OTTIMIZZAZIONE DI RETI NEURALI ARTIFICIALI: COME SFUGGIRE AI PUNTI DI SELLA?

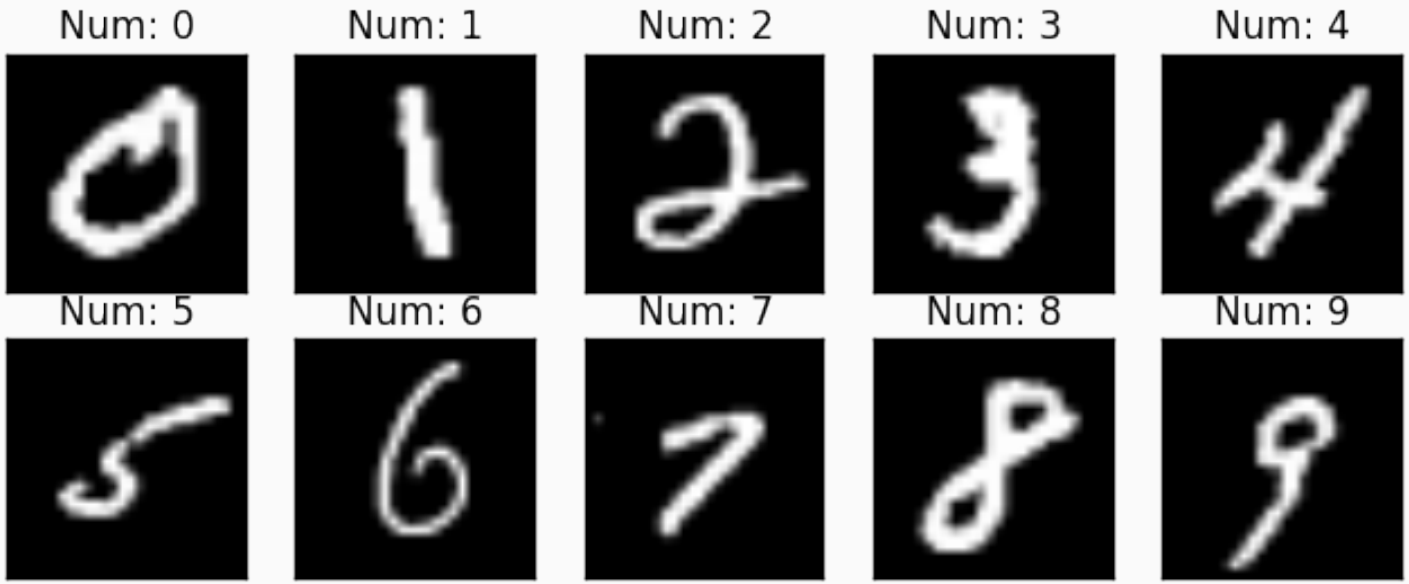
Niccolò Ajroldi
Matricola 846266

Relatore: Danilo Ardagna

25 Febbraio 2019

Politecnico di Milano - Ingegneria Matematica

Un problema di classificazione



Come costruire una **funzione** che sappia classificare correttamente le immagini?

Image source: http://yann.lecun.com/exdb/mnist/, Yann LeCun, Corinna Cortes, Christopher J.C. Burges

Neural network

Rete di neuroni interconnessi, ciascuno elabora i pixel dell'immagine ed emette un output che ne rappresenta l'attivazione.

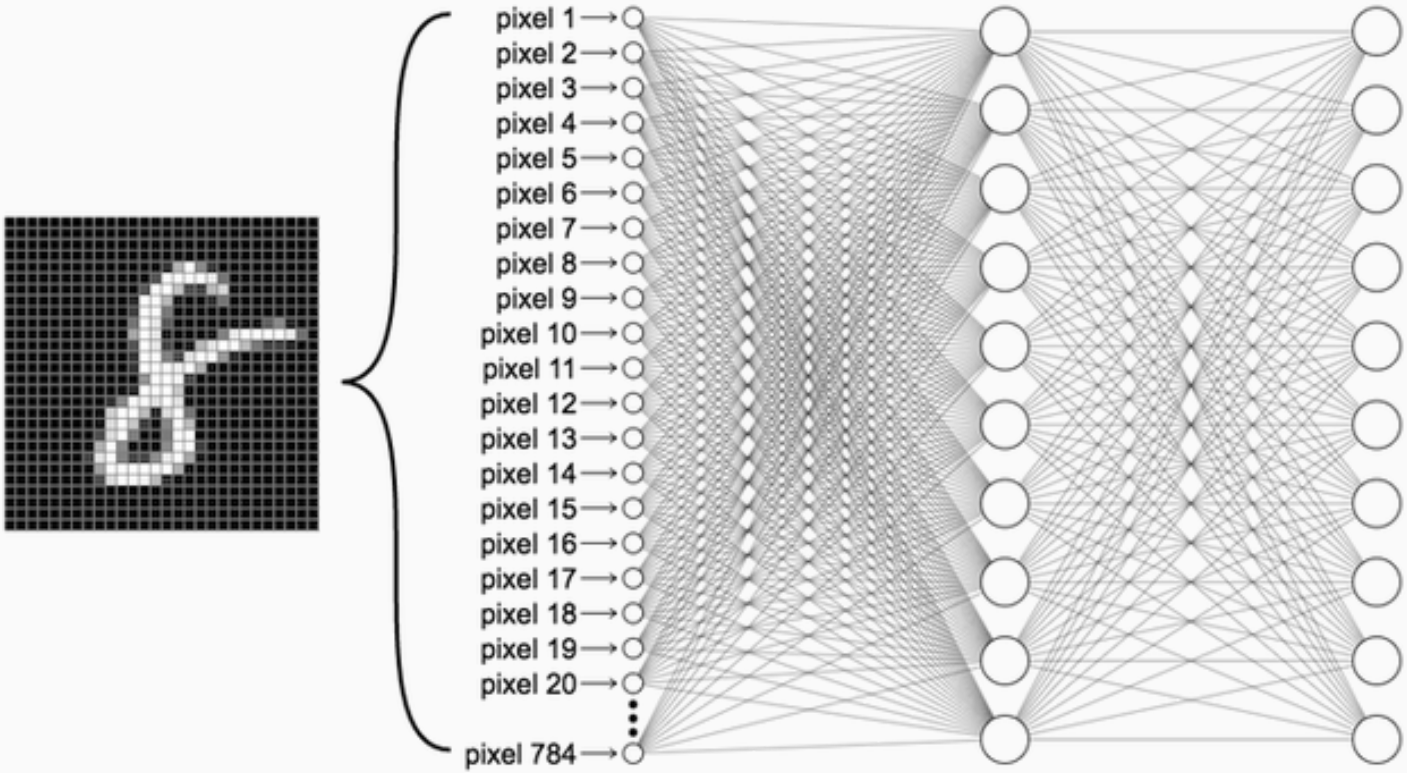


Image source:

https://achintavarna.wordpress.com/2017/11/17/keras-tutorial-for-beginners-a-simple-neural-network-to-identify-numbers-mnist-data/

Neural Network

Risposta della rete

$$\underline{\hat{y}} = \begin{bmatrix} 0.241 \\ 0.084 \\ 0.316 \\ 0.107 \\ 0.064 \\ 0.381 \\ 0.430 \\ 0.024 \\ 0.879 \\ 0.109 \end{bmatrix}$$

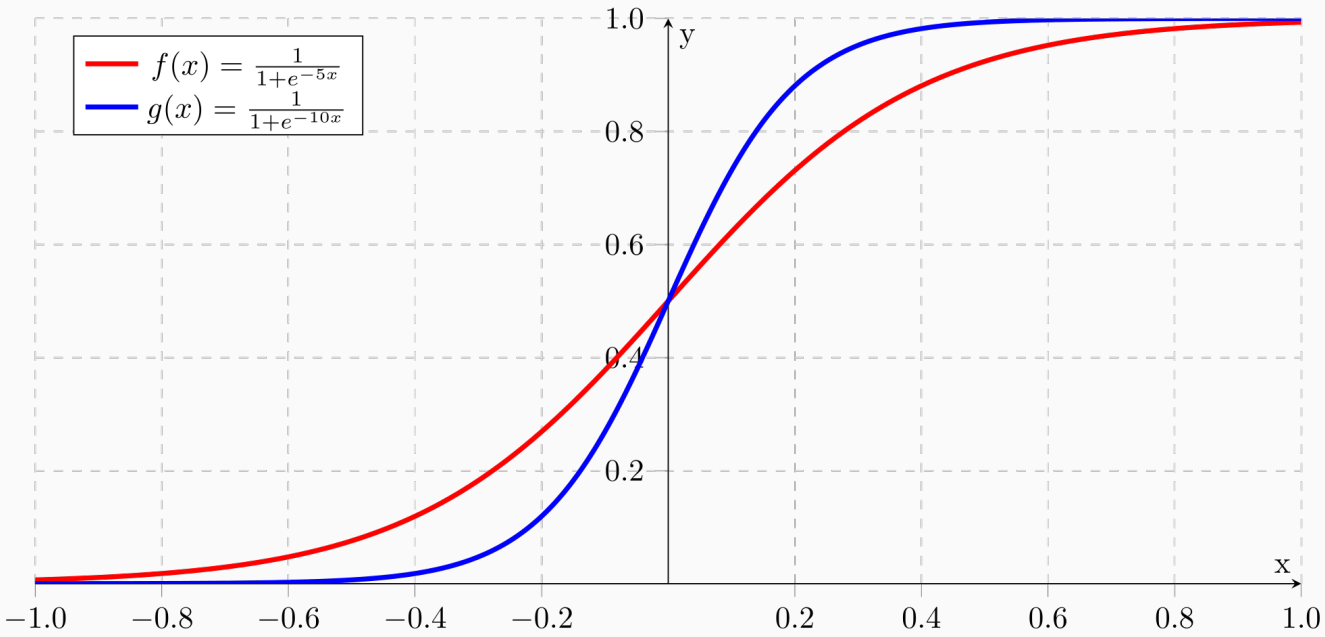
Risposta esatta

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

Attivazione di un nodo

L'output del j-esimo nodo è:

$$z_j = \sigma\left(\sum_{i=1}^{\#input} w_{ij} * x_i + b_j\right)$$



Come scegliere i parametri?

I parametri sono inizializzati random.
Si definisce una funzione che valuti l'**errore** commesso dalla rete neurale nel classificare le N immagini nel dataset:

$$MSE(\underline{w}, \underline{b}) = \frac{1}{N} \sum_{i=1}^{N} \|\underline{y_i} - \underline{\widehat{y}}(x_i, \underline{w}, \underline{b})\|^2$$

Il problema diventa trovare $\underline{w}^*, \underline{b}^*$ tali che:

$$(\underline{w}^*, \underline{b}^*) = \underset{(\underline{w}, \underline{b}) \in \mathbb{R}^{mxn}}{\arg\min}\ MSE(\underline{w}, \underline{b})$$

$\Rightarrow$ Algoritmi di **Ottimizzazione**

Ottimizzazione

Potenziali problemi:

- Minimi locali
- **Selle**

L'MSE è definito su uno spazio parametrico multidimensionale, nel caso di reti profonde si hanno centinaia di migliaia di parametri.

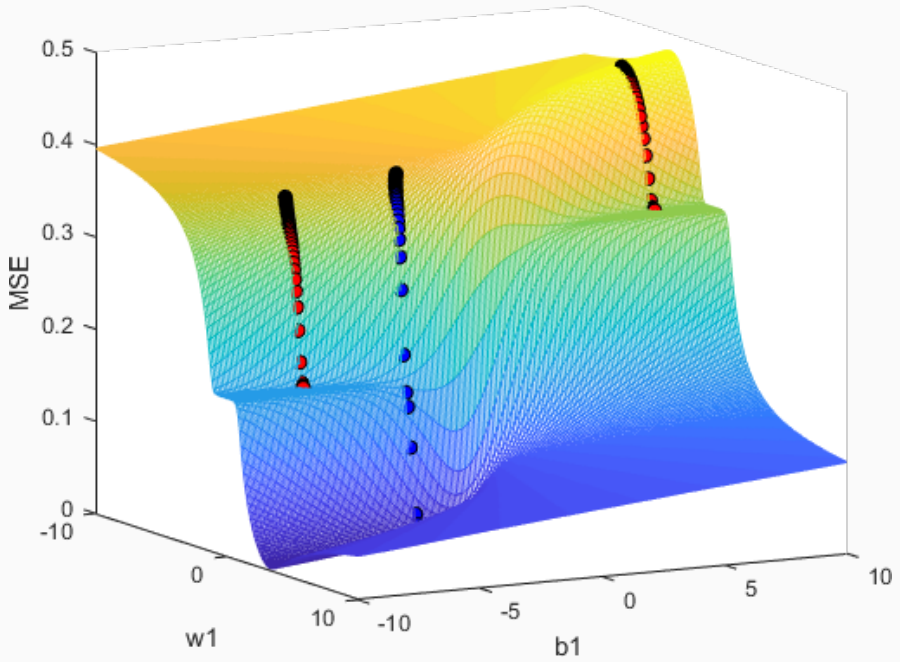
Per visualizzare le dinamiche di ottimizzazione si considera prima un caso molto semplice.

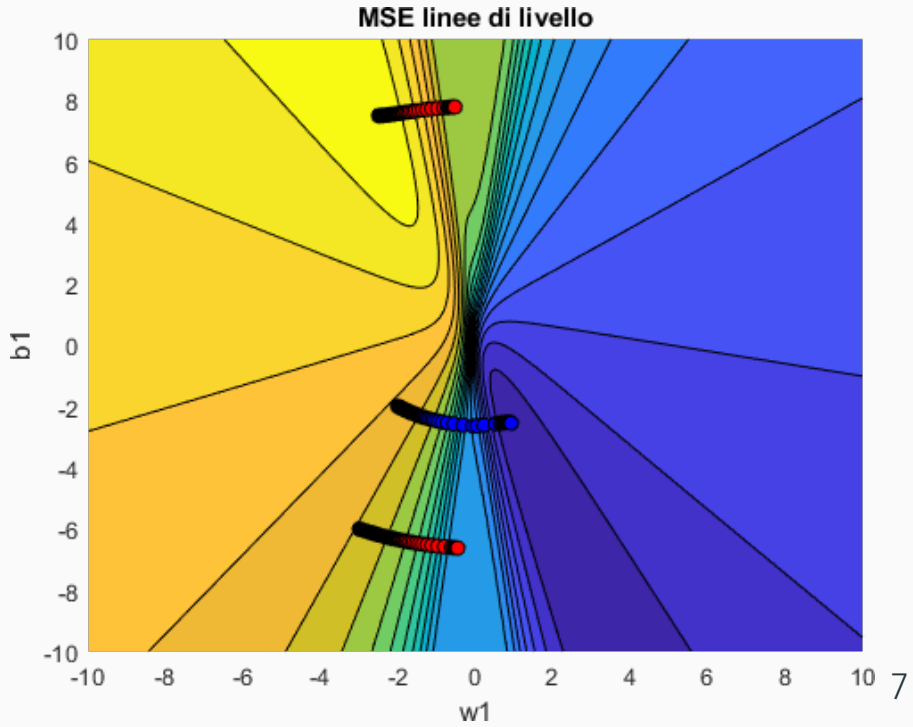
1 neurone: 2 parametri

$$\{(x_i, y_i)\}, x_i \in \mathbb{R}, y_i = \begin{cases} 1 & \text{se } x_i \geq 3 \\ 0 & \text{se } x_i < 3 \end{cases}$$

Un unico neurone, la rete è quindi la funzione

$$y = \widehat{y}(x, w_1, b_1) = \sigma(w_1 * x + b_1)$$





15 neuroni: 1675 parametri

Si implementa una rete neurale con **1675 parametri** per la classificazione di immagini.

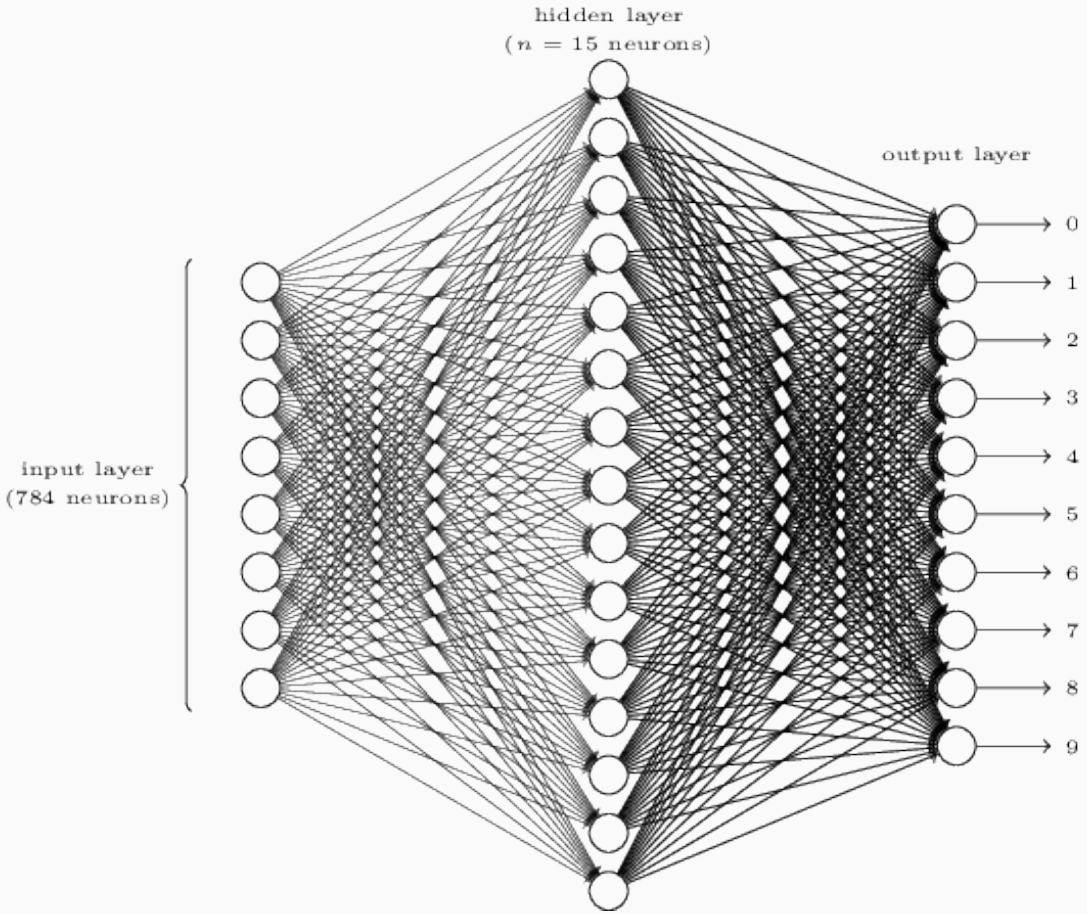
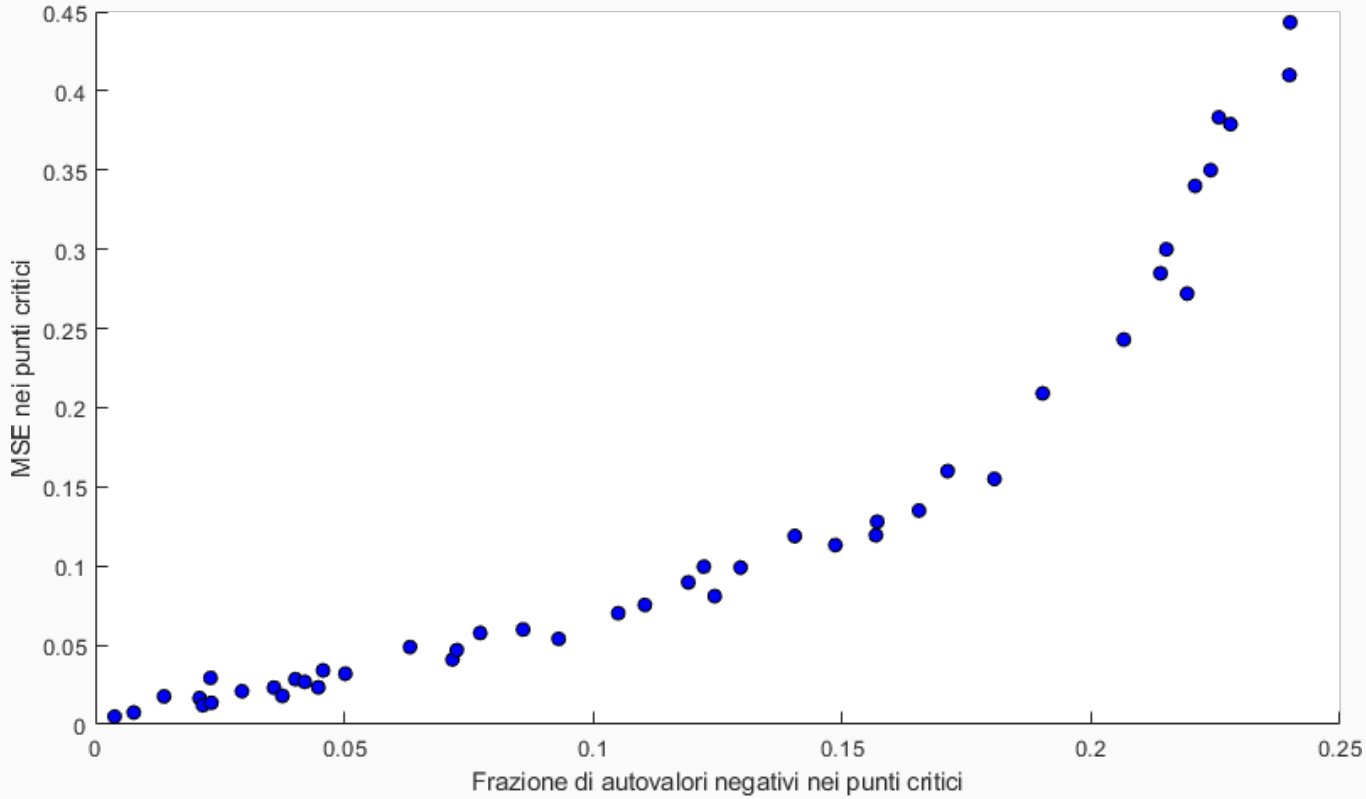


Image source: http://neuralnetworksanddeeplearning.com/chap1.html

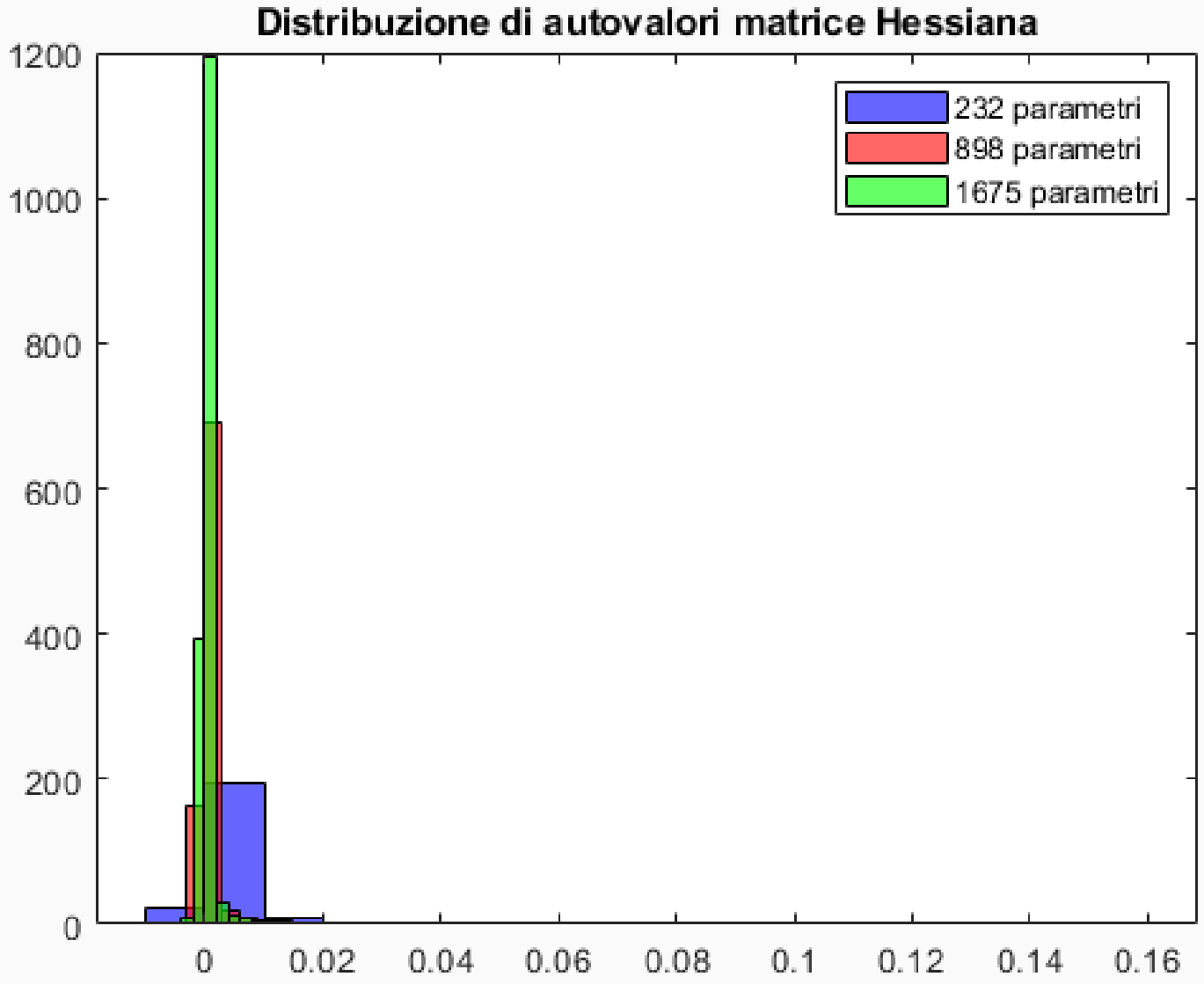
Mse nei punti estremanti

Si allena la rete e si valuta l'**Hessiana** nel punto critico raggiunto.



Si osserva una forte correlazione tra la frazione di **autovalori negativi** e l'**errore** nei punti estremanti.

Aumentando le dimensioni della rete lo spettro di autovalori si concentra attorno a zero.



Come fuggire dai punti di sella?

Metodi del secondo ordine (e.g. Newton)
Si sfrutta l'informazione sulla curvatura contenuta nella matrice hessiana.

- Hessiana malcondizionata
- Elevato costo computazionale

#Neuroni	#Parametri	Tempo
5	565	$\approx$ 1 min
10	1120	$\approx$ 5 min
15	1675	$\approx$ 13 min
20	2230	$\approx$ 25 min

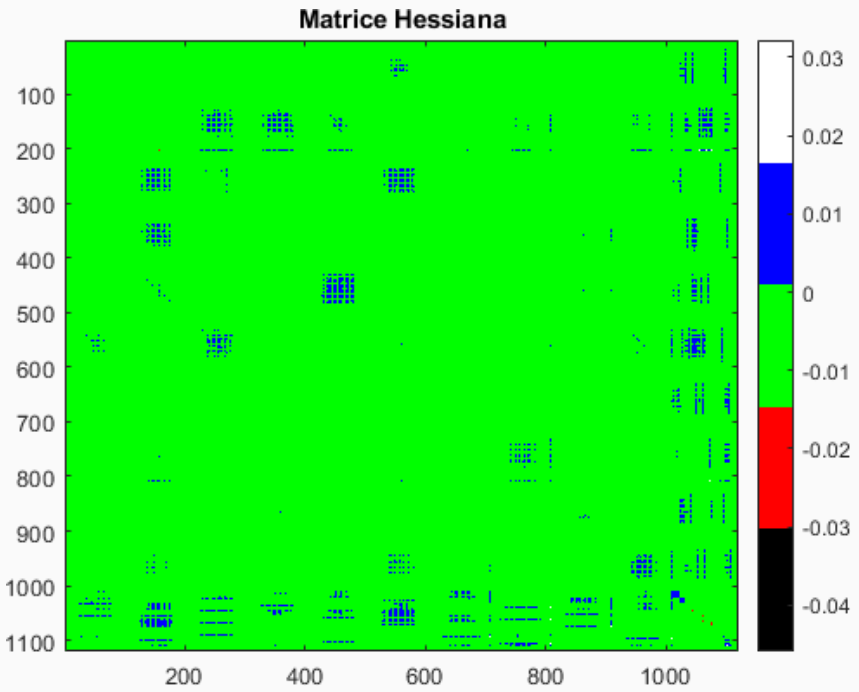
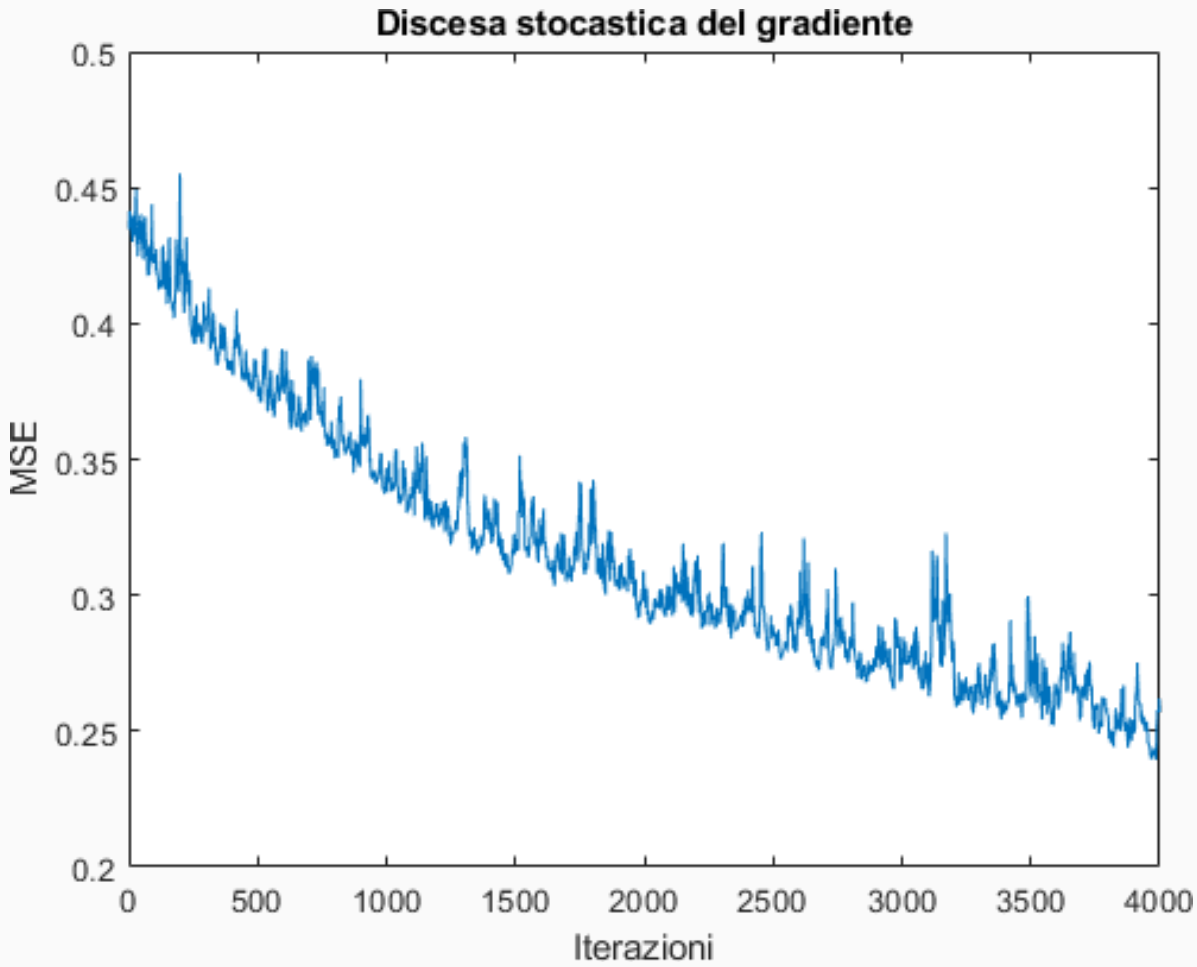


Figure: Hessiana, 1120 parametri

Discesa stocastica del gradiente

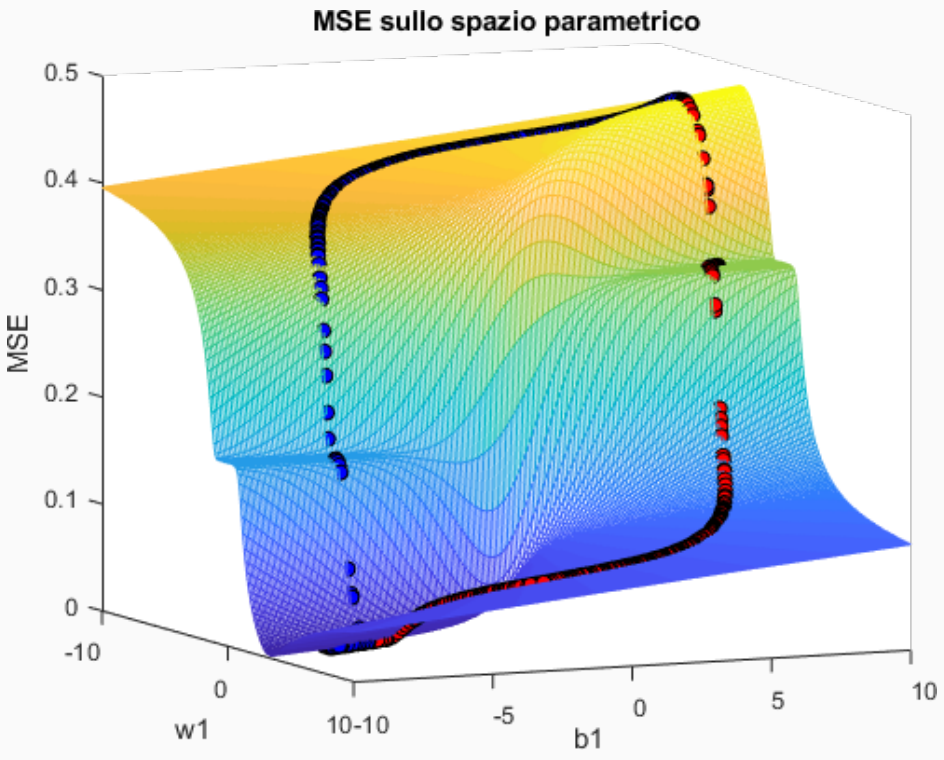
Non si ottimizza l'MSE su tutto il dataset ma su una porzione ristretta di esso.

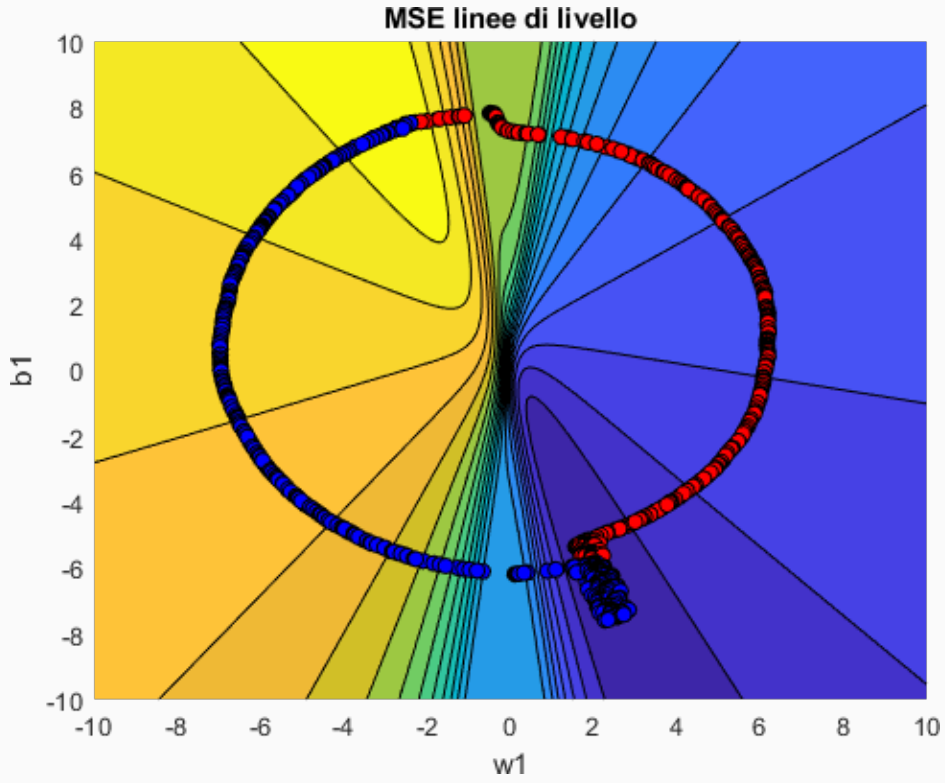
- Semplicità computazionale (primo ordine)
- Può essere implementato in parallelo
- Evita punti critici ad alto errore



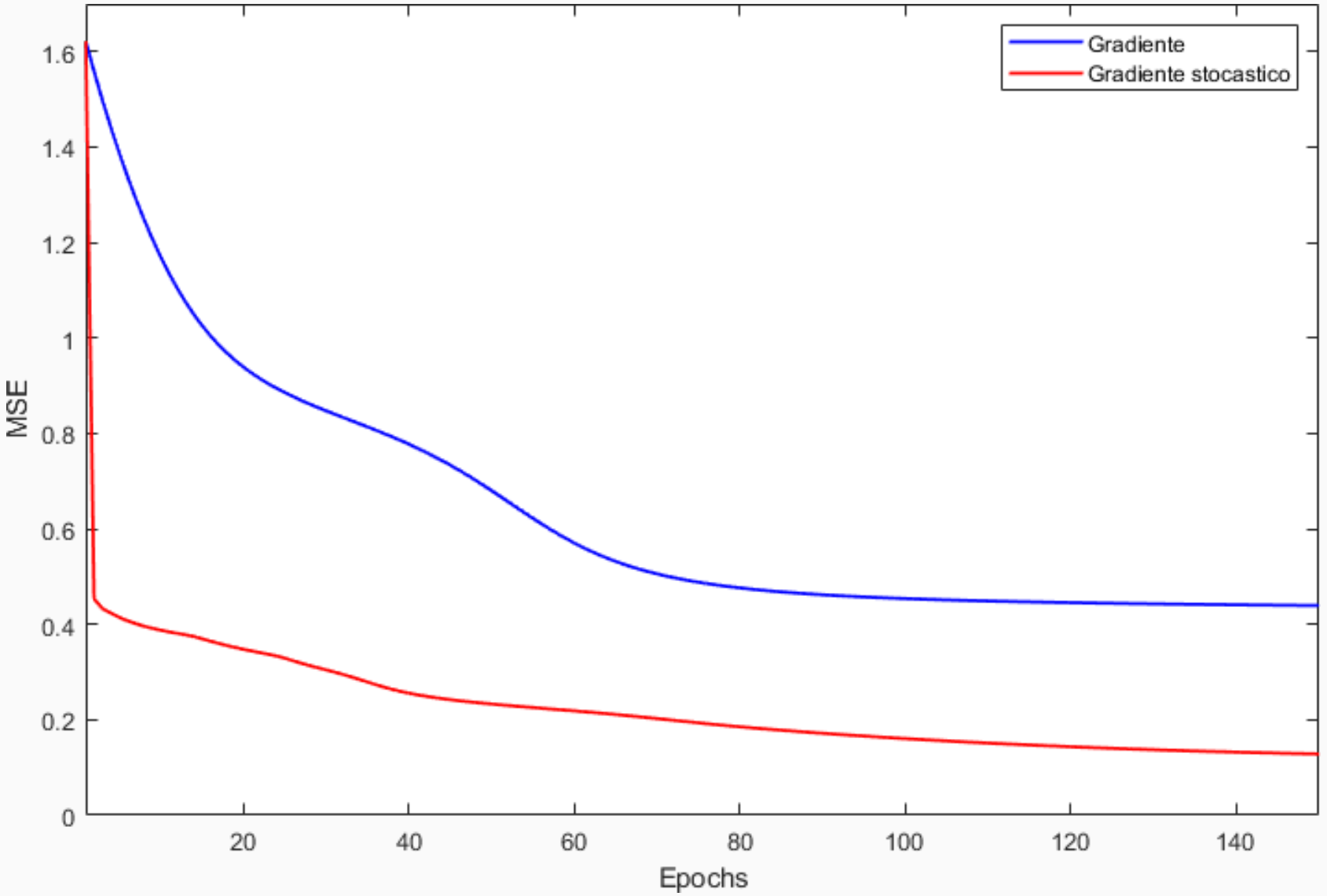
1 neurone: 2 parametri

Si riscontra la capacità del metodo stocastico di superare, o aggirare, i punti di sella in cui il metodo del gradiente si era bloccato.

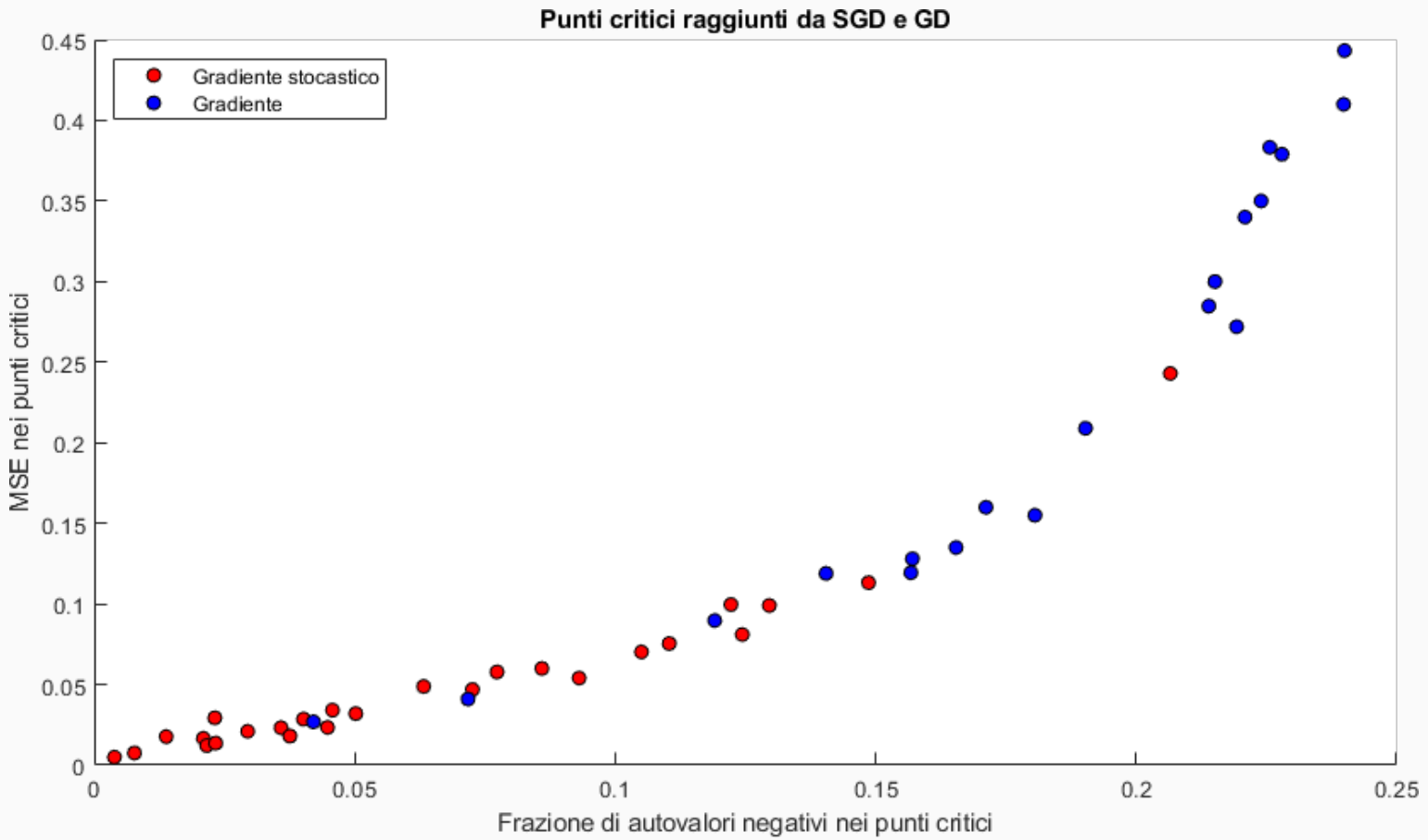




15 neuroni: 1675 parametri



Anche nel caso multidimensionale il metodo di discesa stocastica performa decisamente meglio del metodo del gradiente, raggiungendo punti ad errore minore aventi una percentuale inferiore di autovalori negativi.



Referenze

- Yann N. Dauphin et al. - Identifying and attacking the saddle point problem in high-dimensional non-convex optimization
- Yann LeCun et al. - Eigenvalues of the hessian in deep learning: singularity and beyond
- Catherine F. Higham, Desmond J. Higham - Deep learning: an introduction for applied mathematicians
- Christopher M. Bishop - Exact calculation of the hessian matrix for the multi-layer perceptron