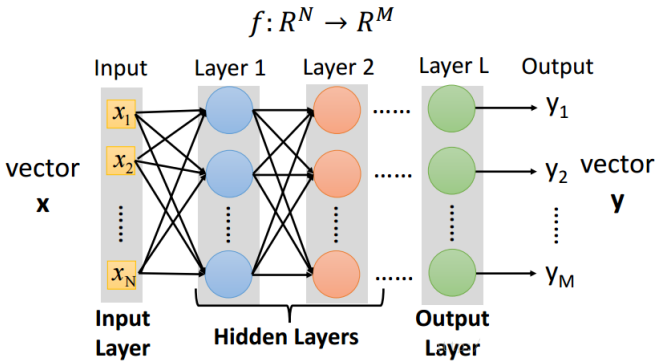
Backpropagation for FC Neural Network

Hao Huang

Friday 9th February, 2018

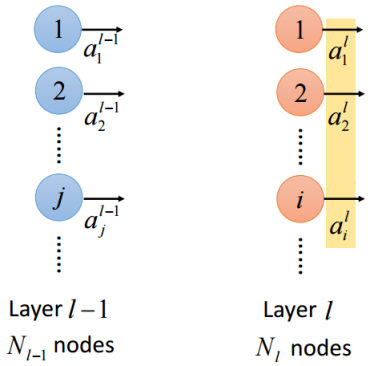
Contents

1 Network Structure



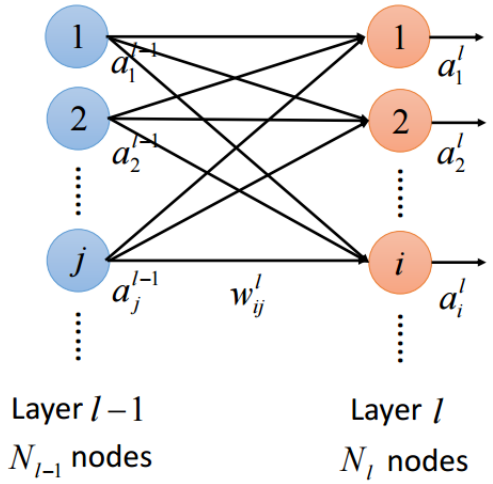
2 Notations

2.1 Output of activation



Output of a neuron: a_i^l (Superscript l is the layer, and subscript is the index in layer)
Output of a layer: a^l (a vector)

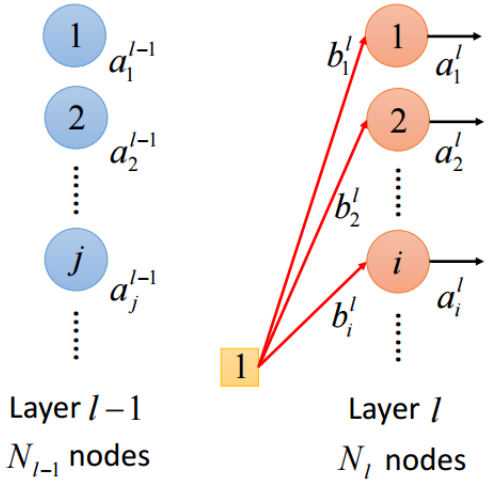
2.2 Weights



w_{ij}^l: from neuron j (layer $l-1$) to neuron i (layer l)

$$W^l = \begin{bmatrix} w_{11}^l & w_{12}^l & \dots & w_{1j}^l & \dots & w_{1N_{l-1}}^l \\ w_{21}^l & w_{22}^l & \dots & w_{2j}^l & \dots & w_{2N_{l-1}}^l \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{N_l1}^l & w_{N_l2}^l & \dots & w_{N_lj}^l & \dots & w_{N_lN_{l-1}}^l \end{bmatrix}_{N_l \times N_{l-1}}$$

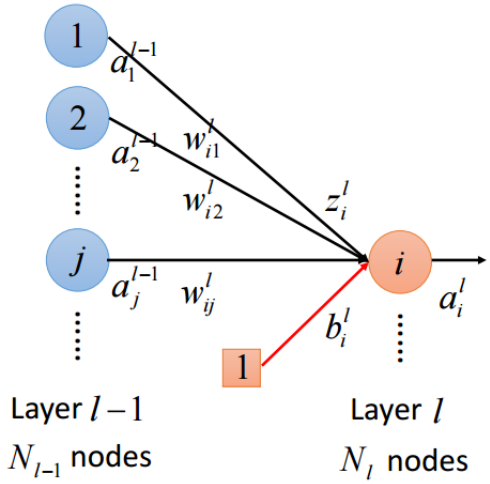
2.3 Biases



b_i^l: bias for neuron i at layer l

$$b^l = \begin{bmatrix} b_1^l & b_2^l & \dots & b_i^l & \dots & b_{N_l}^l \end{bmatrix}^T$$

2.4 Input of activation

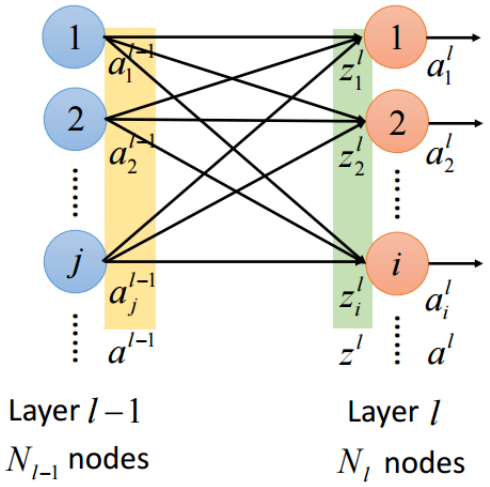


z_i^l: input of the activation function of neuron i at layer l
z^l: input of the activation function of all neurons at layer l

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

$$z^l = \begin{bmatrix} z_1^l & z_2^l & \dots & z_i^l & \dots & z_{N_l}^l \end{bmatrix}^T$$

2.5 Relations between layers



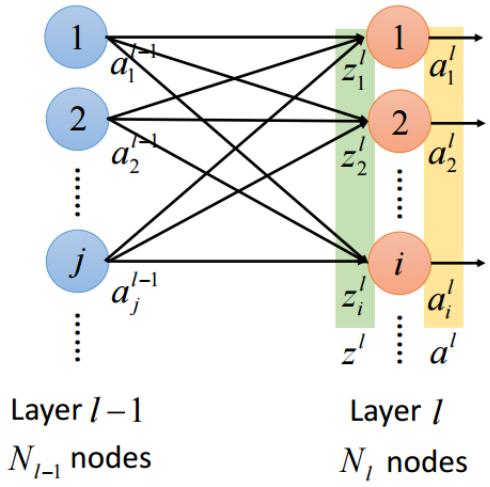
$$z_1^l = w_{11}^l a_1^{l-1} + w_{12}^l a_2^{l-1} + \cdots + w_{1j}^l a_j^{l-1} + \cdots + w_{1N_{l-1}}^l a_{N_{l-1}}^{l-1} + b_1^l$$
$$z_2^l = w_{21}^l a_1^{l-1} + w_{22}^l a_2^{l-1} + \cdots + w_{2j}^l a_j^{l-1} + \cdots + w_{2N_{l-1}}^l a_{N_{l-1}}^{l-1} + b_2^l$$
$$\vdots$$
$$z_i^l = w_{i1}^l a_1^{l-1} + w_{i2}^l a_2^{l-1} + \cdots + w_{ij}^l a_j^{l-1} + \cdots + w_{iN_{l-1}}^l a_{N_{l-1}}^{l-1} + b_i^l$$
$$\vdots$$
$$z_{N_l}^l = w_{N_l1}^l a_1^{l-1} + w_{N_l2}^l a_2^{l-1} + \cdots + w_{N_lj}^l a_j^{l-1} + \cdots + w_{N_lN_{l-1}}^l a_{N_{l-1}}^{l-1} + b_{N_l}^l$$

Rewrite them as matrix multiplication form:

$$\begin{bmatrix} z_1^l \\ z_2^l \\ \vdots \\ z_i^l \\ \vdots \\ z_{N_l}^l \end{bmatrix} = \begin{bmatrix} w_{11}^l & w_{12}^l & \cdots & w_{1j}^l & \cdots & w_{1N_{l-1}}^l \\ w_{21}^l & w_{22}^l & \cdots & w_{2j}^l & \cdots & w_{2N_{l-1}}^l \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{i1}^l & w_{i2}^l & \cdots & w_{ij}^l & \cdots & w_{iN_{l-1}}^l \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{N_l1}^l & w_{N_l2}^l & \cdots & w_{N_lj}^l & \cdots & w_{N_lN_{l-1}}^l \end{bmatrix} \begin{bmatrix} a_1^{l-1} \\ a_2^{l-1} \\ \vdots \\ a_j^{l-1} \\ \vdots \\ a_{N_{l-1}}^{l-1} \end{bmatrix} + \begin{bmatrix} b_1^l \\ b_2^l \\ \vdots \\ b_i^l \\ \vdots \\ b_{N_l}^l \end{bmatrix}$$

That is:

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



For neuron i in layer l:

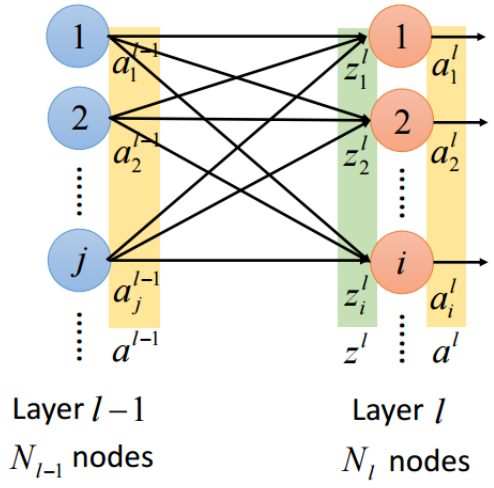
$$a_i^l = \sigma(z_i^l)$$

For the layer l:

$$\begin{bmatrix} a_1^l \\ a_2^l \\ \vdots \\ a_i^l \\ \vdots \\ a_{N_l}^l \end{bmatrix} = \begin{bmatrix} \sigma(z_1^l) \\ \sigma(z_2^l) \\ \vdots \\ \sigma(z_i^l) \\ \vdots \\ \sigma(z_{N_l}^l) \end{bmatrix}$$

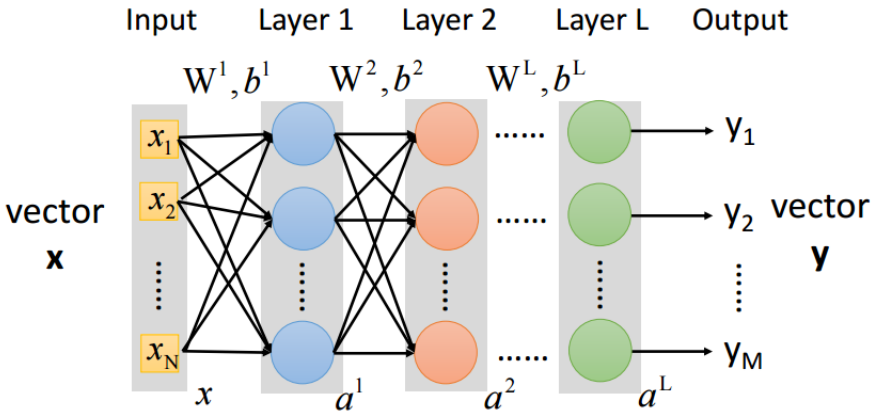
Rewrite it as:

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



$$\left.\begin{aligned} a^l &= \sigma(z^l) \\ z^l &= W^l a^{l-1} + b^l \end{aligned}\right\} \Rightarrow a^l = \sigma(W^l a^{l-1} + b^l)$$

2.6 Summary of notation



$$a^1 = \sigma(W^1 x + b^1)$$
$$a^2 = \sigma(W^2 a^1 + b^2)$$
$$\vdots$$
$$y = a^L = \sigma(W^L a^{L-1} + b^L)$$

Consider a neural network as a function:

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

3 Three gradient descent approaches

θ is the parameter set of a neural network (e.g. $\{W^1, b^1, ..., W^L, b^L\}$), and C is the cost function. The superscript i indicates the i-th update. N is the number of training samples.

3.1 Gradient descent

$$\theta^i = \theta^{i-1} - \eta \nabla C(\theta^{i-1})$$
$$\nabla C(\theta^{i-1}) = \frac{1}{N} \sum_{r=1}^{N} \nabla C_r(\theta^{i-1})$$

3.2 Stochastic gradient descent

Randomly pick a training sample x_r:

$$\theta^i = \theta^{i-1} - \eta \nabla C_r(\theta^{i-1})$$

3.3 Mini-batch gradient descent

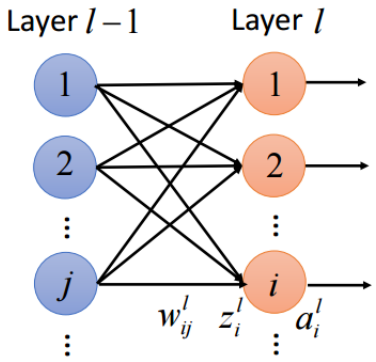
Shuffle the training data and then pick M samples as a batch:

$$\theta^i = \theta^{i-1} - \eta \frac{1}{M} \sum_{x_r \in batch} \nabla C_r(\theta^{i-1})$$

4 Backpropagation

The change in w_{ij}^l leads to the change in C_r, so we need to compute ∇C_r with respect to w_{ij}^l.

4.1 Partial derivative of C_r w.r.t w_{ij}^l



$$\Delta w_{ij}^l \to \Delta z_i^l \to \Delta a_i^l \to \cdots \to \Delta C_r$$

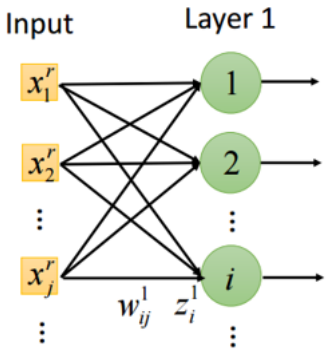
$$\frac{\partial C_r}{\partial w_{ij}^l} = \frac{\partial C_r}{\partial z_i^l} \frac{\partial z_i^l}{\partial w_{ij}^l}$$

4.2 Partial derivative of z_i^l w.r.t w_{ij}^l

We divide this partial derivative into two parts: input layer and hidden layer(s).

4.2.1 Input layer

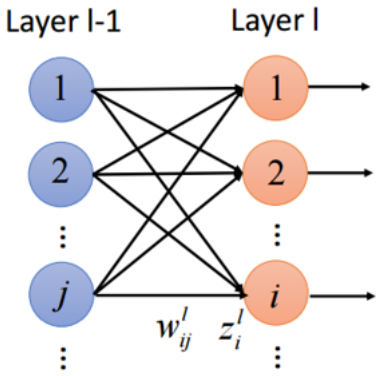
Suppose the input layer has N_0 input nodes. The superscript r of x means the r-th sample.



$$z_i^1 = \sum_j^{N_0} w_{ij}^1 x_j^r + b_i^1$$

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

4.2.2 Hidden layer



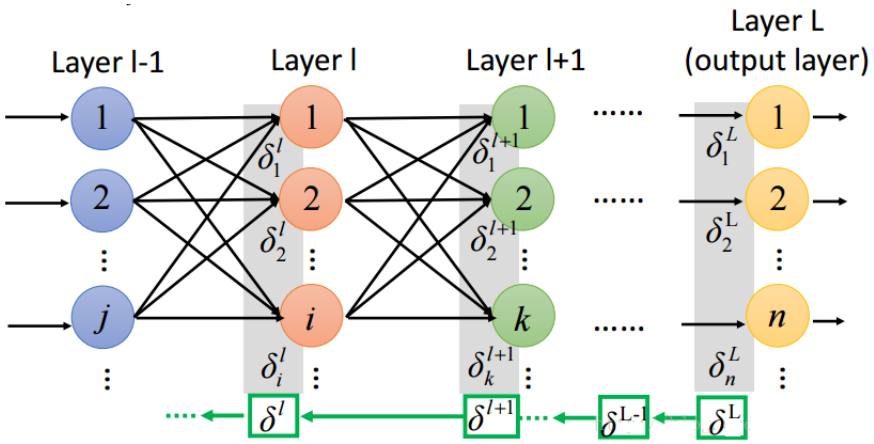
$$z_i^l = \sum_j^{N_{l-1}} w_{ij}^l a_j^{l-1} + b_i^l$$

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

4.3 Partial derivative of C_r w.r.t z_i^l

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

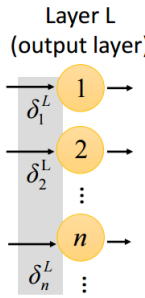
We denote $\frac{\partial C_r}{\partial z_i^l}$ as δ_i^l and define $\delta^l = [\delta_1^l \; \delta_2^l \; \dots \delta_i^l \; \dots \; \delta_{N_l}^l]^T$.



We need to compute δ^l. However, computing δ^l can be divided into two steps:

1. computing δ^L (output layer)
2. figuring out the relationship between δ^{l+1} and δ^l

4.3.1 Computing δ^L (output layer)



$$\Delta z_n^L \to \Delta a_n^L = \Delta y_n^r \to \Delta C_r$$

$$\delta_n^L = \frac{\partial C_r}{\partial z_n^L} = \frac{\partial C_r}{\partial y_n^r}\frac{\partial y_n^r}{\partial z_n^L} = \frac{\partial C_r}{\partial y_n^r}\sigma'(z_n^L)$$

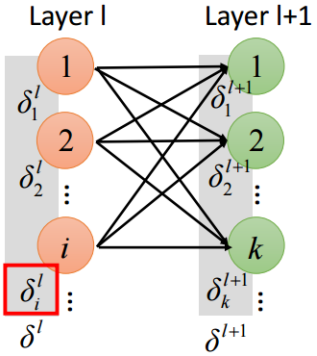
In addition, $\frac{\partial C_r}{\partial y_n^r}$ depends on the definition of the cost function.

$$\delta^L = \begin{bmatrix} \delta_1^L \\ \delta_2^L \\ \vdots \\ \delta_n^L \\ \vdots \\ \delta_{N_L}^L) \end{bmatrix} = \begin{bmatrix} (\partial C_r \,/\, \partial y_1^r) \times \sigma'(z_1^L) \\ (\partial C_r \,/\, \partial y_2^r) \times \sigma'(z_2^L) \\ \vdots \\ (\partial C_r \,/\, \partial y_n^r) \times \sigma'(z_n^L) \\ \vdots \\ (\partial C_r \,/\, \partial y_{N_L}^r) \times \sigma'(z_{N_L}^L) \end{bmatrix} = \begin{bmatrix} (\partial C_r \,/\, \partial y_1^r) \\ (\partial C_r \,/\, \partial y_2^r) \\ \vdots \\ (\partial C_r \,/\, \partial y_n^r) \\ \vdots \\ (\partial C_r \,/\, \partial y_{N_L}^r) \end{bmatrix} \odot \begin{bmatrix} \sigma'(z_1^L) \\ \sigma'(z_2^L) \\ \vdots \\ \sigma'(z_n^L) \\ \vdots \\ \sigma'(z_{N_L}^L) \end{bmatrix}$$

$\odot$ is element-wise multiplication. We rewrite this formula is a vector form:

$$\delta^L = \nabla C_r(y^r) \odot \sigma'(z^L)$$

4.3.2 Relationship between δ^{l+1} and δ^l



Because the output of i-th node in layer l (e.g., a_i^l) will be passed in to every node in layer $l+1$, so we have:

$$\Delta z_i^l \to \Delta a_i^l \to \left\{ \begin{array}{l} \Delta z_1^{l+1} \\ \Delta z_2^{l+1} \\ \vdots \\ \Delta z_k^{l+1} \\ \vdots \\ \Delta z_{N_{l+1}}^{l+1} \end{array} \right\} \dashrightarrow \Delta C_r$$

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

$$\because a_i^l = \sigma(z_i^l) \ \wedge \ z_k^{l+1} = \sum_i^{N_l} w_{ki}^{l+1} a_i^l + b_k^{l+1}$$

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

Let us rewrite it in vector (and matrix) form:

$$\sigma'(z^l) = \begin{bmatrix} \sigma'(z_1^l) \\ \sigma'(z_2^l) \\ \vdots \\ \sigma'(z_i^l) \\ \vdots \\ \sigma'(z_{N_l}^l) \end{bmatrix} \ \wedge \ \delta^{l+1} = \begin{bmatrix} \delta_1^{l+1} \\ \delta_2^{l+1} \\ \vdots \\ \delta_k^{l+1} \\ \vdots \\ \delta_{N_{l+1}}^{l+1} \end{bmatrix}$$

$$\delta^l = \begin{bmatrix} \delta_1^l \\ \delta_2^l \\ \vdots \\ \delta_i^l \\ \vdots \\ \delta_{N_l}^l \end{bmatrix} = \begin{bmatrix} \sigma'(z_1^l) \\ \sigma'(z_2^l) \\ \vdots \\ \sigma'(z_i^l) \\ \vdots \\ \sigma'(z_{N_l}^l) \end{bmatrix} \odot \left(\begin{bmatrix} w_{11}^{l+1} & w_{21}^{l+1} & \dots & w_{k1}^{l+1} & \dots & w_{N_{l+1}1}^{l+1} \\ w_{12}^{l+1} & w_{22}^{l+1} & \dots & w_{k2}^{l+1} & \dots & w_{N_{l+1}2}^{l+1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{1i}^{l+1} & w_{2i}^{l+1} & \dots & w_{ki}^{l+1} & \dots & w_{N_{l+1}i}^{l+1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{1N_l}^{l+1} & w_{2N_l}^{l+1} & \dots & w_{kN_l}^{l+1} & \dots & w_{N_{l+1}N_l}^{l+1} \end{bmatrix} \begin{bmatrix} \delta_1^{l+1} \\ \delta_2^{l+1} \\ \vdots \\ \delta_k^{l+1} \\ \vdots \\ \delta_{N_{l+1}}^{l+1} \end{bmatrix} \right)$$

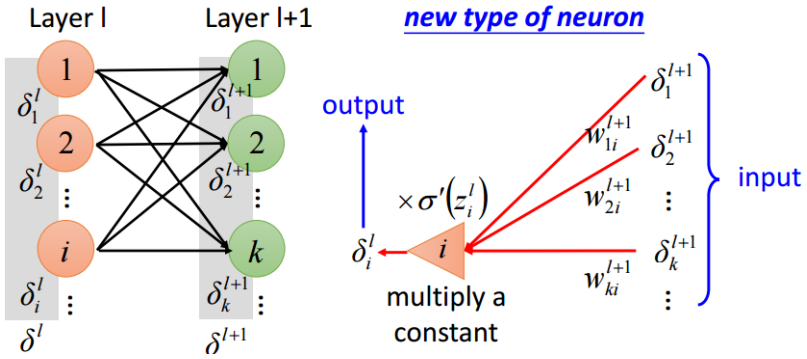
That is:

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

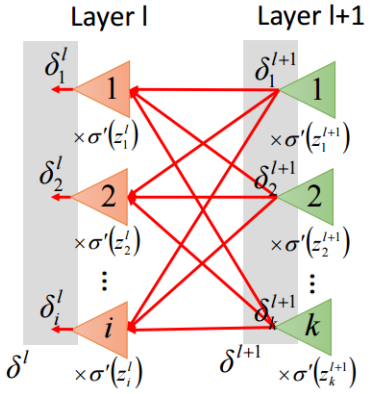
Thus we get the relationship between δ^{l+1} and δ^l.

4.3.3 Another viewpoint

From this formula: $\delta_i^l = \sigma'(z_i^l) \sum_k^{N_{l+1}} w_{ki}^{l+1} \delta_k^{l+1}$, we can view it as a new type of neuron, where $\sigma'(z_i^l)$ is an "amplifier".

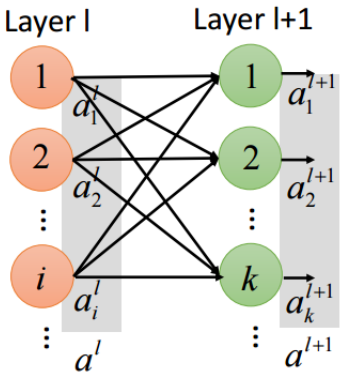


The relationship between the layer l and layer l+1 (e.g., $\delta^l = \sigma'(z^l) \odot [(W^{l+1})^T \delta^{l+1}]$) can be viewed as:



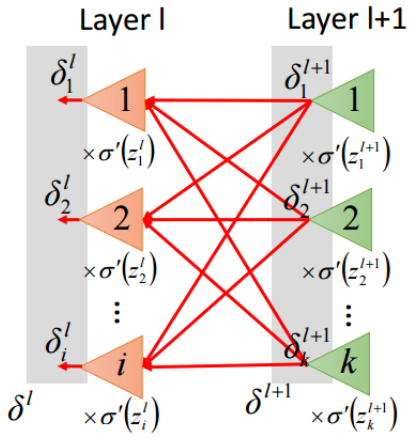
Now we compare the forward-pass network and the backward-pass network:

1. Forward-pass network



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

2. Backward-pass network



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

5 Summary

Our goal is:

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

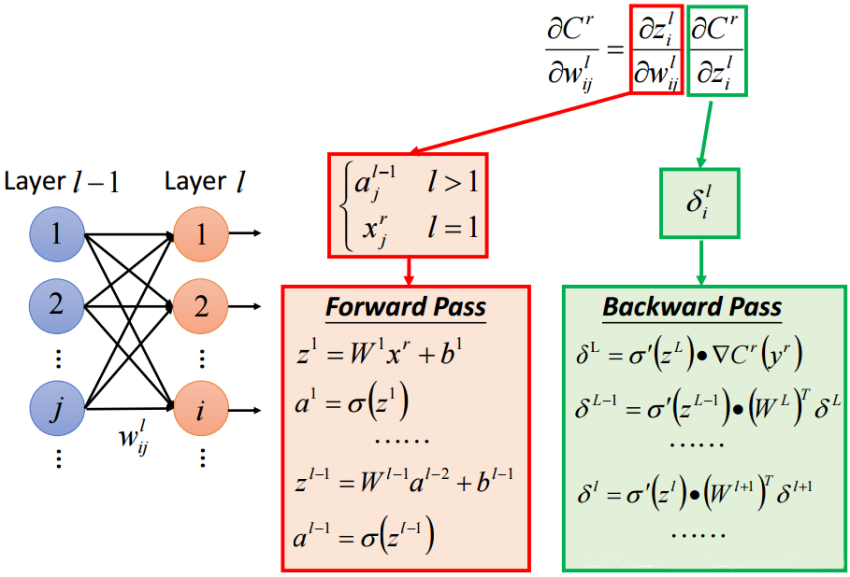
$$\frac{\partial z_i^l}{\partial w_{ij}^l} = \begin{cases} x_j^r & \text{input layer} \\ a_j^{l-1} & \text{hidden layer} \end{cases}$$

To compute $\delta_i^l = \frac{\partial C_r}{\partial z_i^l}$, we divide it into two step:

1. For output layer L:
$$\delta^L = \nabla C_r(y^r) \odot \sigma'(z^L)$$

2. For hidden layers δ^{l+1} and δ^l:
$$\delta^l = \sigma'(z^l) \odot [(W^{l+1})^T \delta^{l+1}]$$

To visualize forward-pass and backward-pass, we have:



The black dot in the figure is equivalent to $\odot$ and means element-wise multiplication.