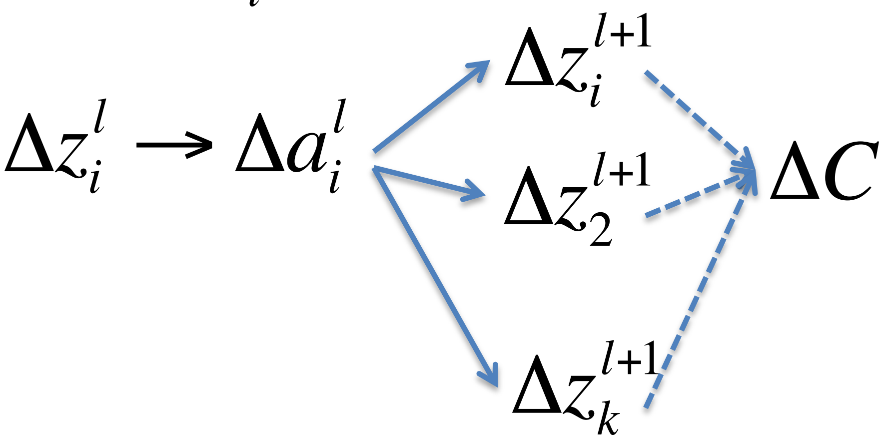
$$\delta_i^l = \frac{\partial C}{\partial z_i^l}$$

$$\Delta z_i^l \rightarrow \Delta a_i^l$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



$$\delta_i^l = \frac{\partial C}{\partial z_i^l}$$



Chain Rules

- Case 1:

$$y = g(x)$$

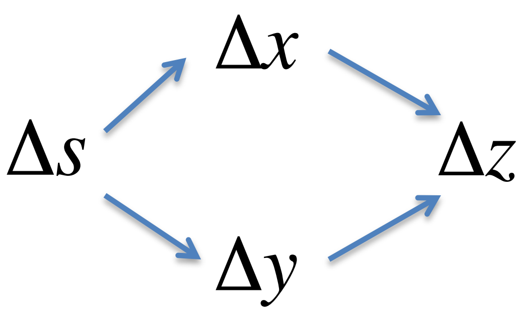
$$z = h(y)$$

$$\Delta x \rightarrow \Delta y \rightarrow \Delta z$$

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

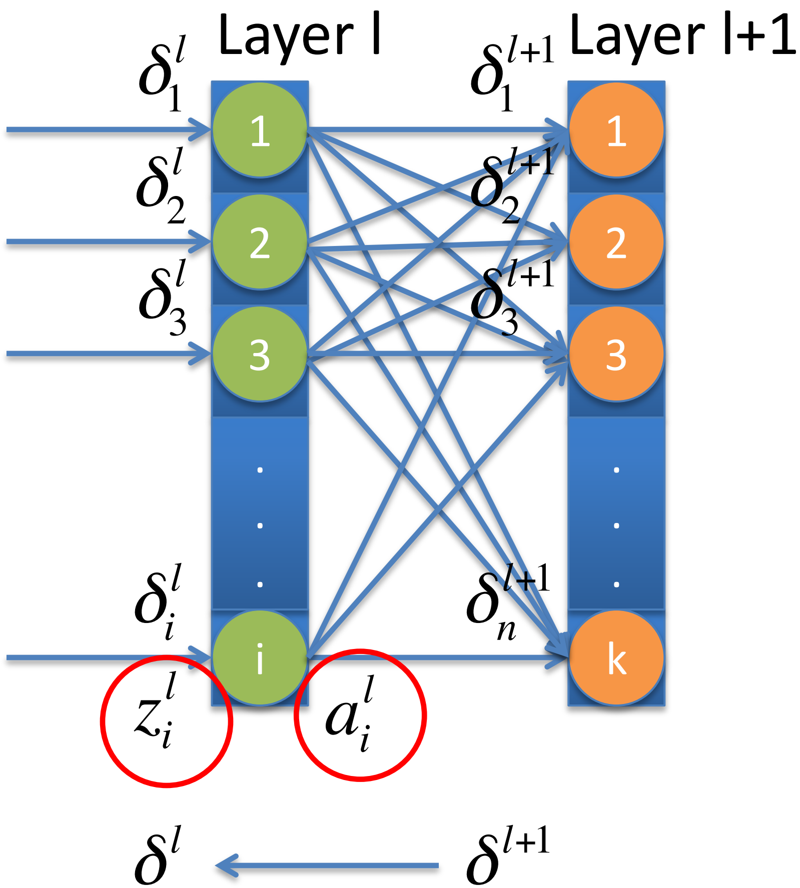
- Case 2:

$$x = g(s) \qquad y = h(s) \qquad z = k(x, y)$$

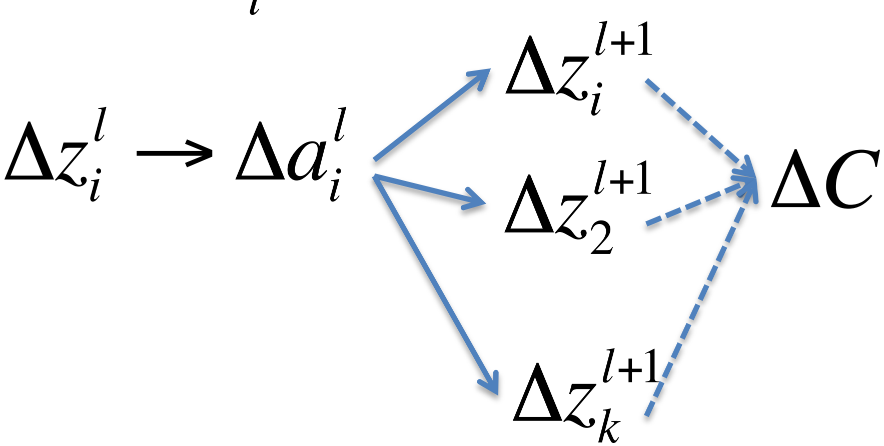


$$\frac{\partial z}{\partial s} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial s}$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term

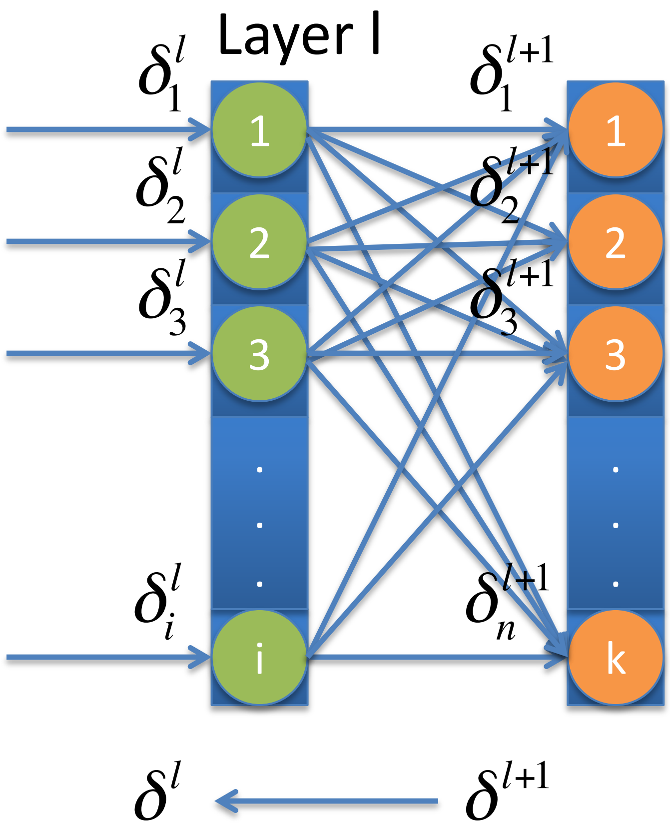


$$\delta_i^l = \frac{\partial C}{\partial z_i^l}$$



$$\delta_i^l = \frac{\partial C}{\partial z_i^l} = \frac{\partial a_i^l}{\partial z_i^l} \sum_k \frac{\partial z_k^{l+1}}{\partial a_i^l} \frac{\partial C}{\partial z_k^{l+1}}$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



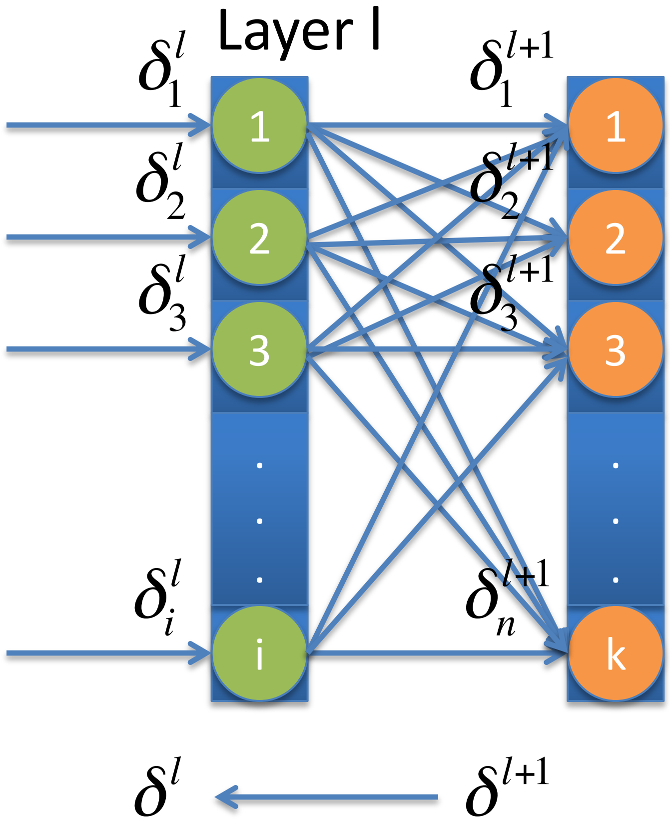
$$\delta_i^l = \frac{\partial C}{\partial z_i^l} = \frac{\partial a_i^l}{\partial z_i^l} \sum_k \frac{\partial z_k^{l+1}}{\partial a_i^l} \frac{\partial C}{\partial z_k^{l+1}}$$

$$\frac{\partial a_i^l}{\partial z_i^l} \rightarrow \sigma'(z_i^l) \qquad \frac{\partial C}{\partial z_k^{l+1}} \rightarrow \delta_k^{l+1}$$

$$z_k^{l+1} = \sum_i w_{ki}^{l+1} a_i^l + b_k^{l+1}$$

$$\delta_i^l = \sigma'(z_i^l) \sum_k w_{ki}^{l+1} \delta_k^{l+1}$$

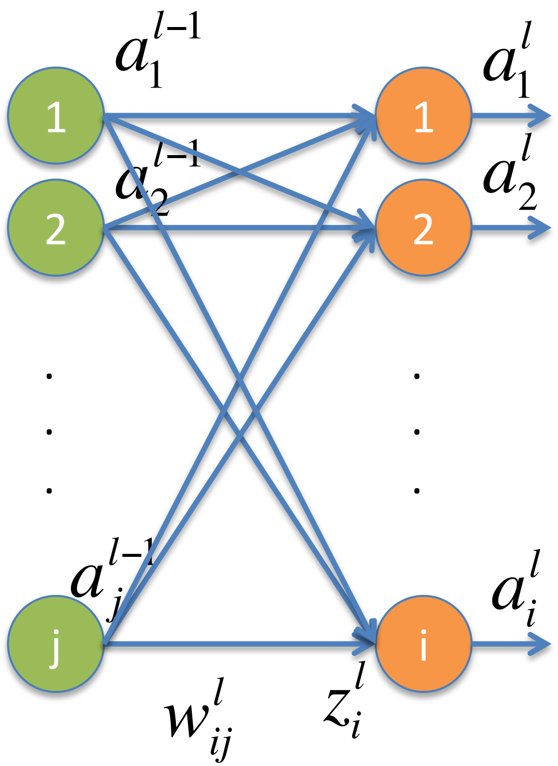
Compute $\frac{\partial C}{\partial w_{ij}^l}$ - Second term



$$\delta_i^l = \sigma'(z_i^l) \sum_k w_{ki}^{l+1} \delta_k^{l+1}$$

$$\delta^l = \sigma'(z^l).(W^{l+1})^T \delta^{l+1}$$

Compute $\frac{\partial C}{\partial w_{ij}^l}$



$$\frac{\partial C}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C}{\partial z_i^l}$$

$$\frac{\partial z_i^l}{\partial w_{ij}^l} = \begin{cases} a_j^{l-1} & l > 1 \\ x_j & l = 1 \end{cases}$$

$$\frac{\partial C}{\partial z_i^l} = \delta_i^l$$

Forward pass

$$z^l = W^l a^{l-1} + b^l$$

$$a^l = \sigma(z^l)$$

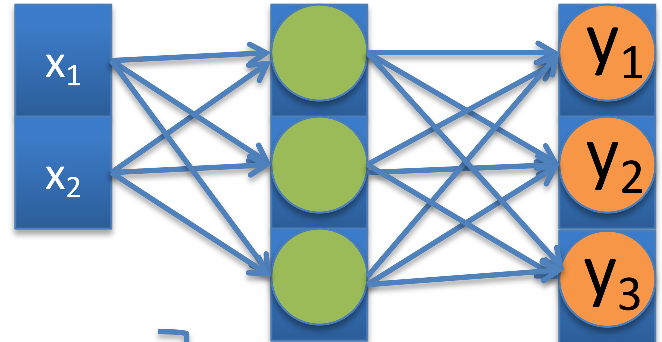
Backward pass

$$\delta^L = \sigma'(z^L)\nabla C(y)$$

$$\delta^l = \sigma'(z^l)(W^{l+1})^T \delta^{l+1}$$

Exercise

- Given the following network
- Layer 1 uses ReLU
- Layer 2 uses Softmax



$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad W^1 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \quad W^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ground truth:

$$y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$b^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad b^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Exercise

- Compute the cross entropy loss
- Compute $\frac{\delta C}{\delta w_{21}^{1}}$

Live Voting

Thanks for listening!