Optmization Methods for Machine Learning
Gradient method for multilayer perceptron

Laura Palagi
http://www.dis.uniroma1.it/∼palagi

Dipartimento di Ingegneria informatica automatica e gestionale A. Ruberti
Sapienza Università di Roma

Via Ariosto 25

SAPIENZA
UNIVERSITÀ DI ROMA

Unconstrained problem

$$\min_{w,b} E(w,b)$$

- Existence of a global solution
- Optimality conditions (for a point to be a local solution)
- Definition of an iterative algorithm

$$\begin{pmatrix} w^{k+1} \\ b^{k+1} \end{pmatrix} = \begin{pmatrix} w^k \\ b^k \end{pmatrix} + \alpha^k d^k$$

- Convergence

BP Gradient method

$$\min_w E(w) = \sum_{p=1}^{P} E_p(w) = \sum_{p=1}^{P} \frac{1}{2} \|e^p(w)\|^2$$

where $e^p(w) = y(w; x^p) - y^p \in \mathbb{R}^K$ being K the dimension of the output y

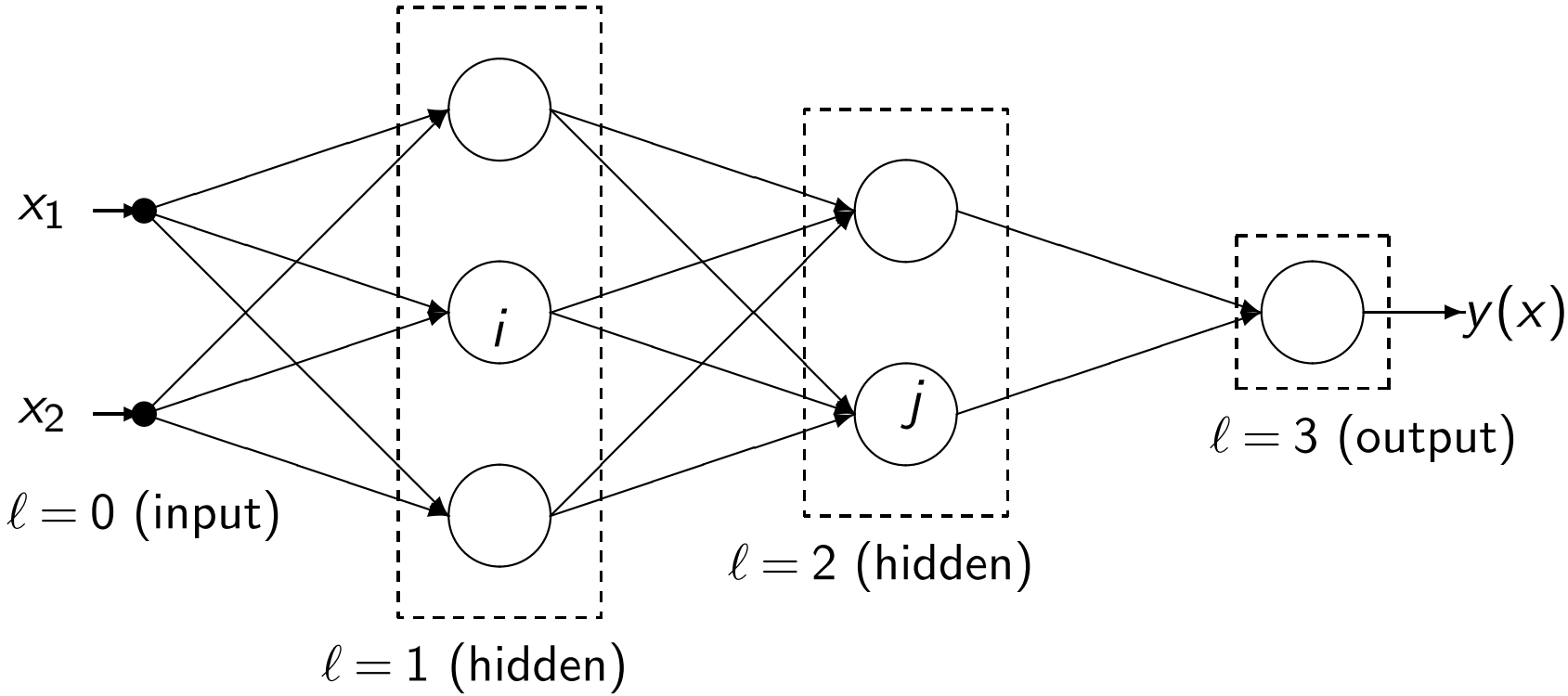
Backpropagation (BP) stands for a technique to evaluate the gradient

- BP *batch*, when the parameter are updated using all the samples in the training set T_t;

$$w^{k+1} = w^k - \eta \nabla E(w^k),$$

- BP *on-line*, when the parameters are updated using one sample of T_t at the time.

$$w^{k+1} = w^k - \eta \nabla E_{p(k)}(w^k).$$

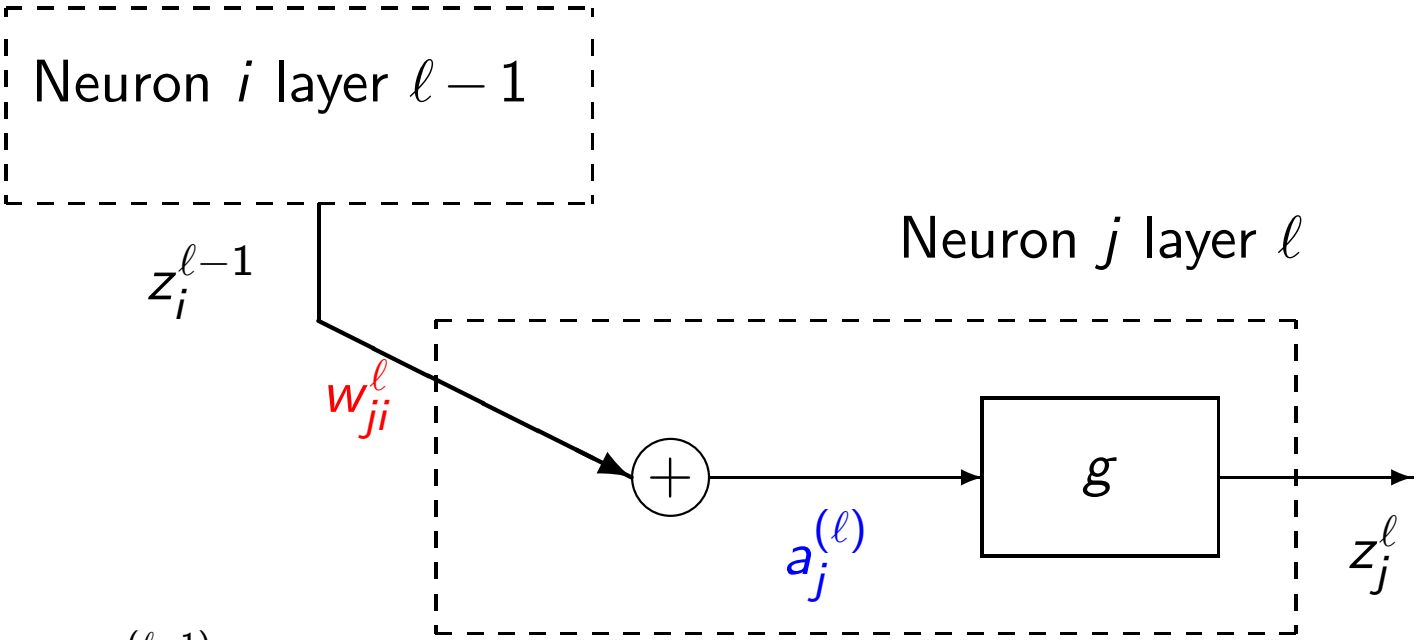


$$\nabla_w E(w) = \sum_{p=1}^{P} \nabla_w E_p(w) \qquad \nabla_w E_p(w) = \left\{ \frac{\partial E_p}{\partial w_{ji}^{(\ell)}} \right\}_{j,i,\ell}$$

SAPIENZA UNIVERSITÀ DI ROMA

Preliminaries

We assume that $g_j^{(\ell)}(\cdot) = g(\cdot)$ for all j, ℓ.



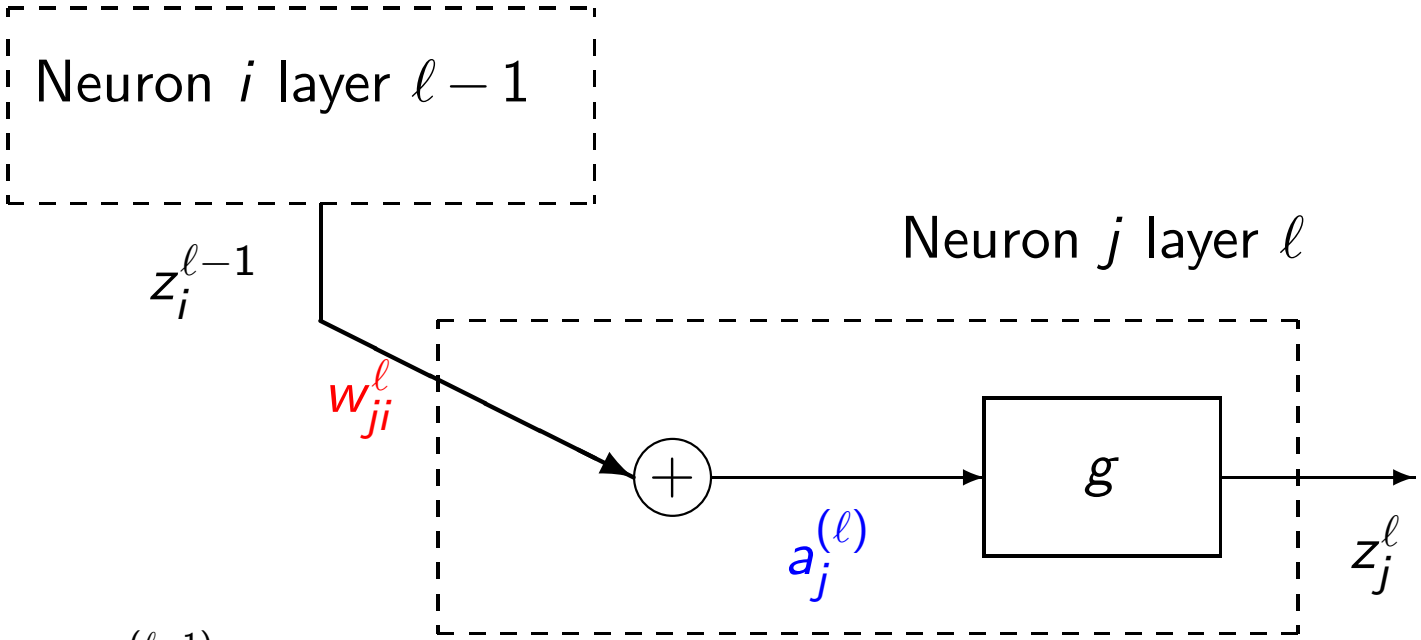
$$a_j^{(\ell)} = \sum_{k=0}^{N^{(\ell-1)}} w_{jk}^{(\ell)} z_k^{(\ell-1)} \qquad , \qquad z_j^{\ell} = g(a_j^{(\ell)})$$

$$\frac{\partial E_p}{\partial w_{ji}^{(\ell)}} = \frac{\partial E_p}{\partial a_j^{(\ell)}} \cdot \frac{\partial a_j^{(\ell)}}{\partial w_{ji}^{(\ell)}} =$$

SAPIENZA UNIVERSITÀ DI ROMA

Preliminaries

We assume that $g_j^{(\ell)}(\cdot) = g(\cdot)$ for all j, ℓ.

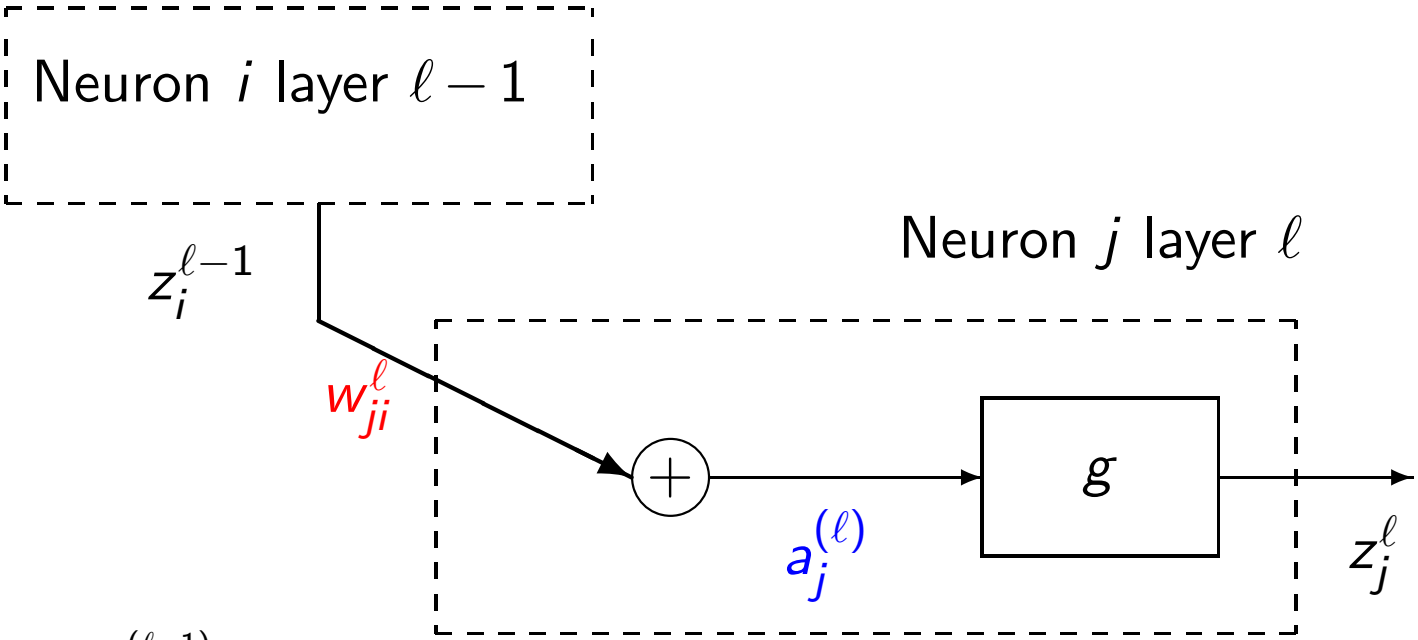


$$a_j^{(\ell)} = \sum_{k=0}^{N^{(\ell-1)}} w_{jk}^{(\ell)} z_k^{(\ell-1)} = (\cdots + w_{ji}^{(\ell)} z_i^{(\ell-1)} + \ldots), \qquad z_j^{\ell} = g(a_j^{(\ell)})$$

$$\frac{\partial E_p}{\partial w_{ji}^{(\ell)}} = \frac{\partial E_p}{\partial a_j^{(\ell)}} \cdot \frac{\partial a_j^{(\ell)}}{\partial w_{ji}^{(\ell)}} =$$

Preliminaries

We assume that $g_j^{(\ell)}(\cdot) = g(\cdot)$ for all j, ℓ.



$$a_j^{(\ell)} = \sum_{k=0}^{N^{(\ell-1)}} w_{jk}^{(\ell)} z_k^{(\ell-1)} = (\cdots + w_{ji}^{(\ell)} z_i^{(\ell-1)} + \ldots), \qquad z_j^{\ell} = g(a_j^{(\ell)})$$

$$\frac{\partial E_p}{\partial w_{ji}^{(\ell)}} = \frac{\partial E_p}{\partial a_j^{(\ell)}} \cdot \frac{\partial a_j^{(\ell)}}{\partial w_{ji}^{(\ell)}} = \frac{\partial E_p}{\partial a_j^{(\ell)}} \cdot z_i^{(\ell-1)} = \delta_j^{(\ell)} z_i^{(\ell-1)}$$

SAPIENZA UNIVERSITÀ DI ROMA

Forward propagation of the input

Forward computation of $z_i^{(\ell)}$

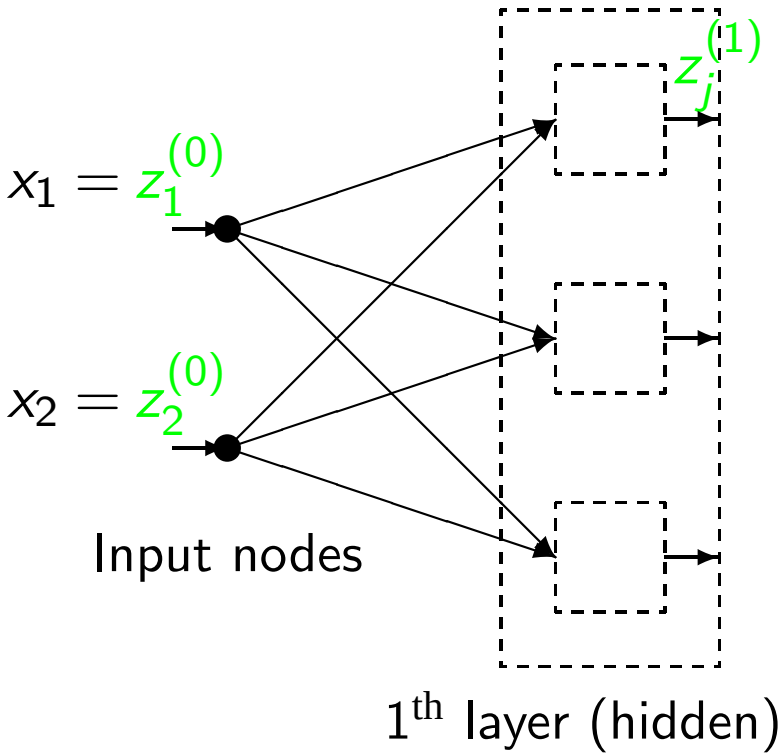
$x_1 = z_1^{(0)}$ 

$x_2 = z_2^{(0)}$ 

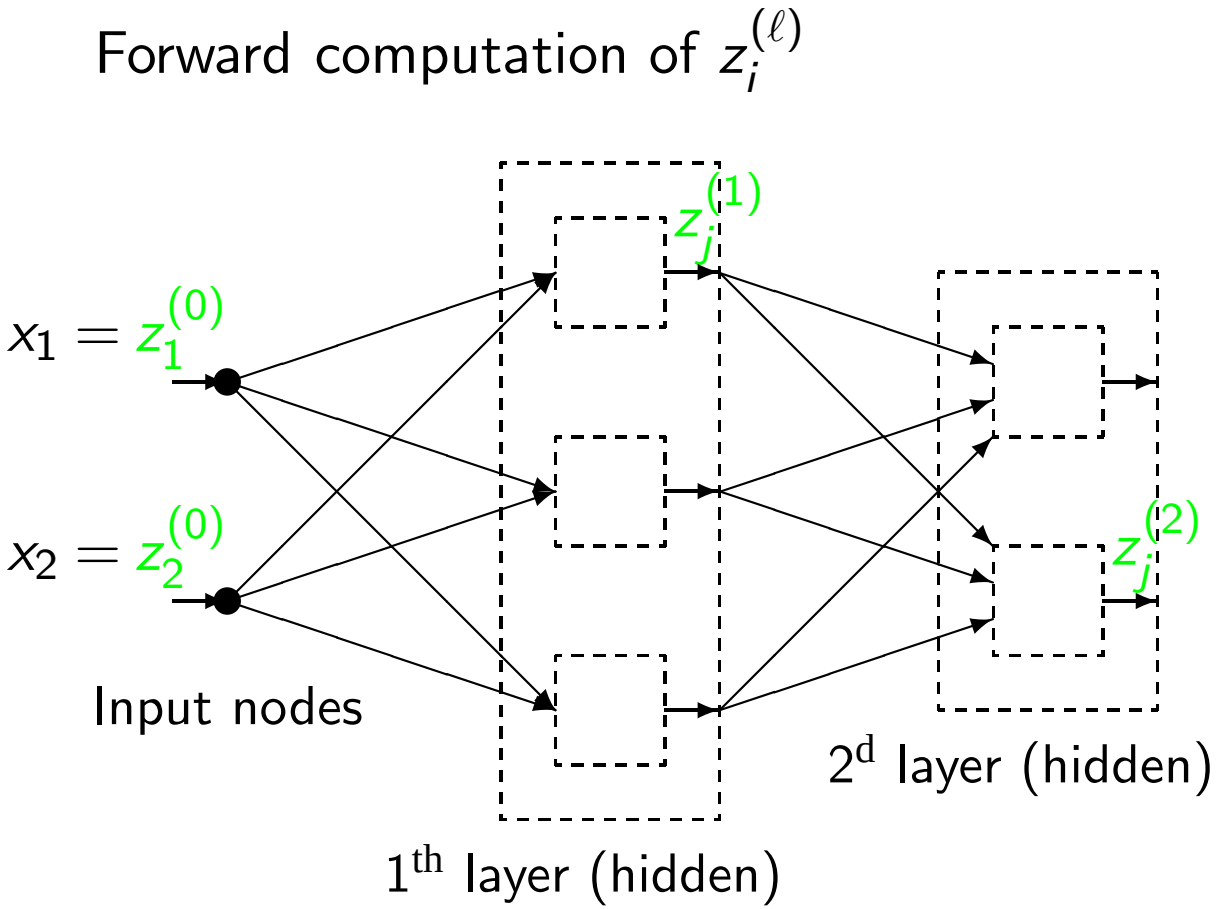
Input nodes

Forward propagation of the input

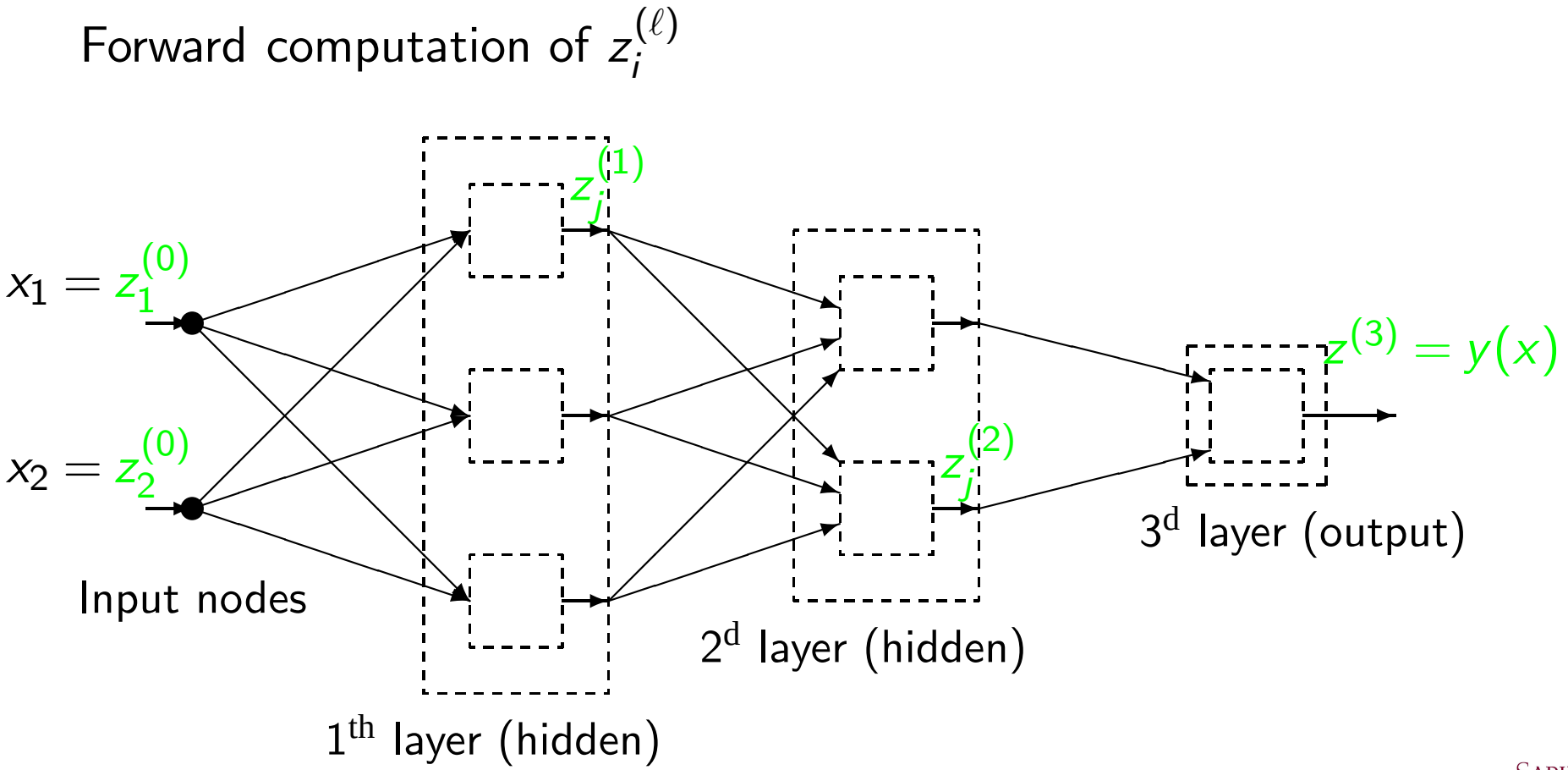
Forward computation of $z_i^{(\ell)}$



Forward propagation of the input

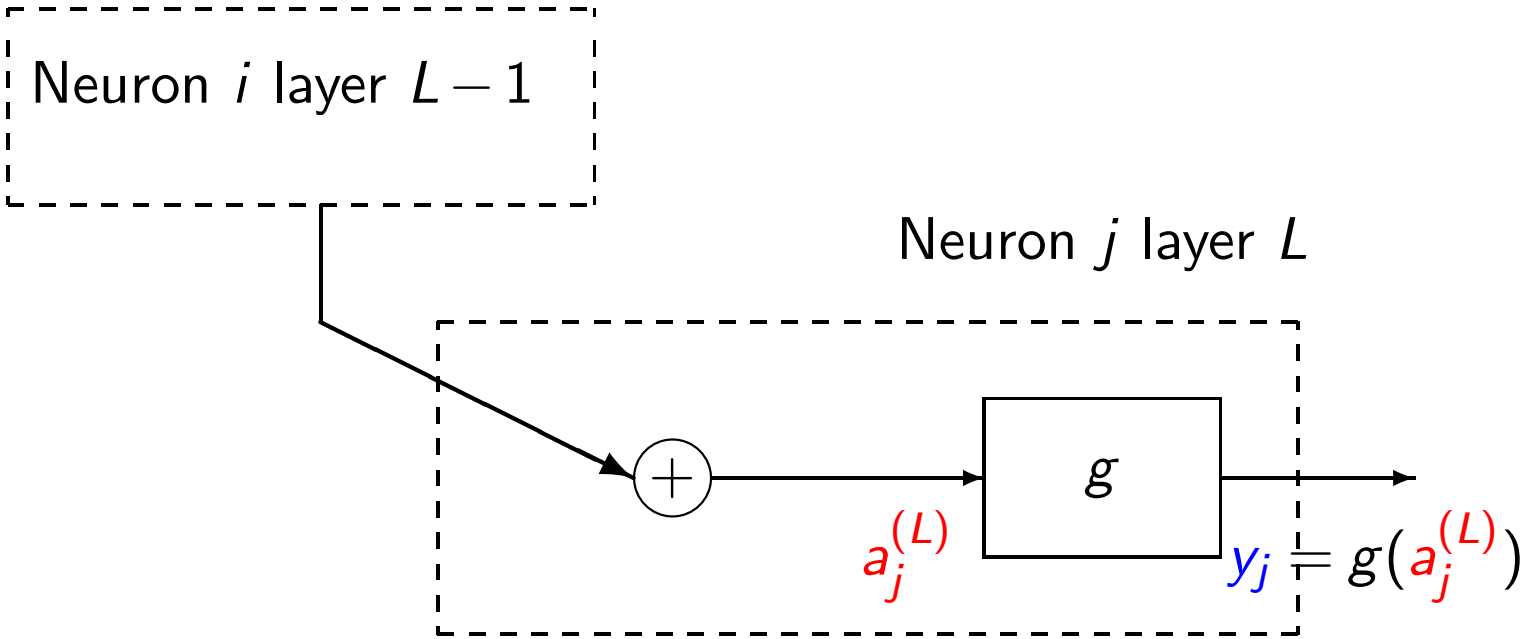


Forward propagation of the input



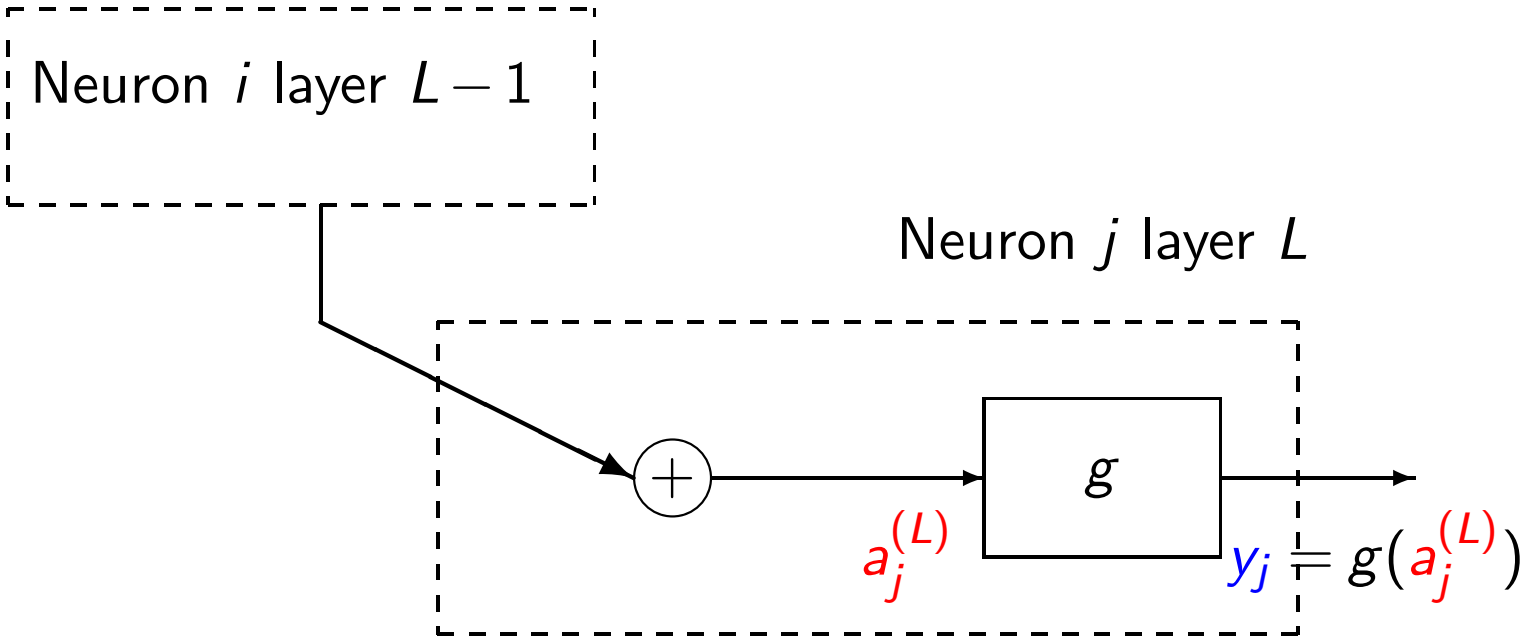
Back computation of errors $\delta_j^{(\ell)} = \frac{\partial E_p}{\partial a_j^{(\ell)}}$: output layer $\ell = L$

$$\delta_j^{(L)} = \frac{\partial E_p}{\partial a_j^{(L)}}$$



Back computation of errors $\delta_j^{(\ell)} = \frac{\partial E_p}{\partial a_j^{(\ell)}}$: output layer $\ell = L$

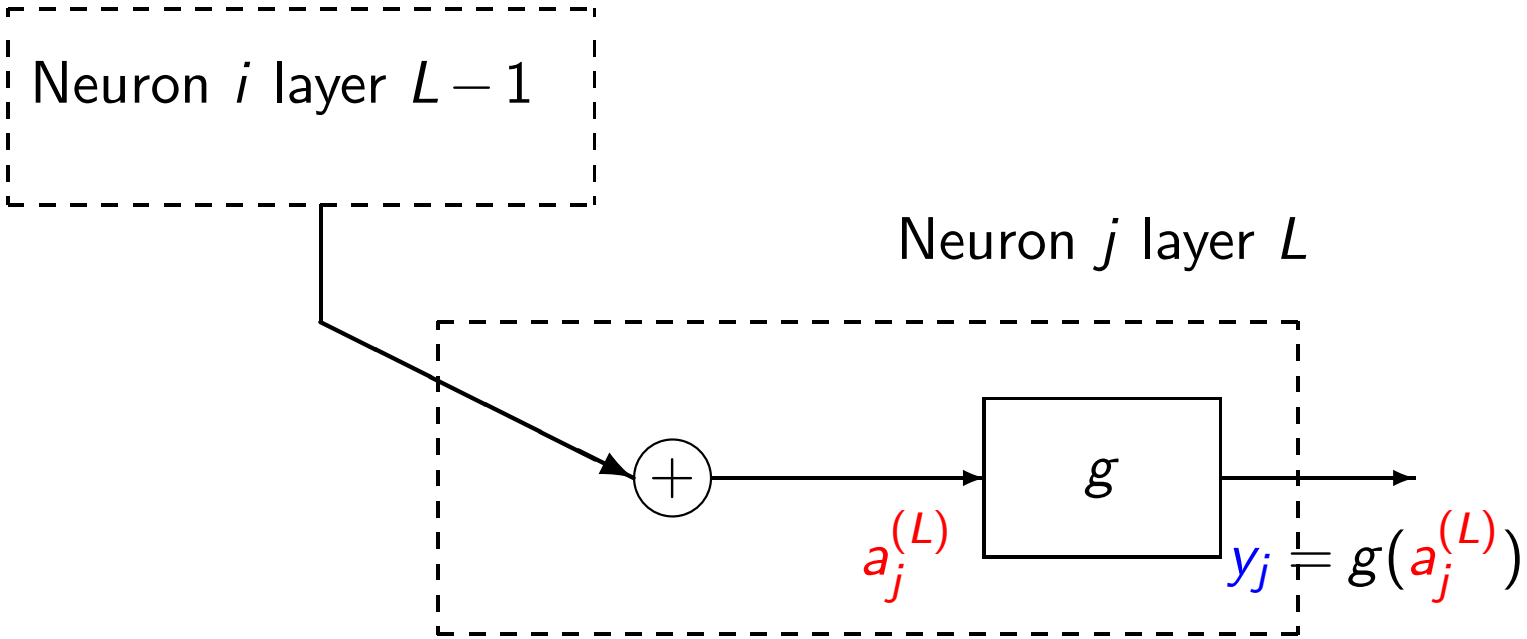
$$\delta_j^{(L)} = \frac{\partial E_p}{\partial a_j^{(L)}}$$



$$\delta_j^L = \frac{\partial E_p}{\partial a_j^{(L)}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial a_j{}^{(L)}} \qquad =$$

Back computation of errors $\delta_j^{(\ell)} = \frac{\partial E_p}{\partial a_j^{(\ell)}}$: output layer $\ell = L$

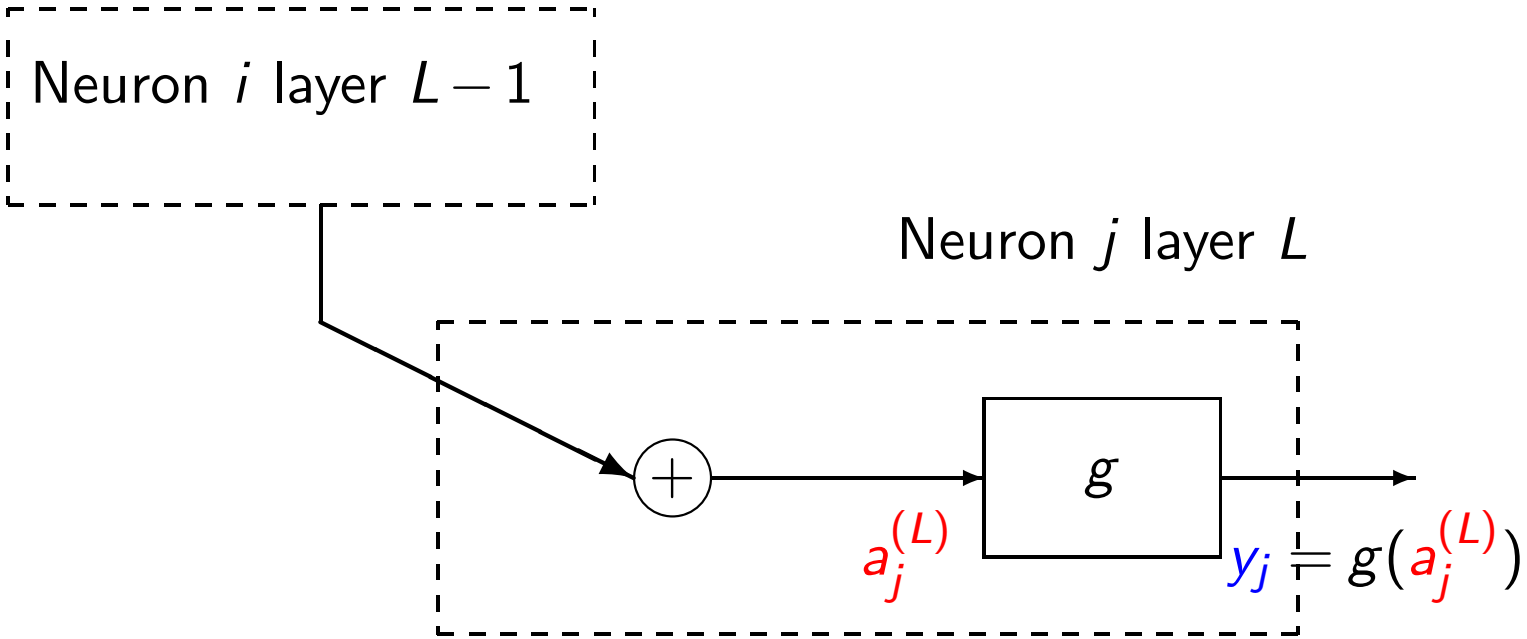
$$\delta_j^{(L)} = \frac{\partial E_p}{\partial a_j^{(L)}} \quad \left(E_p = \frac{1}{2}\|e_p\|^2 = \frac{1}{2}\sum_{j=1}^{K}(y_j(w;x^p) - y^p)^2) \right)$$



$$\delta_j^L = \frac{\partial E_p}{\partial a_j^{(L)}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial a_j{}^{(L)}} \qquad =$$

Back computation of errors $\delta_j^{(\ell)} = \frac{\partial E_p}{\partial a_j^{(\ell)}}$: output layer $\ell = L$

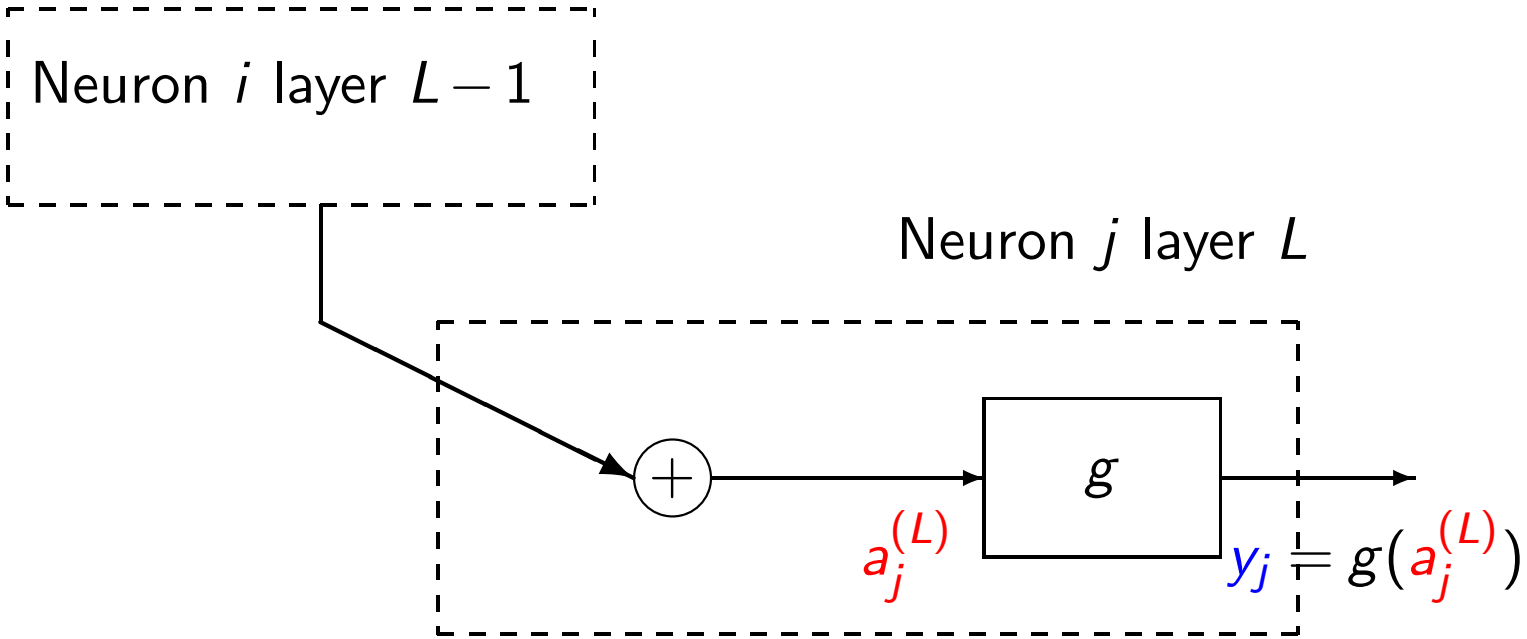
$$\delta_j^{(L)} = \frac{\partial E_p}{\partial a_j^{(L)}} \quad \left(E_p = \frac{1}{2}\|e_p\|^2 = \frac{1}{2}\sum_{j=1}^{K}(y_j(w;x^p) - y^p)^2) \right)$$



$$\delta_j^L = \frac{\partial E_p}{\partial a_j^{(L)}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial a_j{}^{(L)}} = \frac{\partial E_p}{\partial y_j} \cdot \dot{g}(a_j^{(L)}) =$$

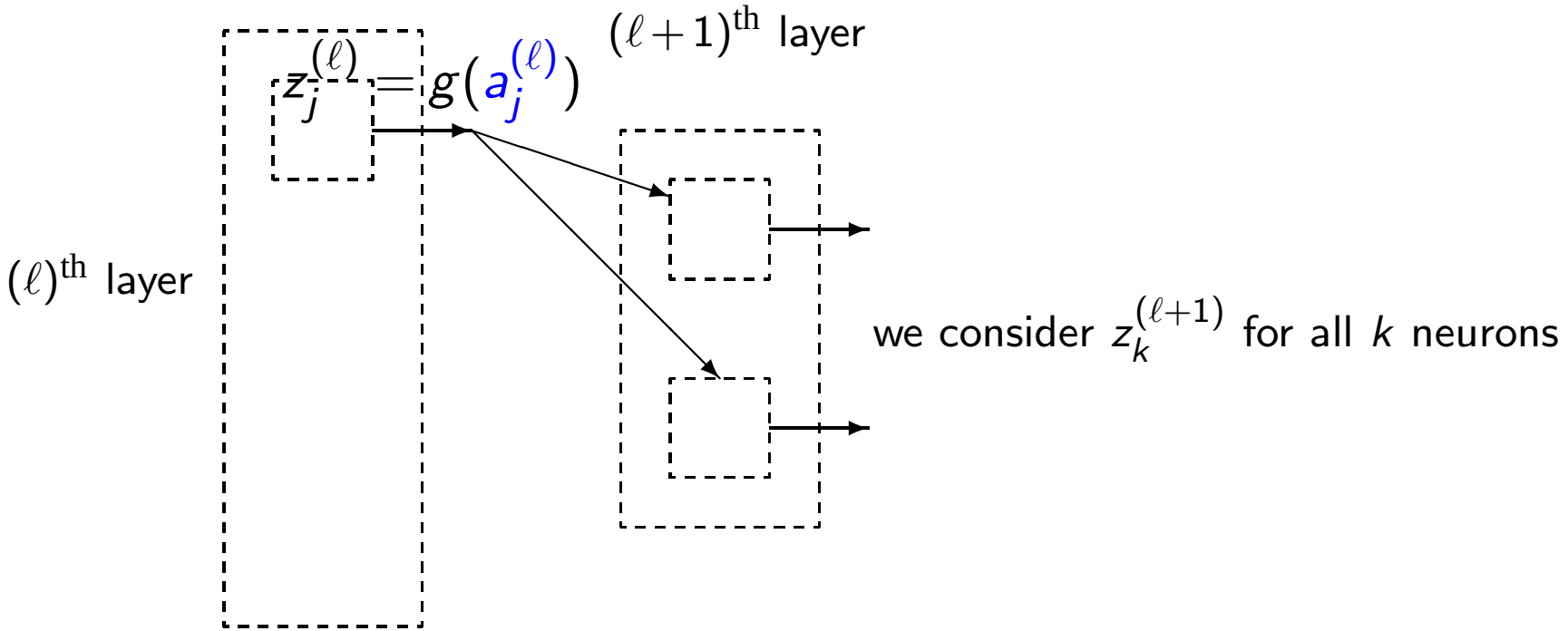
Back computation of errors $\delta_j^{(\ell)} = \frac{\partial E_p}{\partial a_j^{(\ell)}}$: output layer $\ell = L$

$$\delta_j^{(L)} = \frac{\partial E_p}{\partial a_j^{(L)}} \quad \left(E_p = \frac{1}{2}\|e_p\|^2 = \frac{1}{2}\sum_{j=1}^{K}(y_j(w;x^p) - y^p)^2) \right)$$

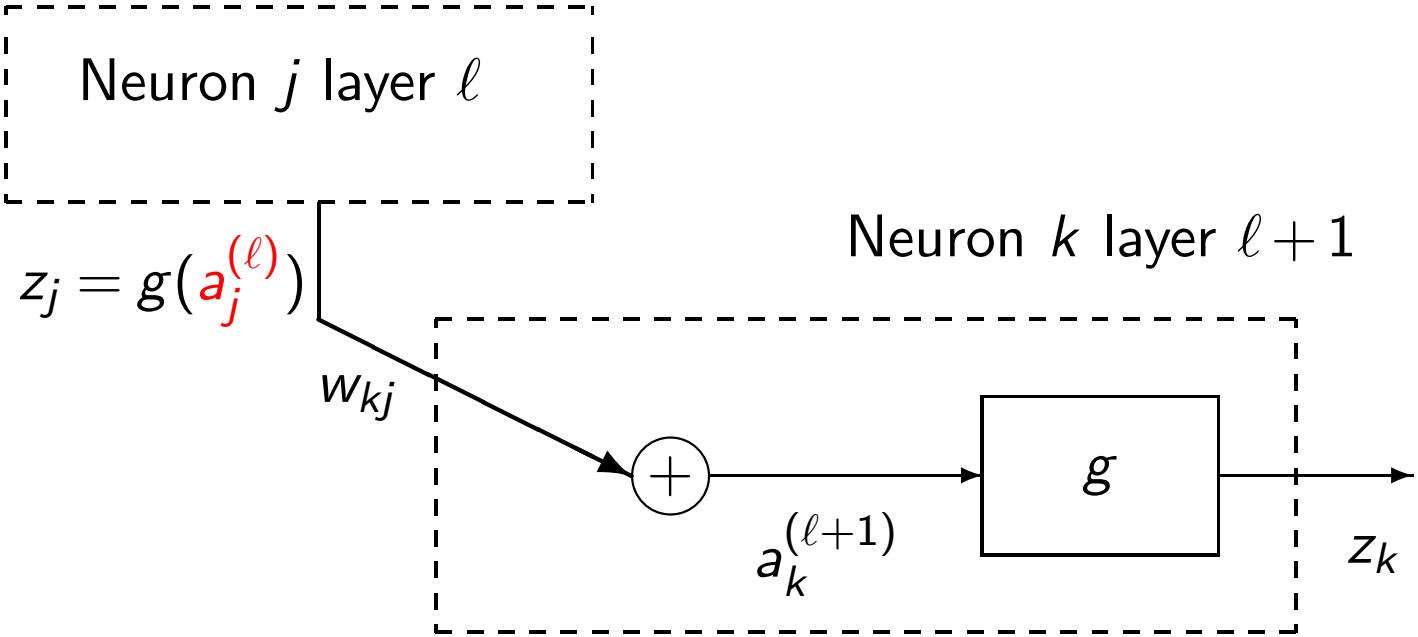


$$\delta_j^L = \frac{\partial E_p}{\partial a_j^{(L)}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial a_j{}^{(L)}} = \frac{\partial E_p}{\partial y_j} \cdot \dot{g}(a_j^{(L)}) = e_j^p(w)\dot{g}(a_j^{(L)})$$

Hidden layer $\ell < L$: $\dfrac{\partial E_p(w)}{\partial a_j^{(\ell)}}$



Back computation from $\delta_j^L = \frac{\partial E_p(w)}{\partial a_j^{(L)}}$ to $\delta_j^\ell = \frac{\partial E_p(w)}{\partial a_j^{(\ell)}}$ for $\ell < L$

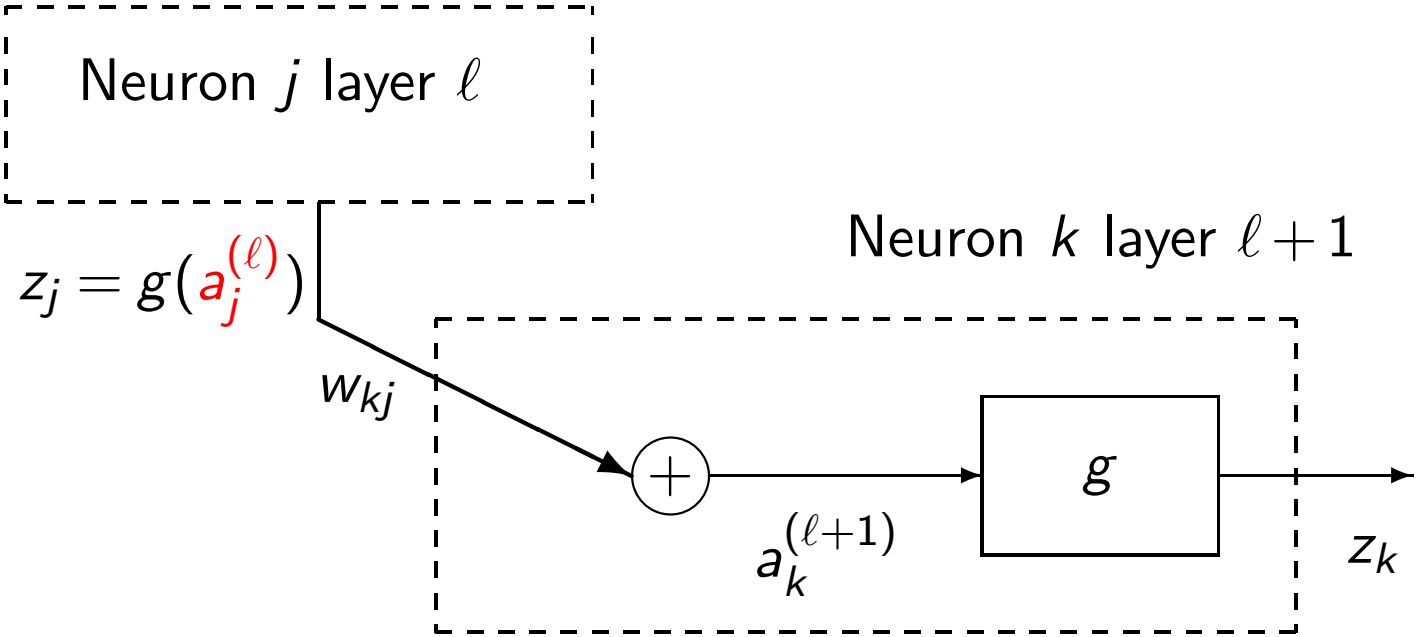


$$a_k^{(\ell+1)} = \ldots\cdots + w_{kj}^{(\ell+1)} g(a_j{}^{(\ell)}) + \ldots\ldots \qquad \text{for all } k = 1, \ldots N^{(\ell+1)}$$

$$\delta_j^\ell = \frac{\partial E_p}{\partial a_j{}^{(\ell)}} = \sum_{k=1}^{N^{(\ell+1)}} \frac{\partial E_p}{\partial a_k^{(\ell+1)}} \cdot \frac{\partial a_k^{(\ell+1)}}{\partial a_j{}^{(\ell)}}$$

SAPIENZA UNIVERSITÀ DI ROMA

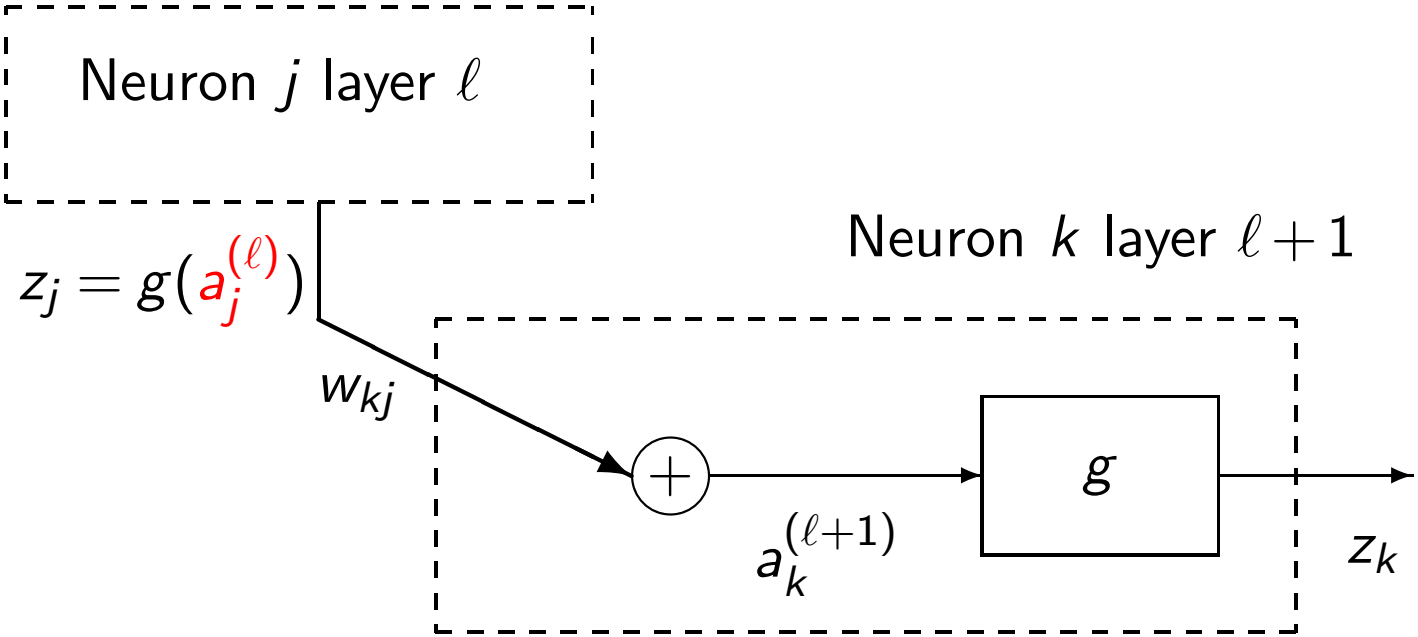
Back computation from $\delta_j^L = \frac{\partial E_p(w)}{\partial a_j^{(L)}}$ to $\delta_j^\ell = \frac{\partial E_p(w)}{\partial a_j^{(\ell)}}$ for $\ell < L$



$$a_k^{(\ell+1)} = \ldots\cdots + w_{kj}^{(\ell+1)} g(a_j{}^{(\ell)}) + \ldots\ldots \qquad \text{for all } k = 1, \ldots N^{(\ell+1)}$$

$$\delta_j^\ell = \frac{\partial E_p}{\partial a_j{}^{(\ell)}} = \sum_{k=1}^{N^{(\ell+1)}} \frac{\partial E_p}{\partial a_k^{(\ell+1)}} \cdot \frac{\partial a_k^{(\ell+1)}}{\partial a_j{}^{(\ell)}} = \sum_{k=1}^{N^{(\ell+1)}} \delta_k^{(\ell+1)} \cdot \frac{\partial a_k^{(\ell+1)}}{\partial a_j{}^{(\ell)}}$$

Back computation from $\delta_j^L = \frac{\partial E_p(w)}{\partial a_j^{(L)}}$ to $\delta_j^\ell = \frac{\partial E_p(w)}{\partial a_j^{(\ell)}}$ for $\ell < L$



$$a_k^{(\ell+1)} = \ldots\cdots + w_{kj}^{(\ell+1)} g(a_j{}^{(\ell)}) + \ldots\ldots \qquad \text{for all } k = 1, \ldots N^{(\ell+1)}$$

$$\delta_j^\ell = \frac{\partial E_p}{\partial a_j{}^{(\ell)}} = \sum_{k=1}^{N^{(\ell+1)}} \frac{\partial E_p}{\partial a_k^{(\ell+1)}} \cdot \frac{\partial a_k^{(\ell+1)}}{\partial a_j{}^{(\ell)}} = \sum_{k=1}^{N^{(\ell+1)}} \delta_k^{(\ell+1)} \cdot \frac{\partial a_k^{(\ell+1)}}{\partial a_j{}^{(\ell)}} = \sum_{k=1}^{N^{(\ell+1)}} \delta_k^{(\ell+1)} w_{kj}^{(\ell+1)} g'$$

Backpropagation gradient evaluation

1. Compute FORWARD

$$z_i^\ell \qquad \ell = 1, \dots, L$$

2. Compute BACKWARD

$$\delta_k^{(L)} = \frac{\partial E_p}{\partial a_k^{(L)}} = e_k^p \cdot \dot{g}(a_k^{(L)}) \ \forall \ k = 1, \dots K$$

For $\ell = L, \dots, 1$

$$\delta_j^{(\ell)} = \sum_{k=1}^{N^{(\ell+1)}} \delta_k^{(\ell+1)} \cdot w_{kj}^{(\ell+1)} \dot{g}(a_j{}^{(\ell)}) \qquad j = 1, \dots, N^{(\ell)}$$

Set

$$\frac{\partial E_p}{\partial w_{ji}{}^{(\ell)}} = \delta_j^{(\ell)} \cdot z_i^{(\ell-1)} \qquad \ell = 1, \dots, L$$

Convergence

Theorem

Assume that a scalar $L > 0$ exists such that for each $w, u \in R^m$ we have:

$$\|\nabla E(w) - \nabla E(u)\| \leq L\|w - u\|$$

(Lipschtizt continuity of the gradient).
Let $\{w^k\}$ be the sequence generated by

$$w^{k+1} = w^k - \eta \nabla E(w^k)$$

with $\varepsilon \leq \eta \leq \bar{\eta}_L - \varepsilon$, and $\varepsilon > 0$,
Assume $\nabla E(w^k) \neq 0$ for all k, then every accumulation point of $\{w^k\}$ is a stationary point for E.
If the level set $\mathscr{L}^0$ is compact, there exists accumulation point of $\{w^k\}$).

Proof BP

From the Lipschitz assumption

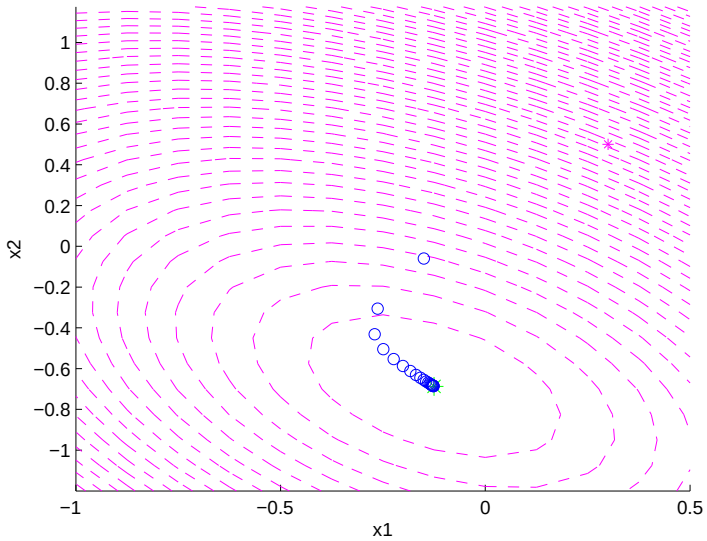
$$\begin{aligned} E(w^k + \eta d^k) &= E(w^k) + \eta \int_0^1 \nabla E(w^k + t\eta d^k)^T d^k \; dt \\ &\leq E(w^k) + \frac{\eta^2 L}{2} ||d^k||^2 + \eta \nabla E(w^k)^T d^k \\ &= E(w^k) - \eta \left(1 - \frac{\eta L}{2}\right) \|\nabla E(w^k)\|^2 \end{aligned}$$

For $0 < \eta < \frac{2}{L}$, we have

- $E(w^{k+1}) < E(w^k)$ hence $\{w^k\} \in \mathscr{L}$, $\exists K \rightsquigarrow \{w^k\}_K \to \widehat{w}$.
- $\{E(w^k)\} \to \bar{E}$ because it is monotonically decreasing bounded below
- $\lim_{k\to\infty} \nabla E(w^k) = 0$.

How FG works

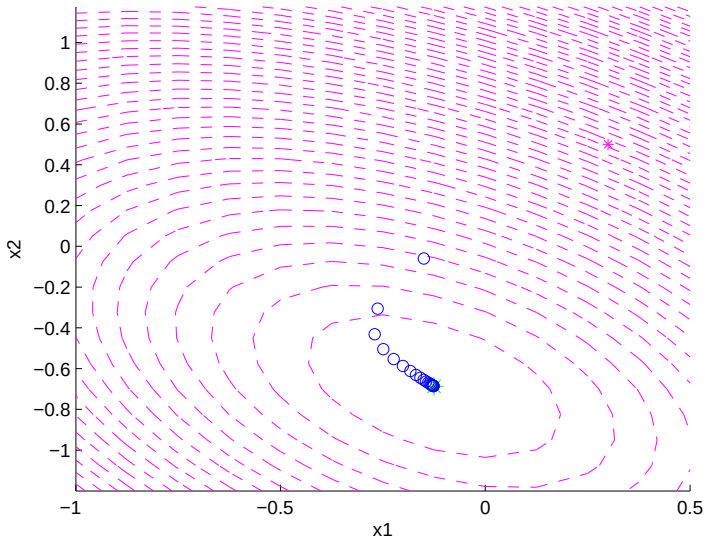
Strictly convex quadratic with constant stepsize

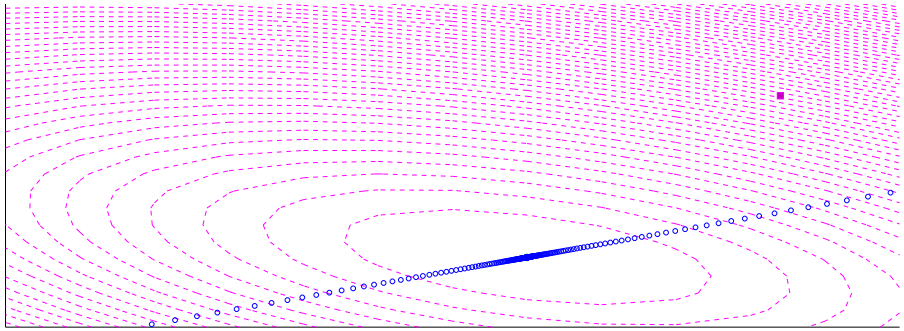


SAPIENZA
UNIVERSITÀ DI ROMA

How FG works

Strictly convex quadratic with constant stepsize





$\eta^k = 0.3$, $\sharp_{it} = 278$

The choice of the stepsize (*learning rate*) is crucial

Momentum modification

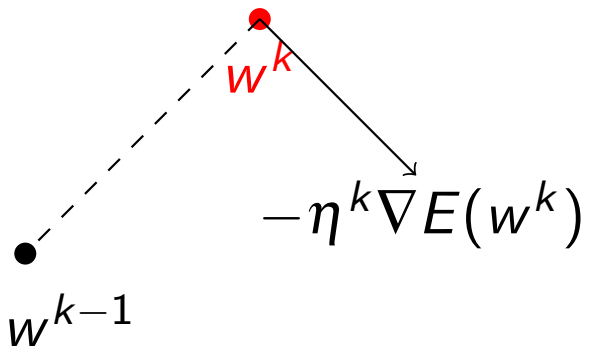
$$w^{k+1} = w^k - \eta \nabla E(w^k) + \beta (w^k - w^{k-1}),$$

where $\beta > 0$ is a given scalar with (typical values $= 0.8 \pm 0.9$).

Momentum term

The momentum term represents an extrapolation along the difference of the two preceding iterates (*heavy ball method*).

$$w^{k+1} = w^k - \eta^k \nabla E(w^k) +$$

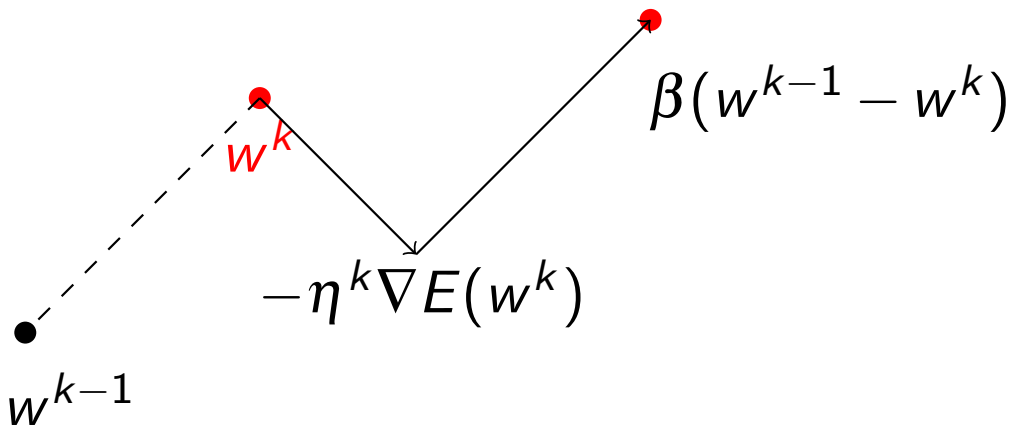


Momentum term

The momentum term represents an extrapolation along the difference of the two preceding iterates (*heavy ball method*).

$$w^{k+1} = w^k - \eta^k \nabla E(w^k) + \beta^k (w^k - w^{k-1})$$

with $w^{-1} = w^0$.

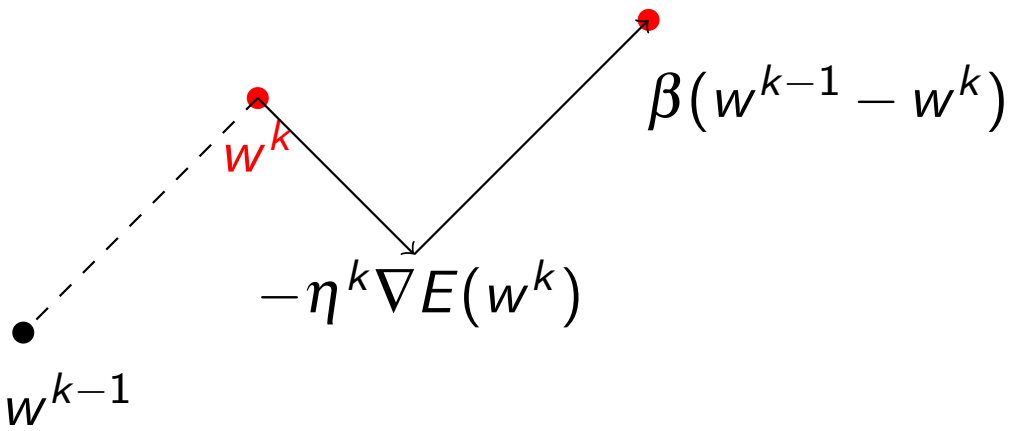


Momentum term

The momentum term represents an extrapolation along the difference of the two preceding iterates (*heavy ball method*).

$$w^{k+1} = w^k - \eta^k \nabla E(w^k) + \beta^k (w^k - w^{k-1})$$

with $w^{-1} = w^0$.



If $\beta^k = \beta$ and $\eta^k = \eta$ we get

$$w^{k+1} = w^k - \eta \sum_{\ell=0}^{k} (\beta)^\ell \nabla E(w^{k-\ell})$$