Gradient-based optimization methods for neural network training

Ph.D. in Industrial Engineering
Università di Firenze, A.A. 2020/2021

Simone Rebegoldi
Dipartimento di Ingegneria Industriale
Università di Firenze

Neural network: ϕ_i-evaluation

$$\begin{cases} \mathbf{v}_1 = a \\ \mathbf{v}_j = \boldsymbol{\sigma}_j \left(\mathbf{W}_j \mathbf{v}_{j-1} + \beta_j \right) & j = 2, \ldots, m \quad m \text{ layers} \end{cases}$$

We have to minimize the empirical risk:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} \underbrace{\ell(\mathbf{v}_m(x; \mathbf{a}_i), \mathbf{b}_i)}_{\phi_i} \qquad x = \left(\mathrm{vec}(\boldsymbol{W}_2), \boldsymbol{\beta}_2^{\mathrm{T}}, \ldots, \mathrm{vec}(\boldsymbol{W}_m), \boldsymbol{\beta}_m^{\mathrm{T}} \right)^{\mathrm{T}},$$

$$W_j = \begin{bmatrix} w_{j,1,1} & w_{j,1,2} & \cdots & \cdots & w_{j,1,q_j} \\ w_{j,2,1} & w_{j,2,2} & \cdots & \cdots & w_{j,2,q_j} \\ \vdots & \vdots & & & \vdots \\ w_{j,q_{j-1},1} & w_{j,q_j-1,2} & \cdots & \cdots & w_{j,q_{j-1},q_j} \end{bmatrix}$$

Forward propagation Algorithm

Input: $m \geqslant 2$, $\boldsymbol{W}_j$, $\boldsymbol{\beta}_j$, $\boldsymbol{\sigma}_j$ for $j = 2, \ldots, m$, $\ell\ (a_i, b_i)$ sample-label
Output: $L = \phi_i(x)$
Step 0: $\boldsymbol{v}_1 \geqslant tsa_i$
Inner loop:
For $j = 2, \ldots, m$
$\quad r_j \leftarrow \boldsymbol{\beta}_j + \boldsymbol{W}_j \boldsymbol{v}_{j-1}$ $\qquad$ m matrix-vector products
$\quad \boldsymbol{v}_j \leftarrow \boldsymbol{\sigma}_j(\boldsymbol{r}_j)$

$L \leftarrow \ell(\boldsymbol{v}_m, b_i)$

Neural network: $\nabla\phi$-evaluation

$$\begin{cases} \mathbf{v}_1 = a \\ \mathbf{v}_j = \boldsymbol{\sigma}_j\left(\mathbf{W}_j\mathbf{v}_{j-1} + \beta_j\right) \quad j = 2,\dots,m \end{cases} \qquad \phi(x) = \ell(\mathbf{v}_m(x;\mathbf{a}_i), \mathbf{b}_i)$$

Backward propagation Algorithm

Input: $m \geqslant 2$, $\boldsymbol{W}_j$, $\boldsymbol{\beta}_j$, $\boldsymbol{\sigma}_j$ for $j = 2,\dots,m$, ℓ, (a_i, b_i) sample-label,
L, $\boldsymbol{v}_j$ e $\boldsymbol{r}_j$, $j = 2,\dots,m$ computed by the Forward propagation algorithm
Output: $\tilde{\nabla}_{\boldsymbol{W}_j}\phi_i(x)$ e $\nabla_{\boldsymbol{\beta}_j}\phi_i(x)$ for $j = 2,\dots,m$
Step 0: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{v}_m} L = \nabla_{\boldsymbol{v}_m}\ell(\boldsymbol{v}_m, b_i)$
Inner loop:
For $j = m, m-1,\dots,2$

$\boldsymbol{g} \leftarrow \left(\frac{d\boldsymbol{\sigma_j}(\boldsymbol{r}_j)}{d\boldsymbol{r}_j}\right)^{\mathrm{T}}\boldsymbol{g}$ DERIVATIVES CHAIN-RULE

$\tilde{\nabla}_{\boldsymbol{W}_j}\phi_i(x) = \boldsymbol{g}\,\boldsymbol{v}_{j-1}^{\mathrm{T}}$ $m-1$ matrix-vector products

$\nabla_{\boldsymbol{\beta}_j}\phi_i(x) = \boldsymbol{g}$

$\boldsymbol{g} \leftarrow= \boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{g}$

where $\tilde{\nabla}_{\boldsymbol{W}_j}\phi(x) = mat(\nabla_{vec(\boldsymbol{W}_j)}\phi(x))$.

Binary classification problems - Logistic loss and derivative

The logistic loss with ℓ_2 regularization :

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + e^{-b_i a_i^T x}) + \frac{1}{2N}\|x\|^2$$

Given $\hat{x}$ resulting from the classifier training, $\hat{a} \in \mathbb{R}^n$

$$\frac{1}{1+e^{-\hat{a}^T\hat{x}}} \geqslant 0.5 \quad \hat{b} = 1$$
$$\frac{1}{1+e^{-\hat{a}^T\hat{x}}} < 0.5, \quad \hat{b} = -1$$

- Labels in {-1,1}

$$\phi_i(x) = \log(1 + c(a_i, b_i; x)) + \frac{1}{2}\|x\|^2$$

where $c(a_i, b_i; x) = e^{-b_i a_i^T x}$

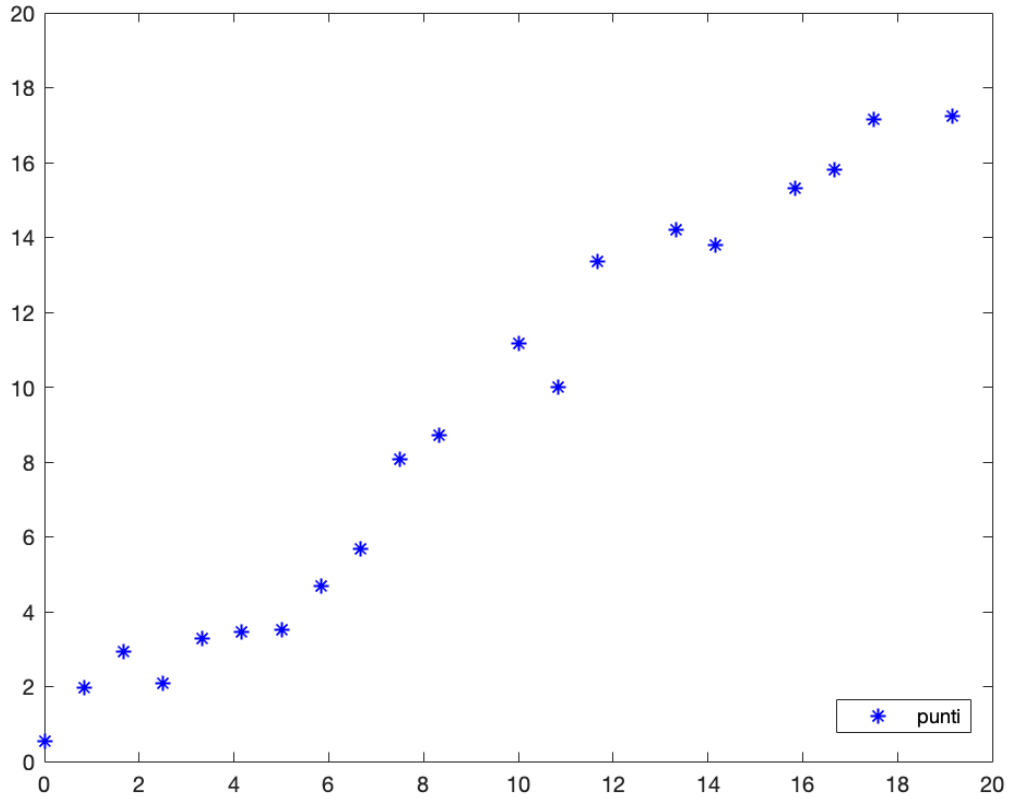
- $$\nabla\phi_i(x) = -\frac{c(a_i, b_i; x)}{1 + c(a_i, b_i; x)} b_i a_i + x = (\frac{b_i}{1 + c(a_i, b_i; x)} - b_i)a_i + x$$

- Epoch: each set of N consecutives evaluations of a sample gradient $\nabla\phi_i(x)$.

Overfitting

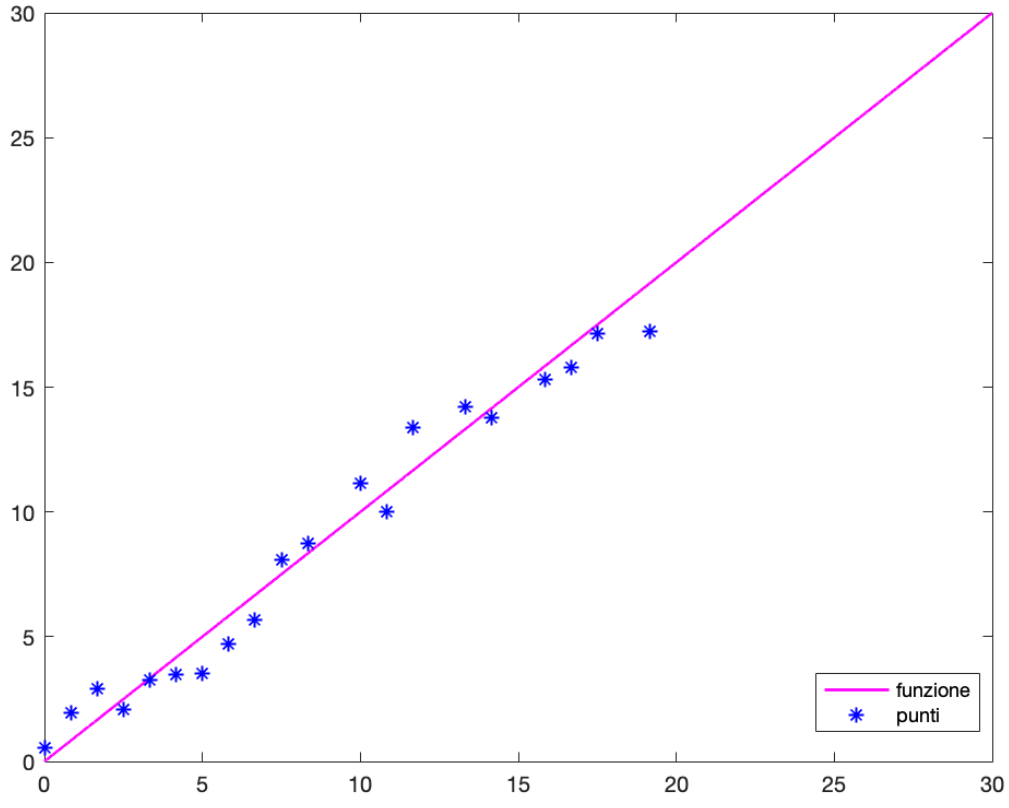
A regression problem

Dataset $(a_i, b_i) \in \mathbb{R} \times \mathbb{R}$, $N = 20$, b_i are values affected by noise



Overfitting

Linear model -square loss

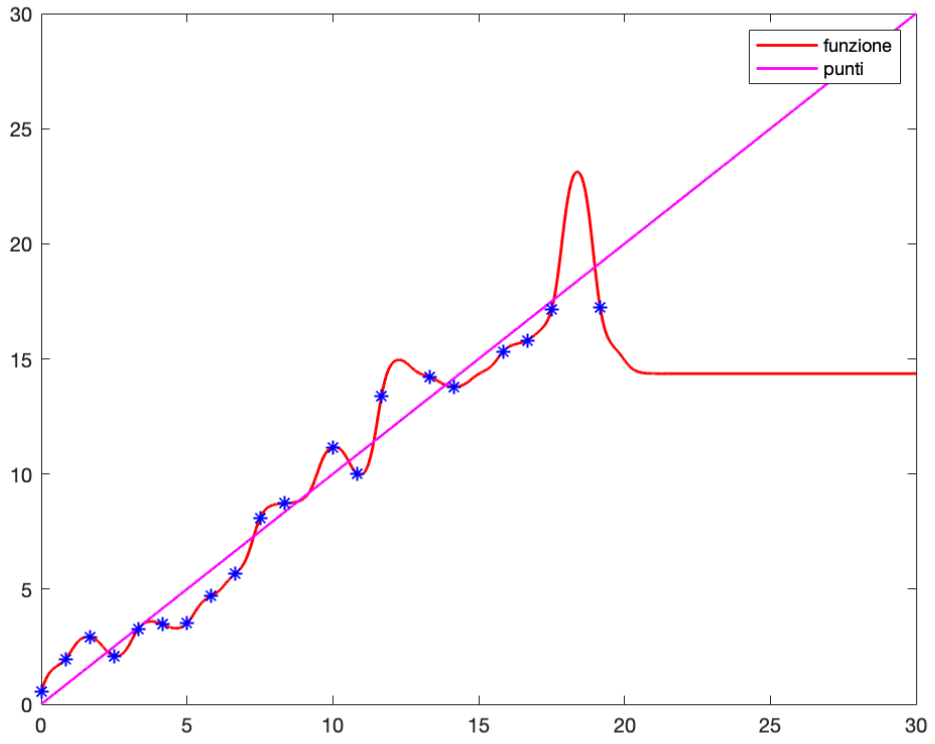


Overfitting

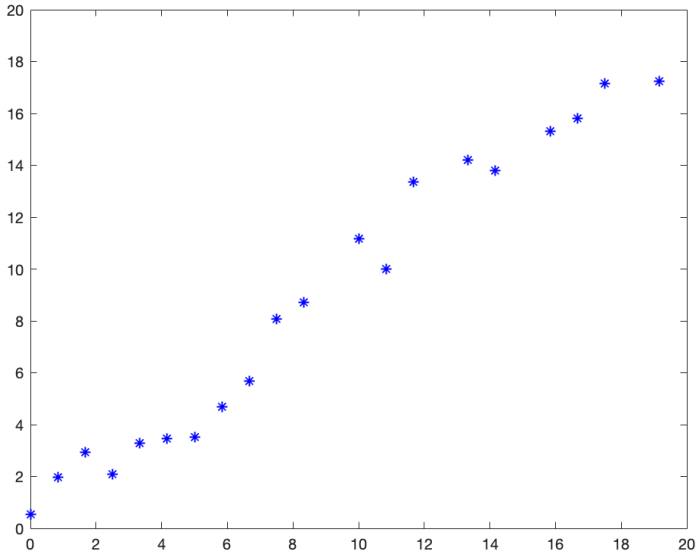
Neural Network-square loss

$$\begin{array}{ccccc} 1 & - & 10 & - & 1 \\ \text{input} & & \text{hidden} & & \text{output} \end{array}$$

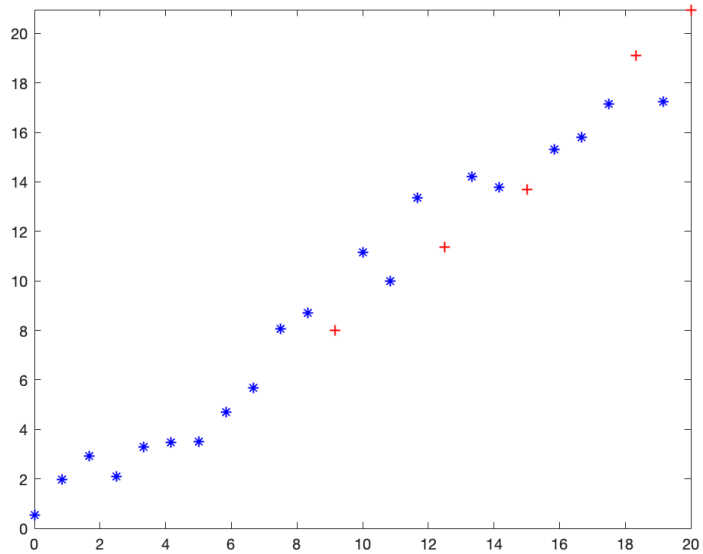
20 weights, 11 bias, 31 parameters.



Overfitting - Validation set



(a) Training set $\mathcal{T}$

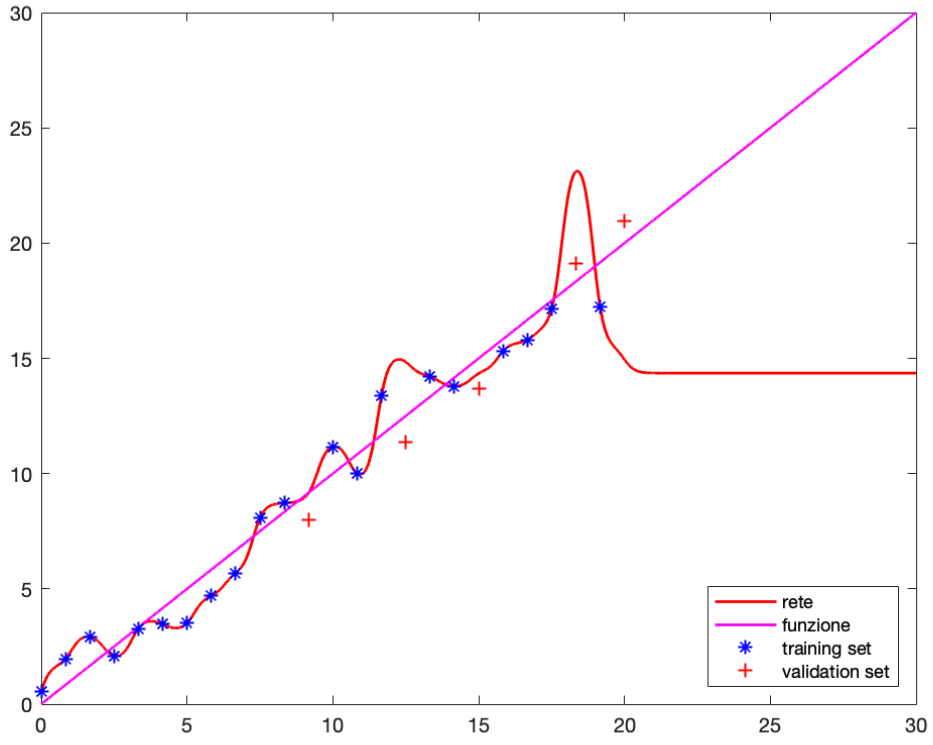


(b) Training set $\mathcal{T}$ + Validation set $\mathcal{V}$

The data-set is split in

- training set (blu)
- validation set (red)
- testing set

Overfitting - training error and validation erro

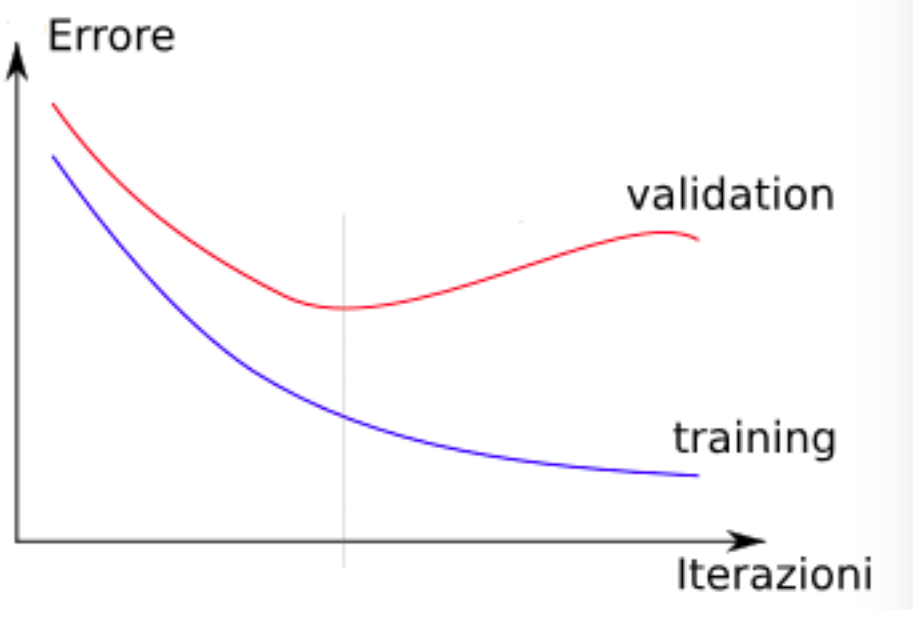


Errors

- $err_{\mathcal{T}}(x) = \frac{1}{|\mathcal{T}|} \sum_{(a_j,b_j)\in\mathcal{T}} (\mathbf{v_3}(x;a_j) - b_j)^2$
- $err_{\mathcal{V}}(x) = \frac{1}{|\mathcal{V}|} \sum_{(a_j,b_j)\in\mathcal{V}} (\mathbf{v_3}(x;a_j) - b_j)^2$

Overfitting - stopping criterion

- $err_{\mathcal{T}}(x) = \frac{1}{|\mathcal{T}|} \sum_{(a_j,b_j)\in\mathcal{T}} (\mathbf{v_3}(x;a_j) - b_j)^2$ (in blu)
- $err_{\mathcal{V}}(x) = \frac{1}{|\mathcal{V}|} \sum_{(a_j,b_j)\in\mathcal{V}} (\mathbf{v_3}(x;a_j) - b_j)^2$ (in rosso)



Stopping criterion: Training is stopped when the validation error increases.