Intelligenza Artificiale per Sistemi Industriali

Lezione 05 — 10 Marzo 2026

Alice Plebe

www.aliceplebe.com alice.plebe@unitn.it

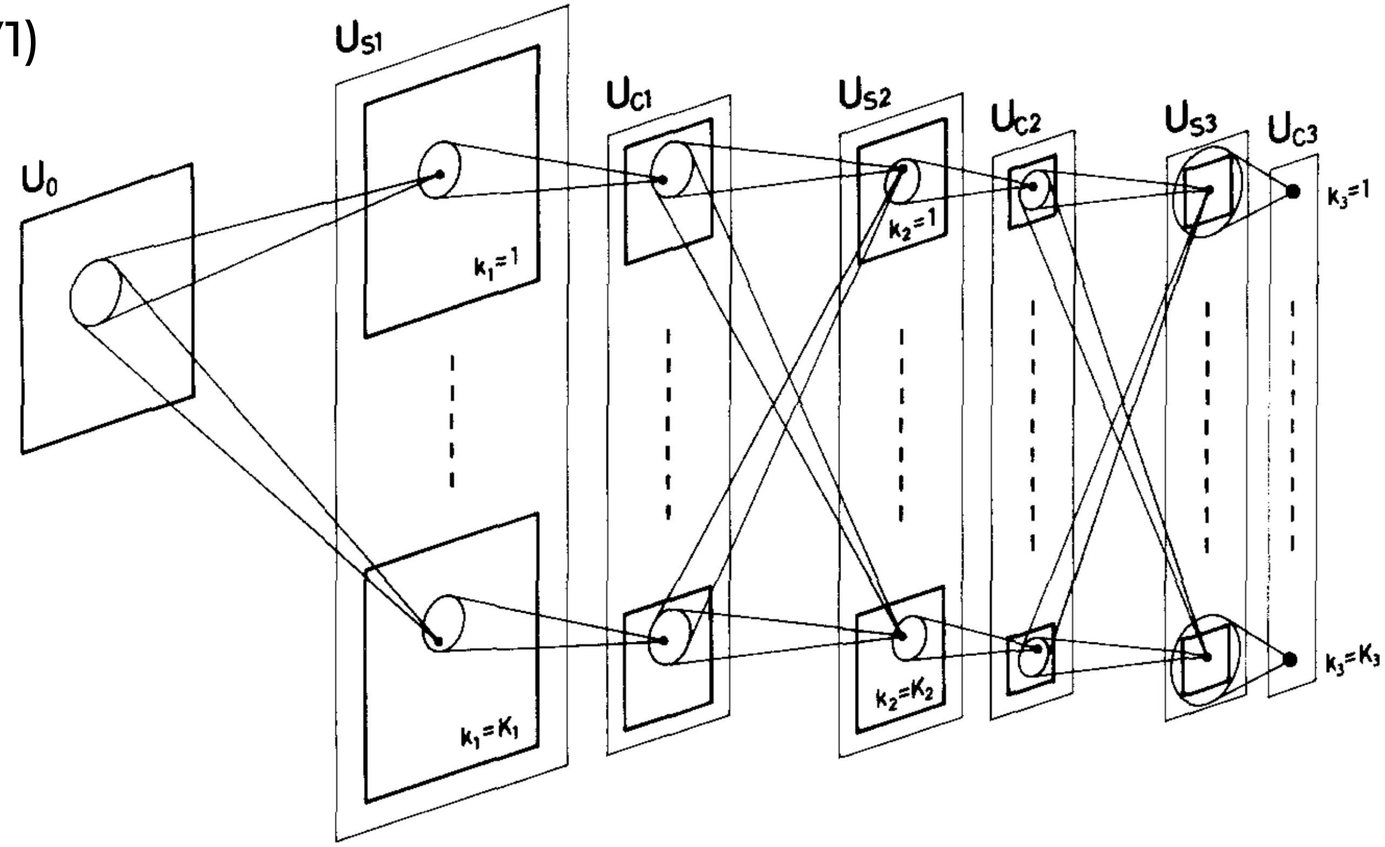
Oltre al Perceptron

APPROCCI ALTERNATIVI

- *Neocognitron*, Kunihiko Fukushima, 1980
- *Boltzmann Machine*, Geoffrey Hinton & Terry Sejnowski, 1985
- *Self-organizing maps*, Teuvo Kohonen, 1982

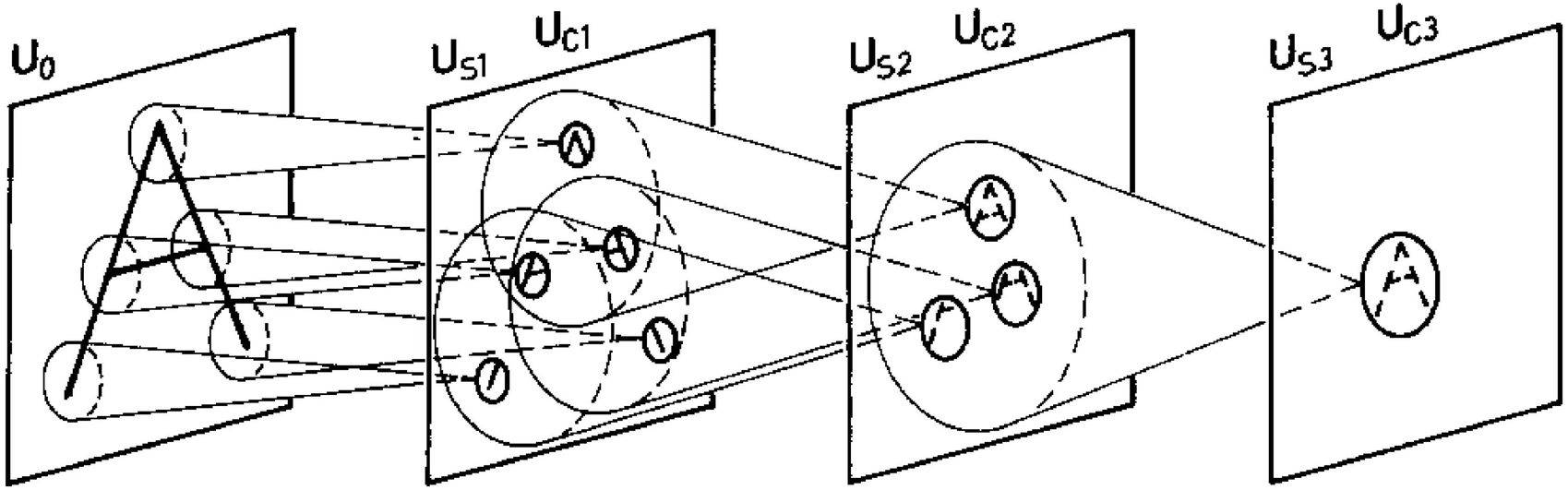
NEOCOGNITRON

- Per riconoscimento di pattern visivi con una certa invarianza spaziale
- Ispirazione dalla corteccia visiva (V1)
- Antenato delle CNNs
- **S-cells**: caratteristiche locali
- **C-cells**: aggregazione di S-cells



NEOCOGNITRON

- **Gerarchizzazione** (S-cells) + **pooling** (C-cells)
- Invarianza locale
- Niente backpropagation! Apprendimento ispirato alla regola di Hebb



BOLTZMANN MACHINE

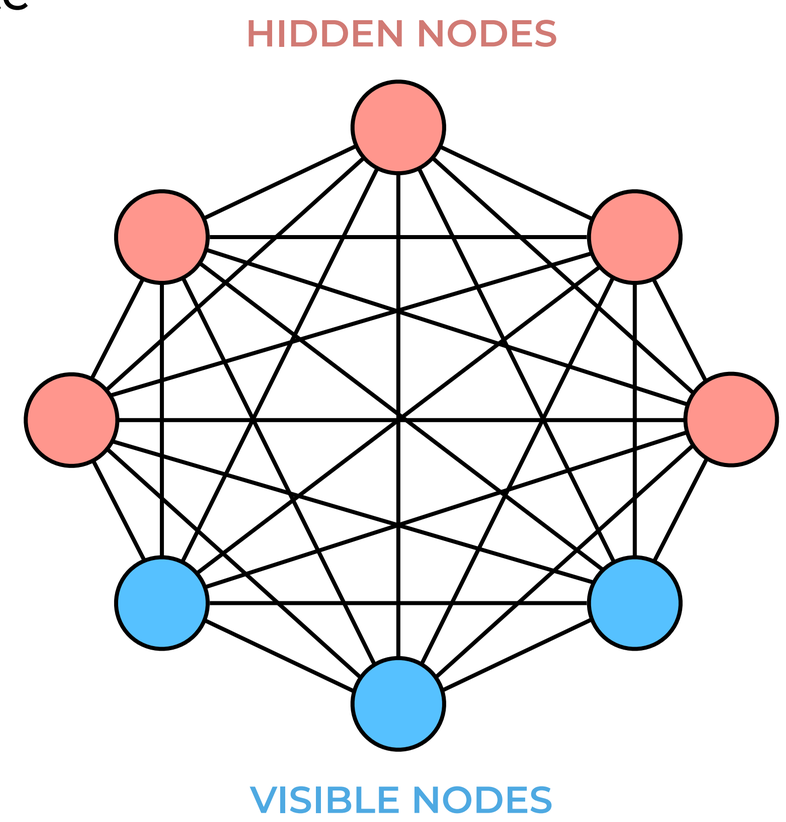
- Per modellare distribuzione di dati, no classificazione
- Rete completamente connessa, non direzionata, con pesi simmetrici
- Funzione di energia, dove $\mathbf{s} = (s_1, \ldots, s_n)$ è lo stato della rete

$$E(\mathbf{s}) = -\sum_{i<j} w_{ij}\, s_i\, s_j \; - \; \sum_i b_i\, s_i$$

- La probabilità sulle configurazioni è determinata dalla distribuzione di Boltzmann

$$P(\mathbf{s}) = \frac{1}{Z}\, e^{\left(-E(\mathbf{s})\right)}$$

dove $Z = \sum_{\mathbf{s}} e^{\left(-E(\mathbf{s})\right)}$



BOLTZMANN MACHINE

FUNZIONAMENTO

$$P\left(s_i = 1 \mid s_{\_i}\right) = \frac{1}{1 + \exp\left(-\sum_j w_{ij}\, s_j - b_i\right)}$$

- Pesi già fissati
- Generare configurazioni tramite Gibbs sampling
  che tendono all'equilibrio termico

ADDESTRAMENTO

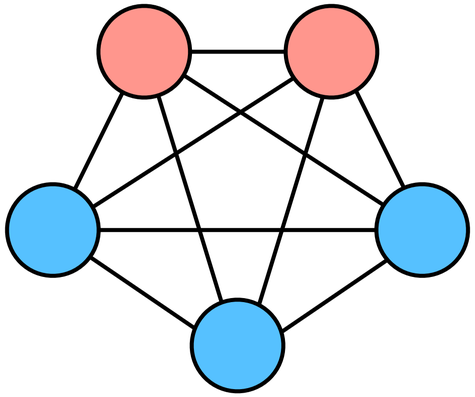
- Modificare i pesi per far coincidere distribuzione di equilibrio con distribuzione dei dati
- Differenza tra correlazione tra i e j
  con rete "vincolata" e rete "libera"

$$\Delta w_{ij} = \eta\left(\langle s_i\, s_j\rangle_V - \langle s_i\, s_j\rangle_L\right)$$

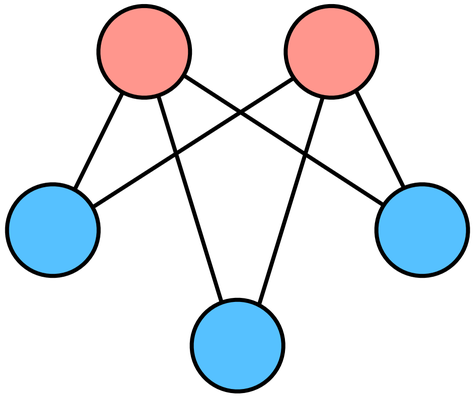
BOLTZMANN MACHINE

- Addestramento di BM "complete" instabile e computazionalmente troppo costoso
- *Restricted* Boltzmann machine come alternativa trattabile
- Separazione tra layer visibile e layer nascosto

GENERAL
BOLTZMANN MACHINE



RESTRICTED
BOLTZMANN MACHINE



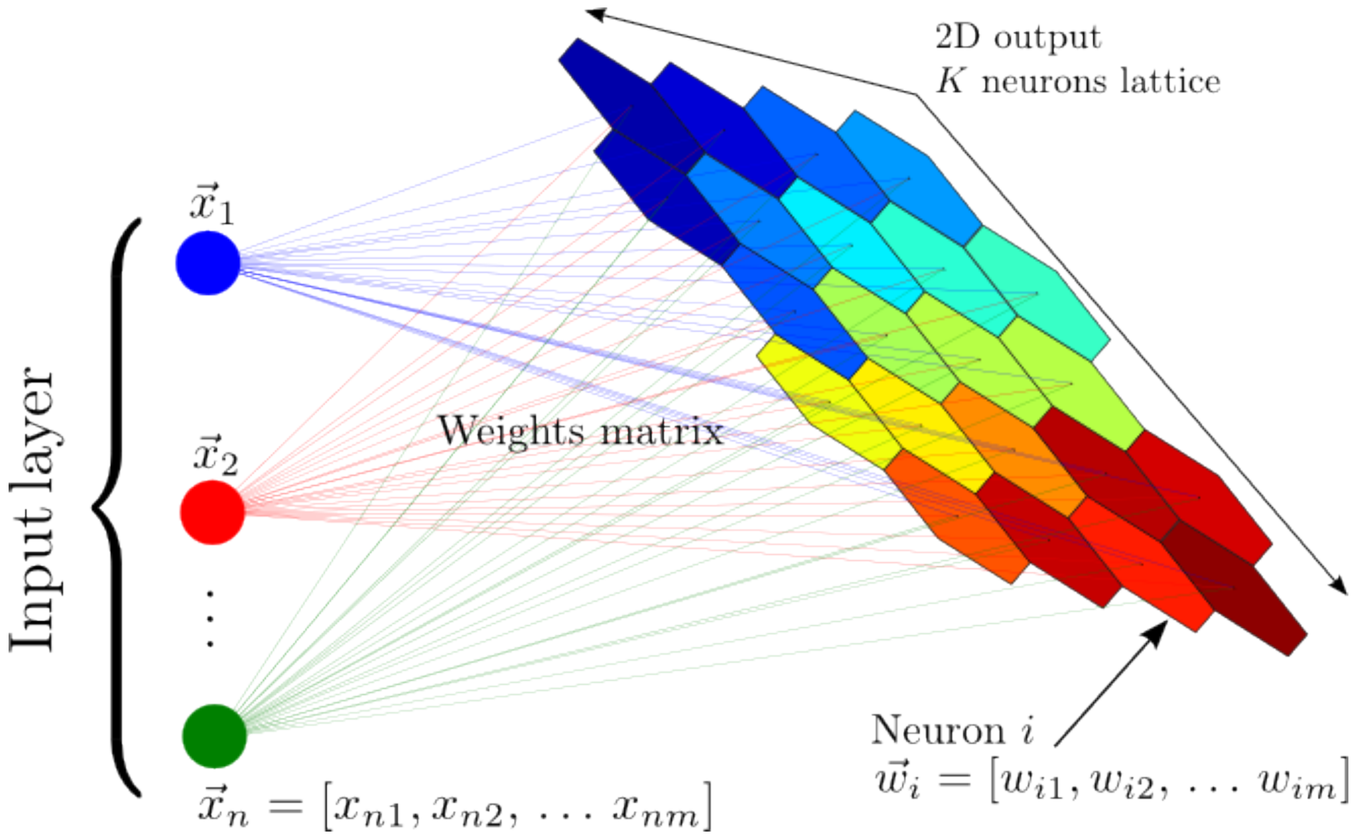
SELF-ORGANIZING MAPS

- Per riduzione dimensionale
- No layer, singola griglia di neuroni
- Apprendimento *winner-takes-all*

$$c = \arg\min_i \| \mathbf{x} - \mathbf{w}_i \|$$

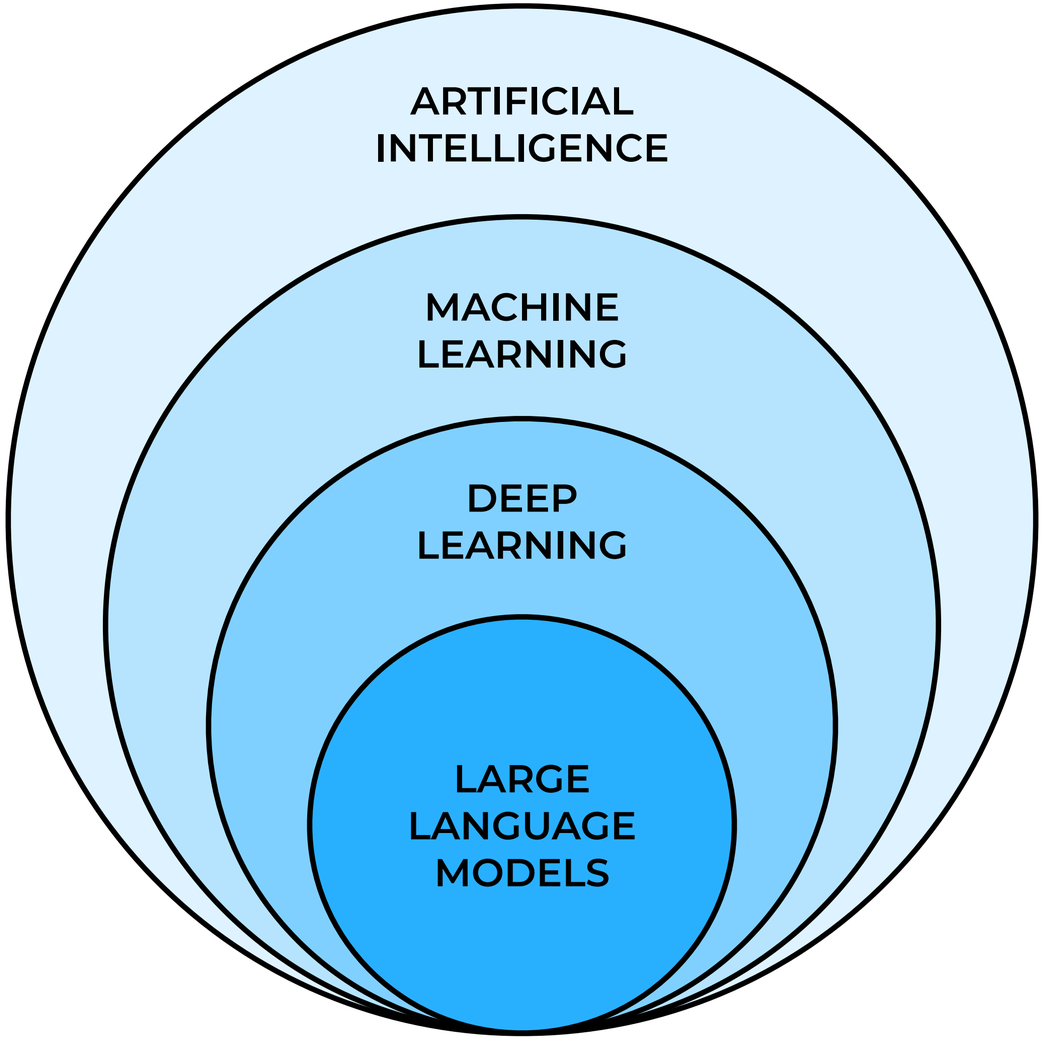
- Ma si aggiornano i pesi di tutti i neuroni vicino all vicincitore

$$\Delta \mathbf{w}_i = \eta \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_i\|^2}{2\sigma^2}\right)(\mathbf{x} - \mathbf{w}_i)$$



Reti artificiali moderne

CLASSIFICAZIONE ODIERNA



RETE NEURALE

RETE COME FUNZIONE PARAMETRICA

$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x})$$

FUNZIONE DI LOSS

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i \right)$$

COMPOSIZIONE DI HIDDEN LAYERS

$$\mathbf{h}_1 = \varphi\left(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1\right)$$

$$\mathbf{h}_2 = \varphi\left(\mathbf{W}_2 \mathbf{h}_1 + \mathbf{b}_2\right)$$

$$\dots$$

$$\mathbf{h}_M = \varphi\left(\mathbf{W}_M \mathbf{h}_{M-1} + \mathbf{b}_M\right)$$

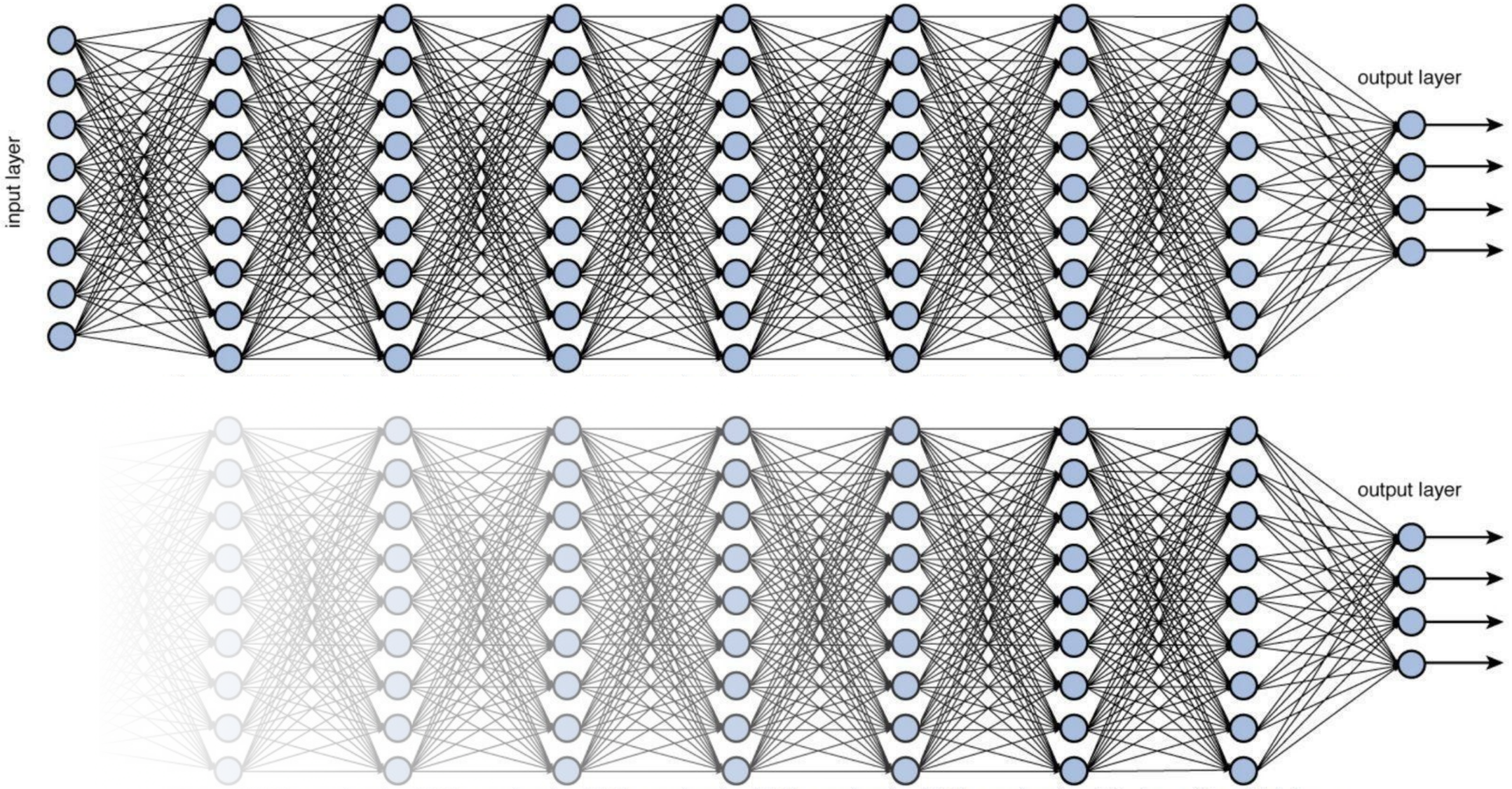
OBIETTIVO DEL TRAINING

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

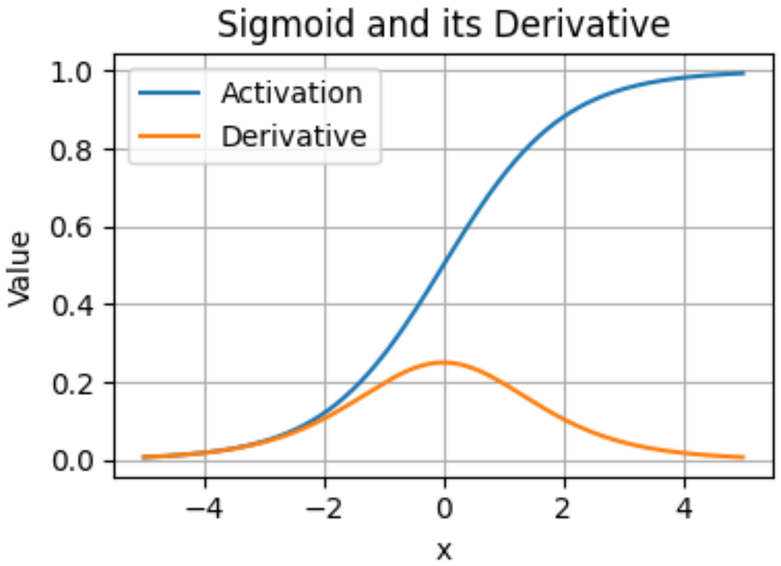
DISCESA DEL GRADIENTE

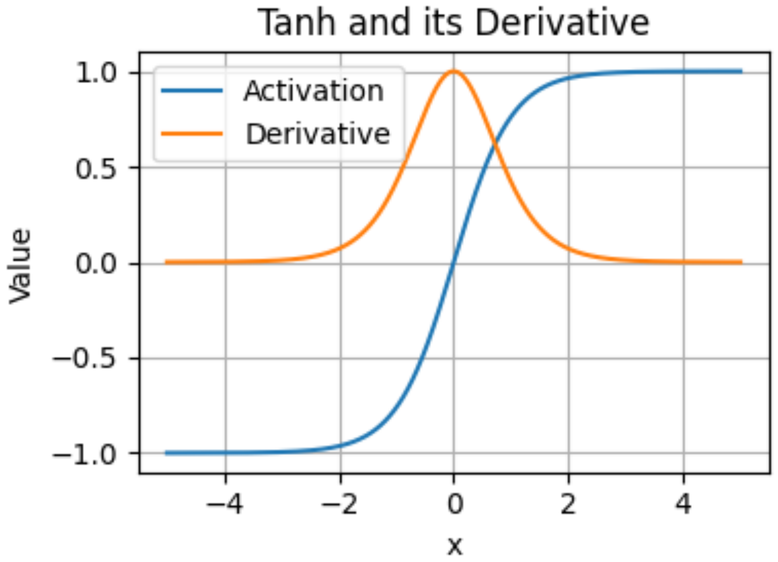
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

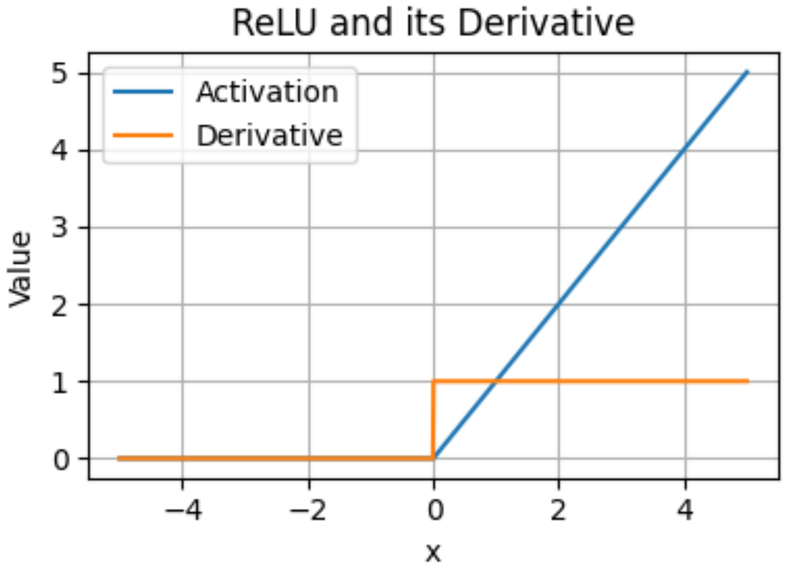
VANISHING GRADIENT PROBLEM



ACTIVATION FUNCIONS







MINI-BATCH / STOCHASTIC GRADIENT DESCENT

FUNZIONE DI LOSS

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i\right)$$

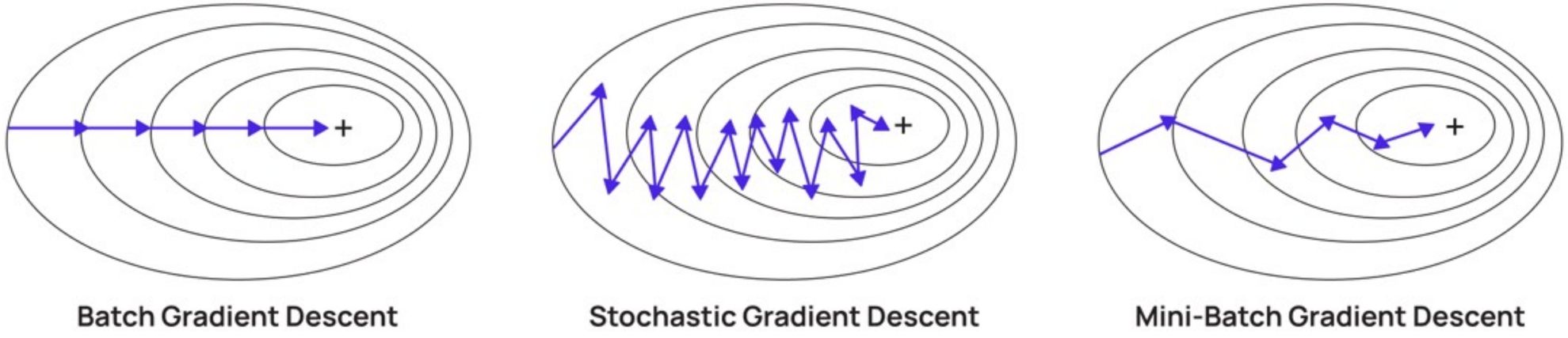
AGGIORNAMENTO SGD

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta})$$

USARE $\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta})$ **COME STIMA DI** $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$

MINI-BATCH GRADIENT DESCENT

$$\nabla_{\boldsymbol{\theta}} L_B(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i \in B} \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta})$$



Batch Gradient Descent | Stochastic Gradient Descent | Mini-Batch Gradient Descent

DEEP NEURAL NETWORKS

- **Backpropagation** rende possibile calcolare il gradiente in reti a più strati
- **ReLU** rende il gradiente stabile lungo molti strati
- **Mini-batch SGD** rende l'ottimizzazione scalabile su grandi dataset