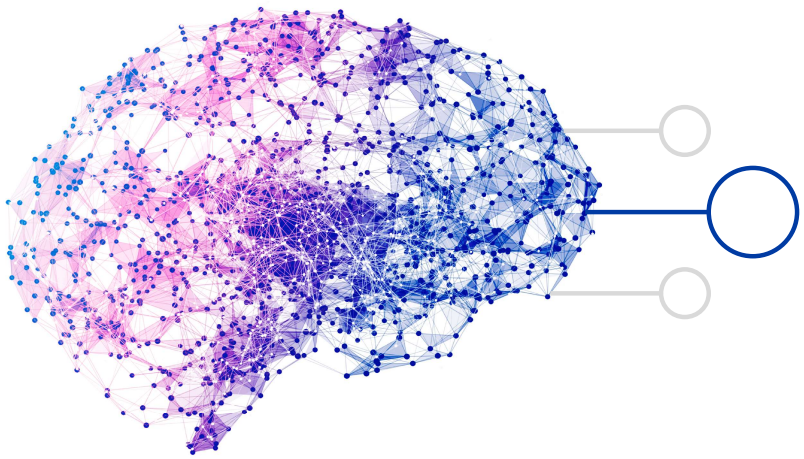
Background DEEPTECH

Dalle reti neurali all'AI generativa

Alessandro Bria (a.bria@unicas.it)
Professore Associato, Università di Cassino

Sommario



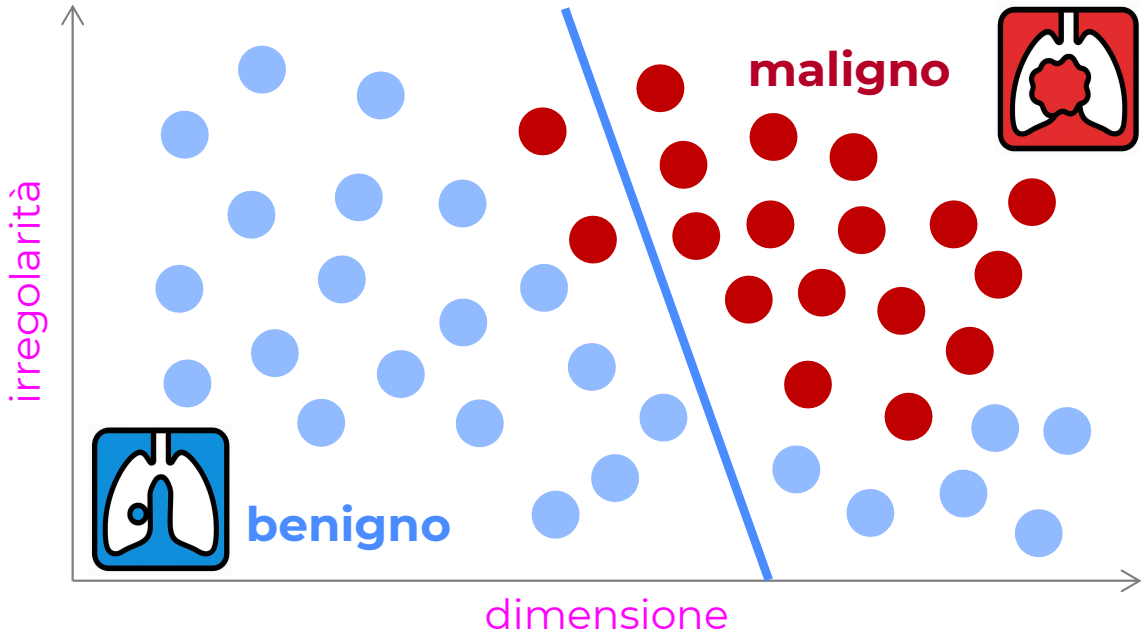
01

Come imparano le macchine?

Reti neurali artificiali

Machine Learning

- *modelli* matematici (il più semplice: una retta!)
- dipendono da parametri appresi in modo *statistico* da **dati storici**
- utilizzano misure caratteristiche estratte dai dati



modello **lineare** $(2D)$
n. parametri = 3
errore = 15%

$y = mx + q$

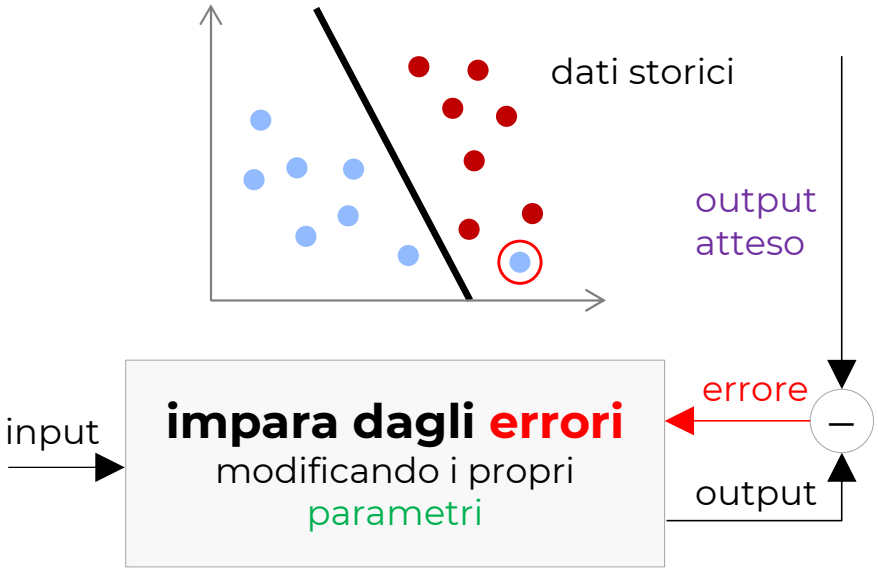
(scuola superiore)

$w_1 x_1 + w_2 x_2 + b = 0$

(università)

Machine Learning

- *modelli* matematici **addestrabili**
- dipendono da parametri appresi in modo *statistico* da **dati storici**

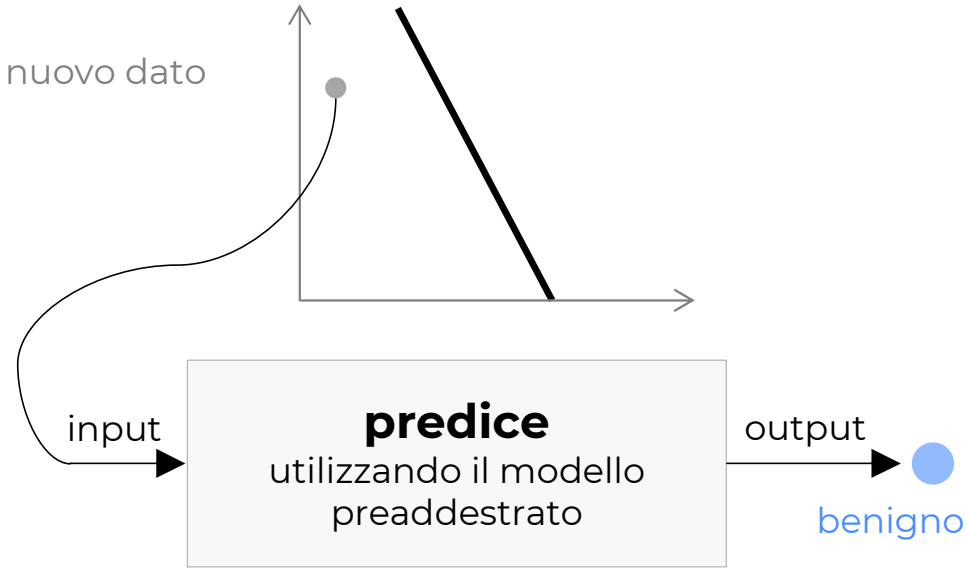


addestramento (*ore ~ mesi*)

Machine Learning

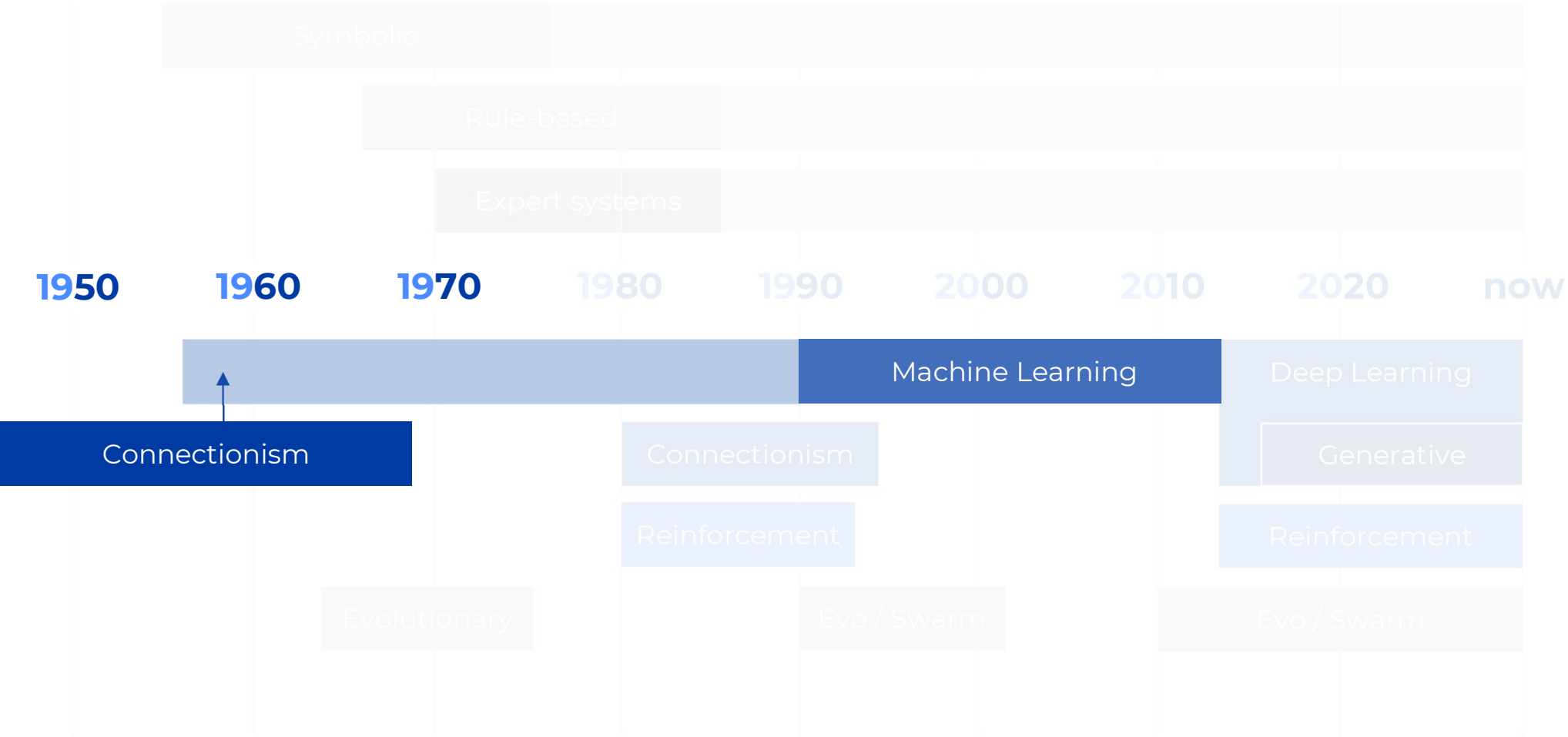
- *modelli* matematici **addestrabili**
- dipendono da parametri appresi in modo *statistico* da **dati storici**

addestramento (*ore ~ mesi*)



inferenza (*real-time*)

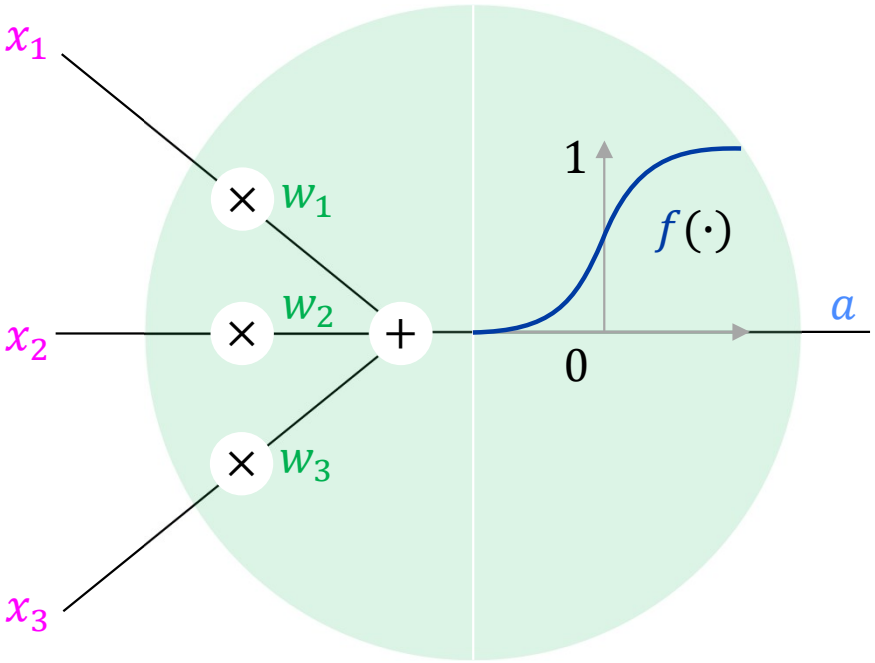
Connessionismo (1943-1969)



Connessionismo

- imita il funzionamento del cervello
- basato su neuroni artificiali interconnessi
  - reti neurali artificiali
- il paradigma AI più antico
  - confluito nel moderno Deep Learning

Neurone artificiale (1943-1958)



(1) ogni **input** x_i **viene pesato con** w_i

- i pesi (*weights*) sono i **parametri** che vengono modificati in addestramento

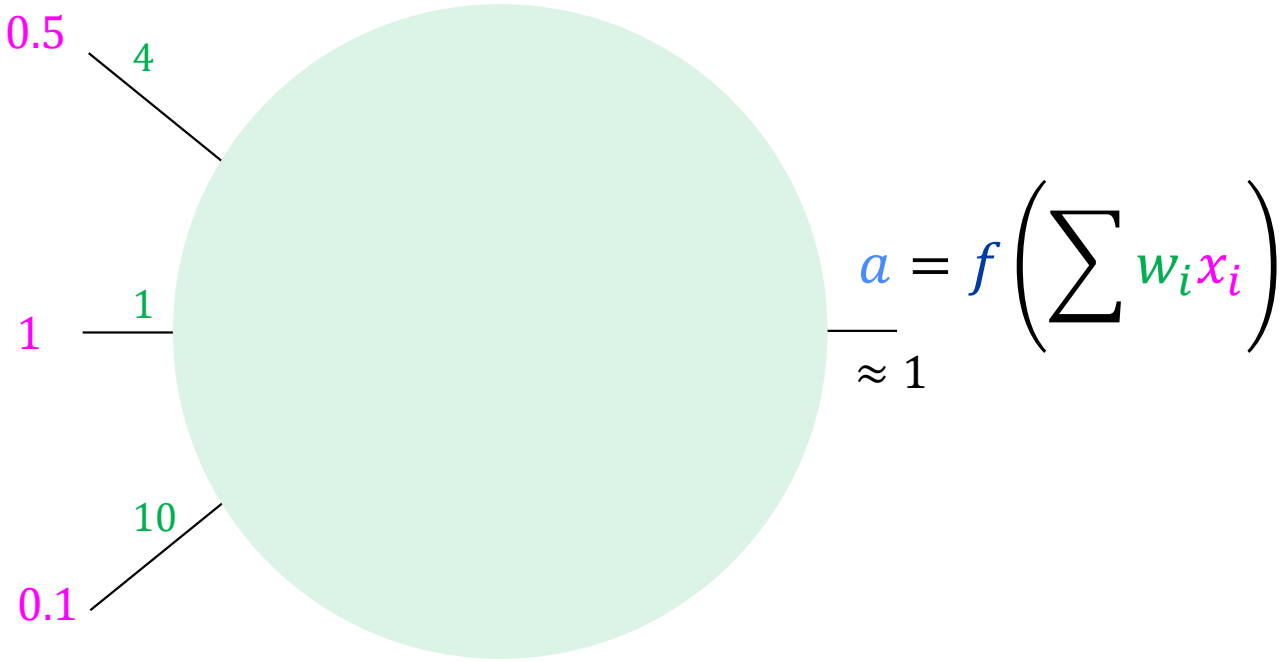
(2) gli input pesati vengono **sommati**

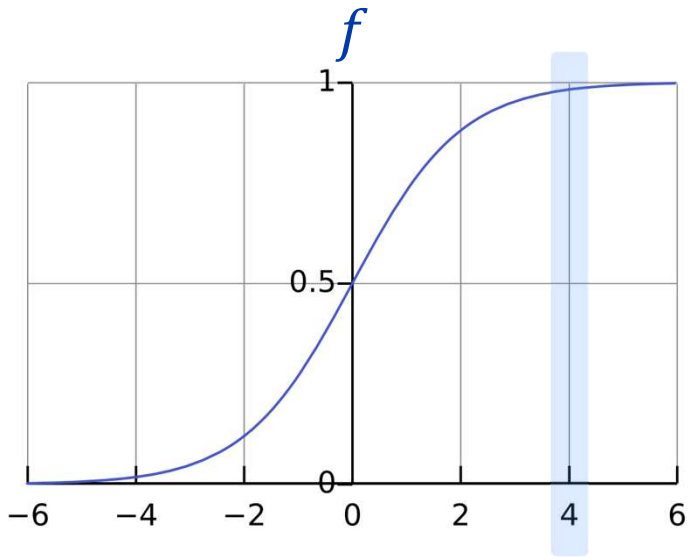
- così da ottenere un unico valore

(3) **attivazione**

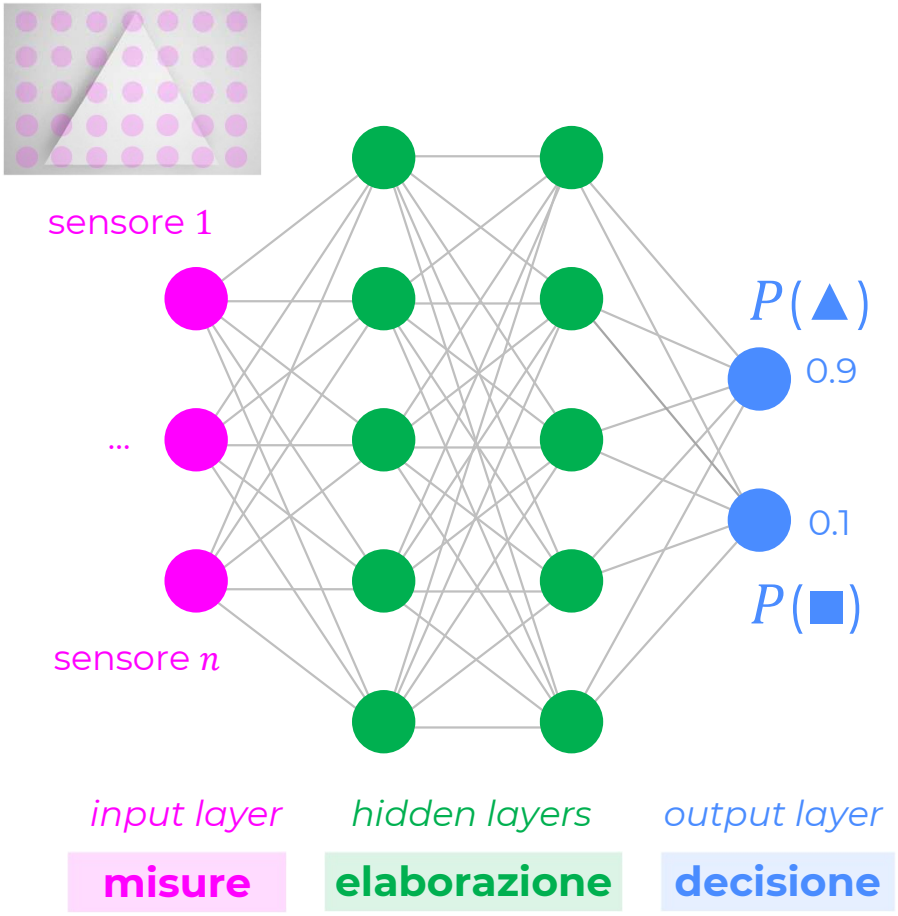
- il valore viene mappato da una funzione di attivazione f in un range tra
  - 0 (neurone spento)
  - 1 (neurone attivo)

Neurone artificiale





Rete neurale artificiale (1958)



- stratificazioni (*layer*) di **neuroni** artificiali **interconnessi**
  - *input layer* riceve le misure
  - uno o più *hidden layer* per l'elaborazione
  - *output layer* per la decisione
    - ad es. stima della probabilità P che l'input appartenga ad una certa categoria (classificazione)

Multi-Layer Perceptron (MLP) (1958)

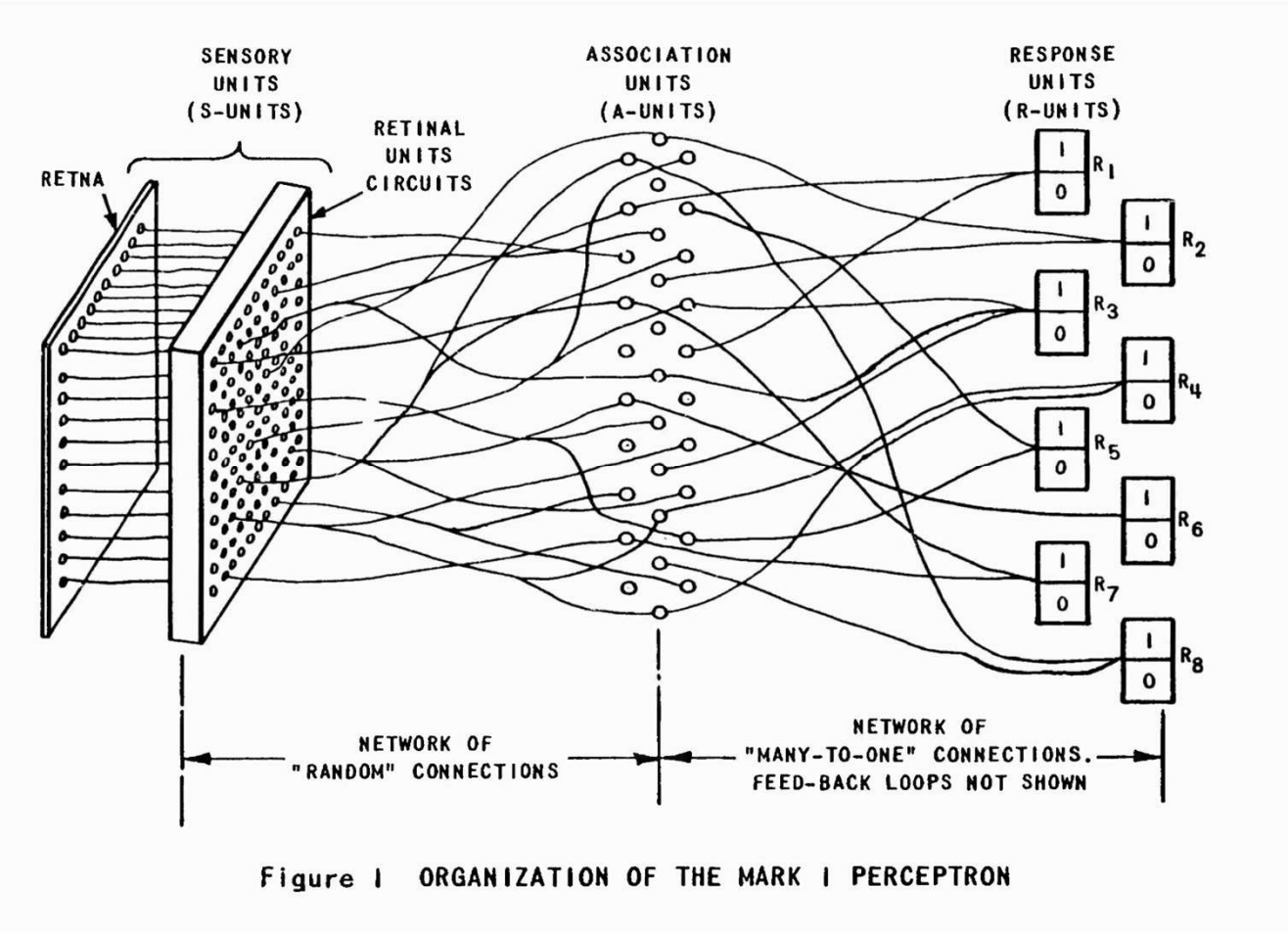
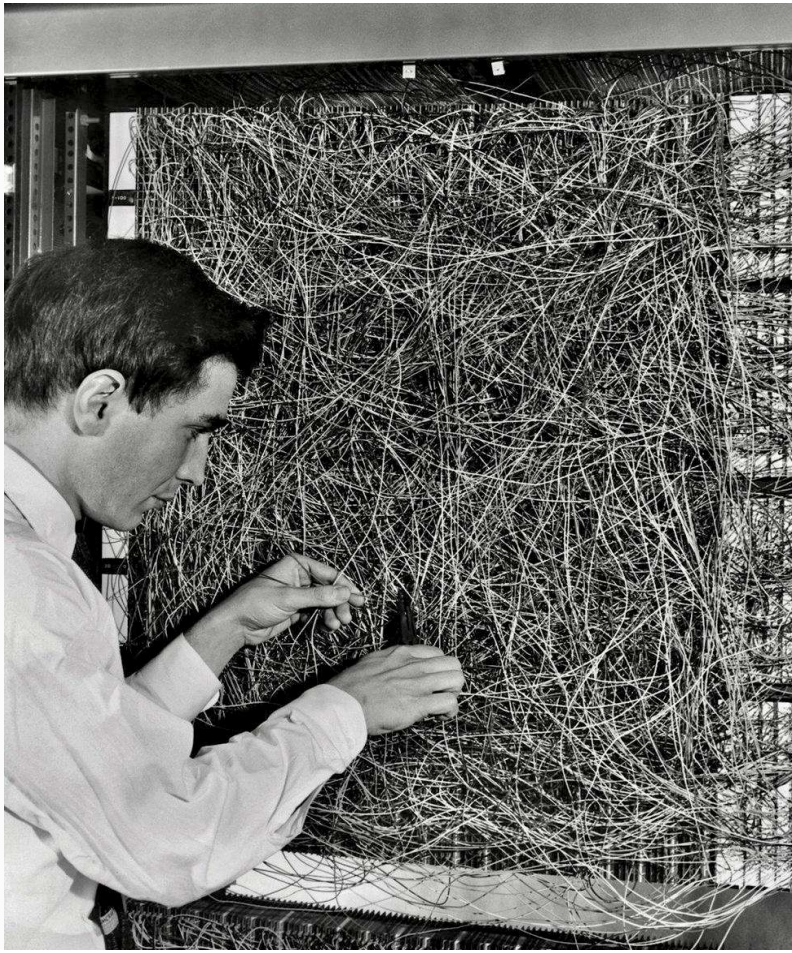
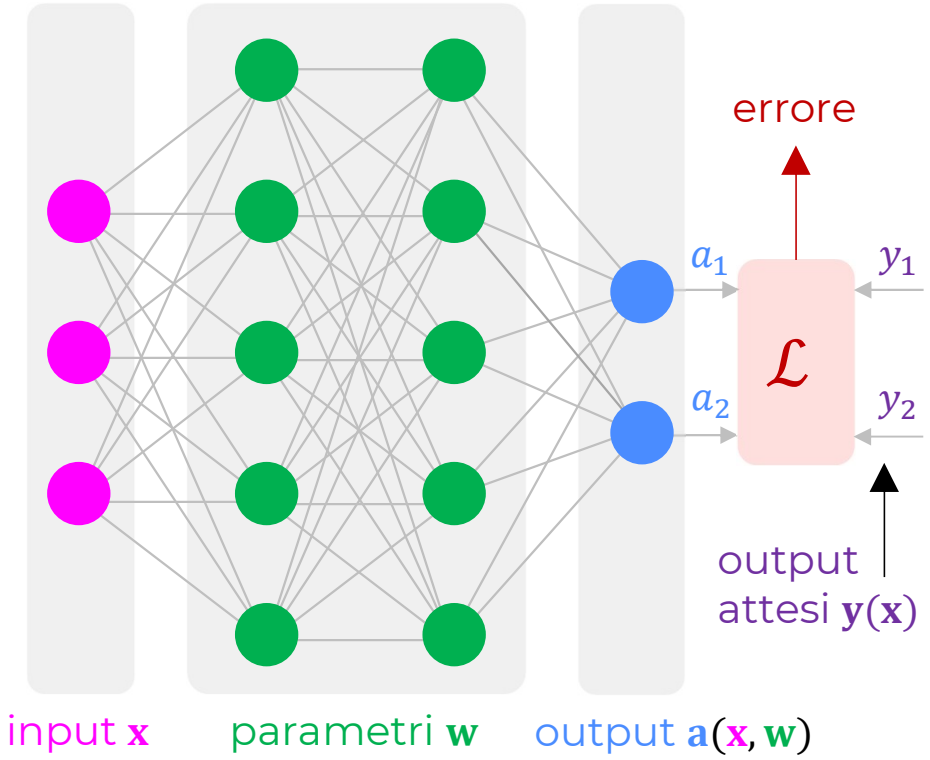


Figure 1 ORGANIZATION OF THE MARK I PERCEPTRON



Funzione di errore (1960)

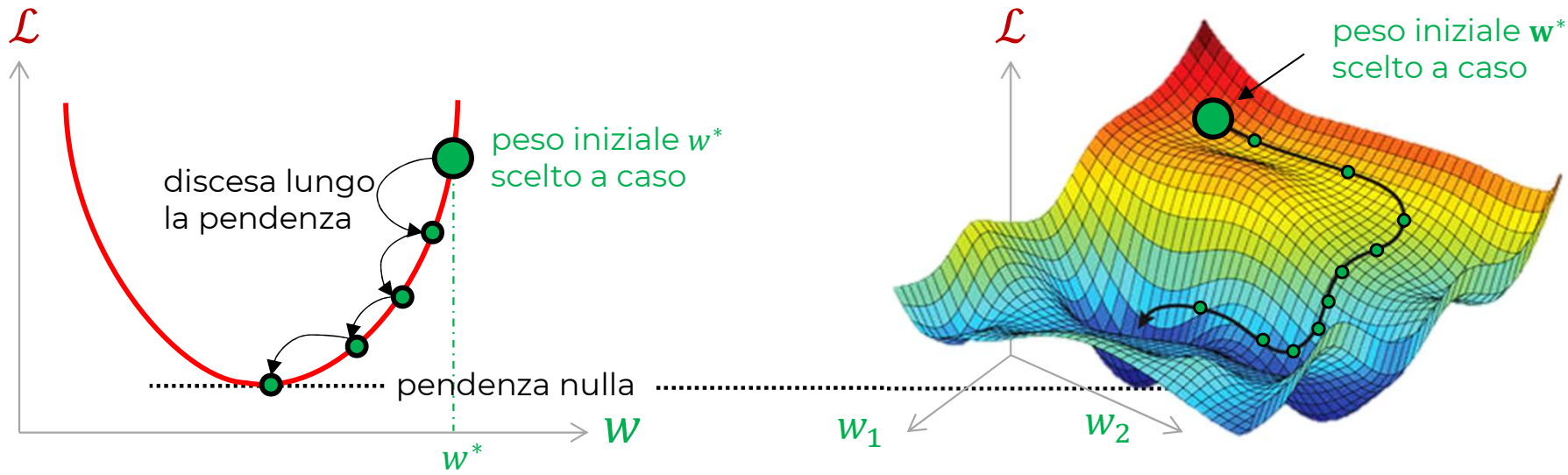


- forniamo dati di input $\mathbf{x}$ alla rete e calcoliamo gli output $\mathbf{a}(\mathbf{x}, \mathbf{w})$
- per **imparare**, ci serve quantificare *quanto simile* $\mathbf{a}(\mathbf{x}, \mathbf{w})$ è rispetto all'output atteso $\mathbf{y}(\mathbf{x})$
  - usiamo una funzione di errore (*Loss function*)
- la più semplice funzione di errore è il *Mean Squared Error* (MSE):

$$\mathcal{L}_{MSE}(\mathbf{w}) = \frac{1}{N} \sum_{\mathbf{x}} (\mathbf{y} - \mathbf{a})^2$$

Addestramento (1967)

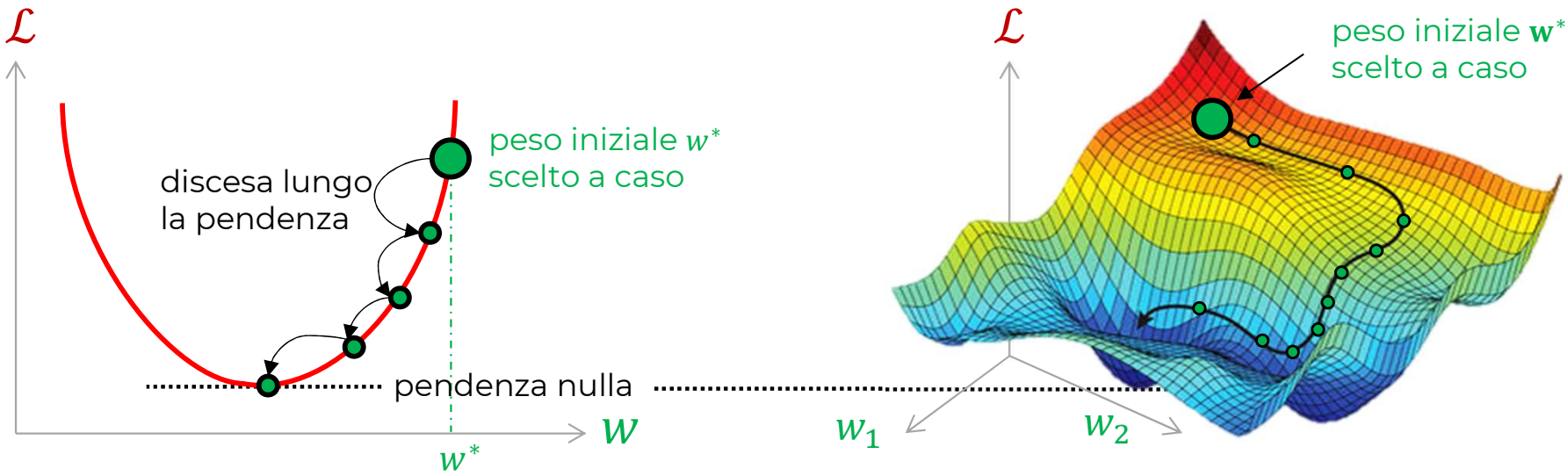
- metodo del gradiente discendente (*Cauchy*, 1847)
  1. inizializzazione *casuale* dei pesi $\mathbf{w}$
  2. aggiornamenti successivi di $\mathbf{w}$ verso la **direzione di massima discesa** dell'errore $\mathcal{L}$
  3. termino se errore $\mathcal{L}$ minimo ⇒ pesi $\mathbf{w}$ ottimali ⇒ rete addestrata ■



Addestramento (1967)

- metodo del gradiente discendente (*Cauchy*, 1847)
  1. inizializzazione *casuale* dei pesi $\mathbf{w}$
  2. aggiornamenti successivi di $\mathbf{w}$ verso la **direzione di massima discesa** dell'errore $\mathcal{L}$
  3. termino se errore $\mathcal{L}$ minimo ⇒ pesi $\mathbf{w}$ ottimali ⇒ rete addestrata ■

– **problema**
  - serve calcolare il **gradiente** $\stackrel{\text{def}}{=}$ **derivata (pendenza)** di $\mathcal{L}$ rispetto a ciascun parametro $w_1, w_2, \ldots, w_n$



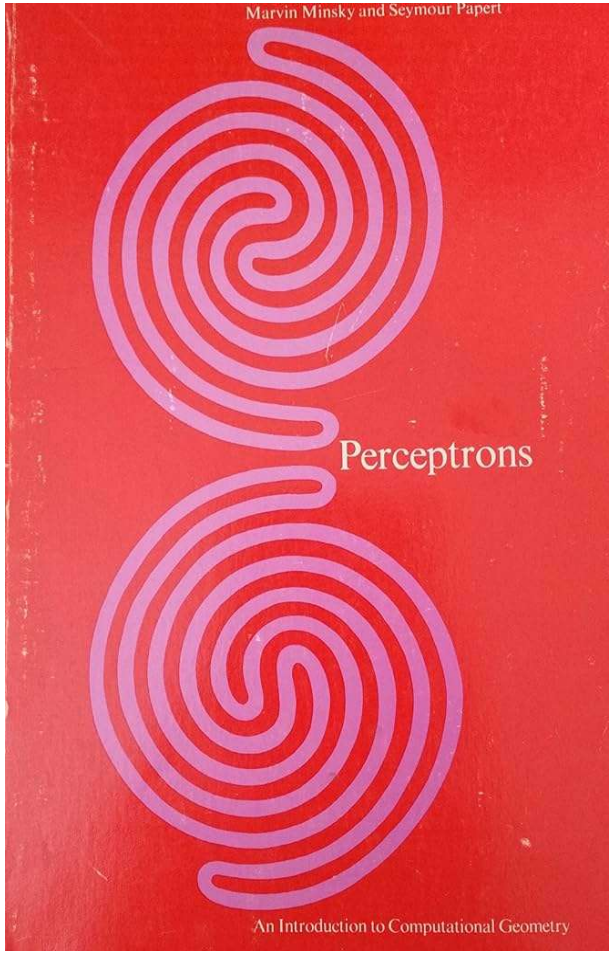
Crisi (1969-1980)

- libro “*Perceptrons*” (*Minsky*, 1969)
  - dimostrò che un singolo strato di percettroni non poteva risolvere la funzione XOR

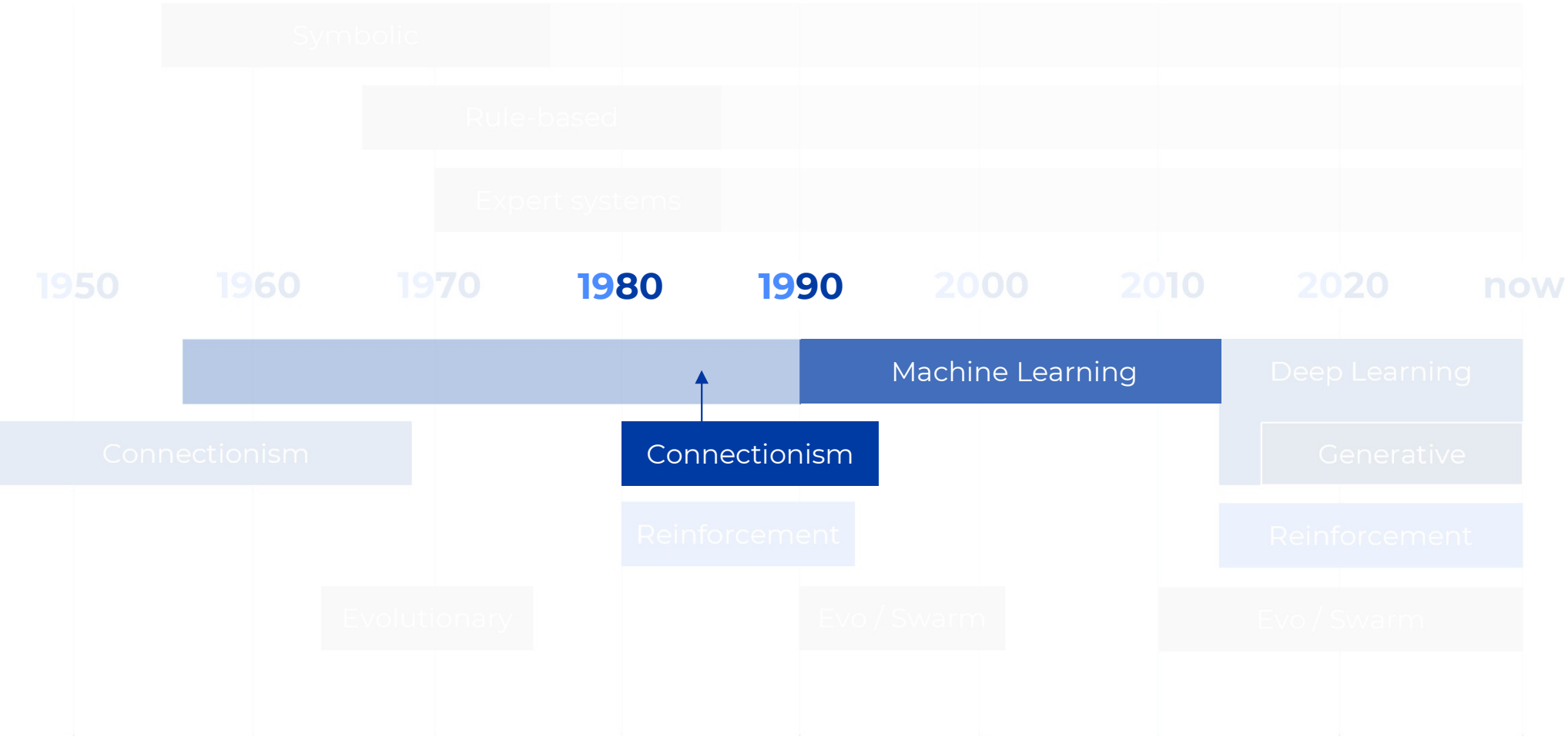
x_1	x_2	XOR
1	1	0
0	1	1
1	0	1
0	0	0

  - i ricercatori dell’epoca interpretarono il risultato come «*TUTTE le reti neurali sono fondamentalmente limitate*»

- mancanza di
  1. metodologie per calcolare le **derivate** automaticamente
  2. sufficienti quantità di **dati** per l’addestramento
  3. **hardware** ad elevata capacità computazionale

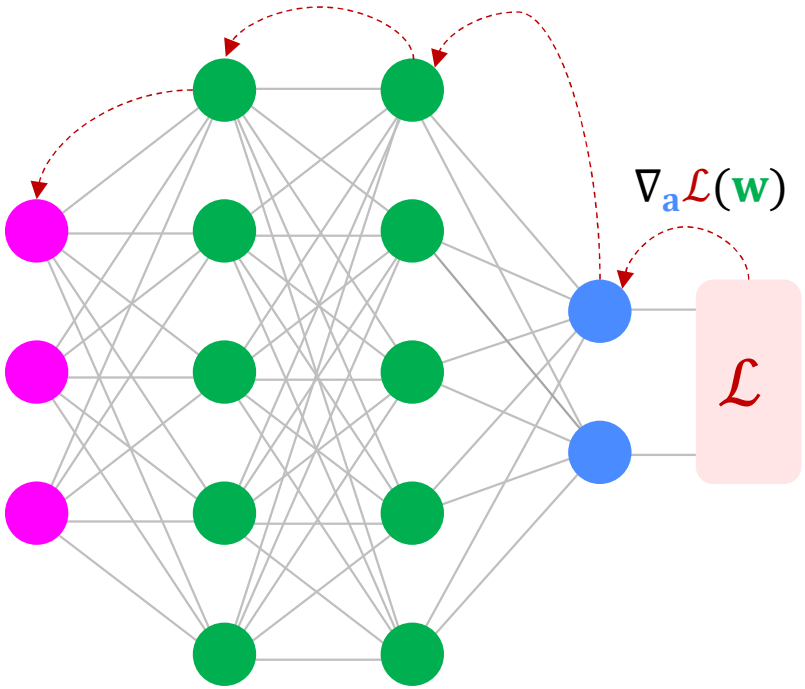


Connessionismo II (1980-1993)

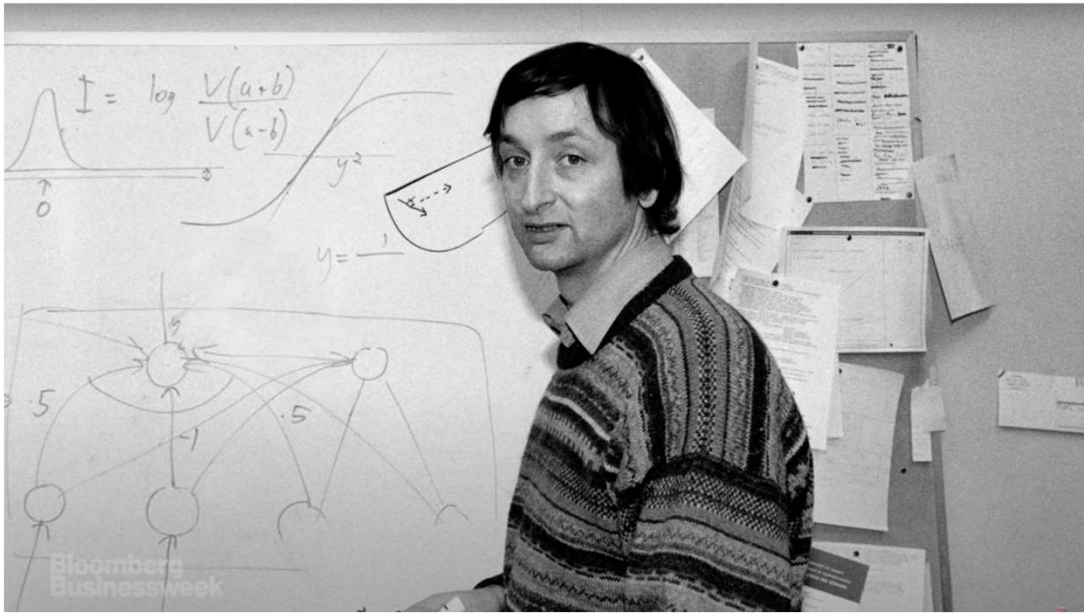


Backpropagation (1986)

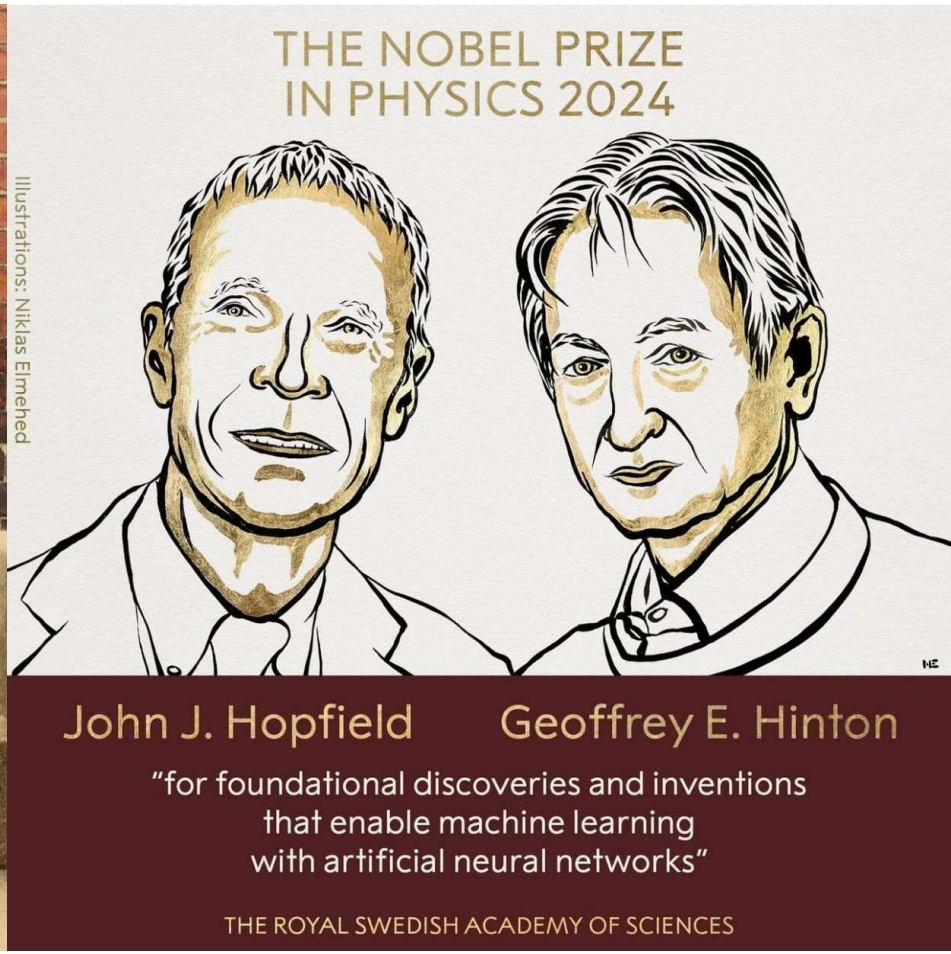
$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \left(\ldots, \frac{\partial \mathcal{L}}{\partial w_j^{\ell}}, \ldots \right)^T$$

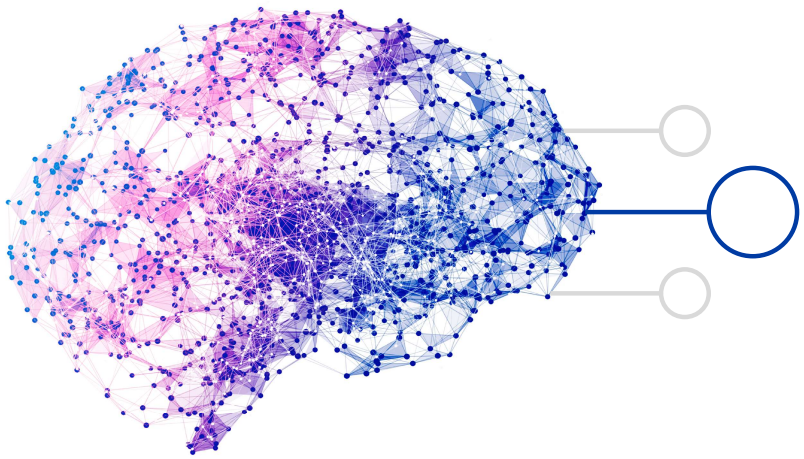


- il tassello mancante
  - un algoritmo per calcolare le **derivate** della funzione di errore rispetto a ciascun parametro della rete
  - addestramento = gradiente discendente + backpropagation



Nobel per la Fisica (2024)

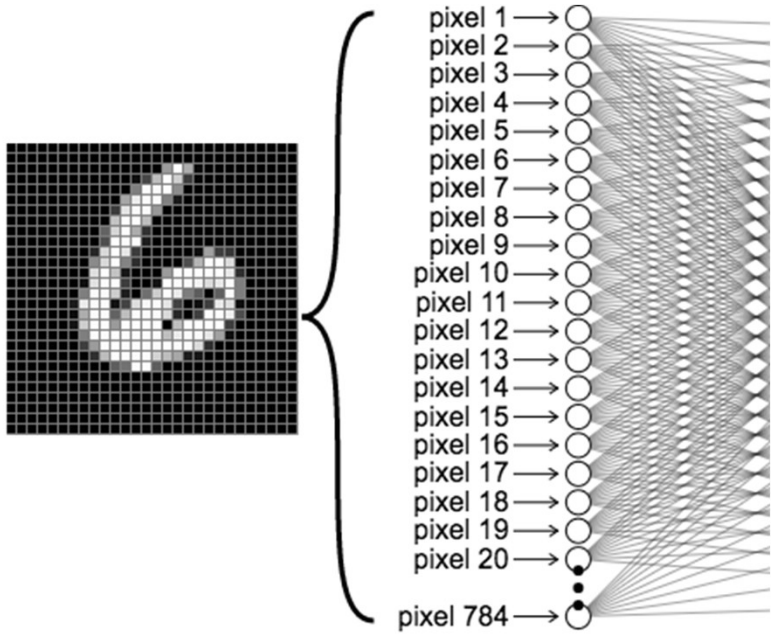




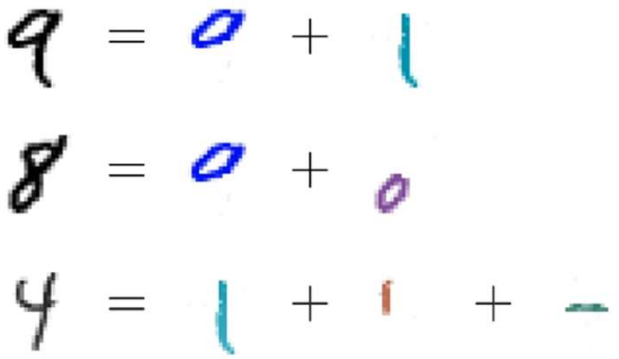
02
Come percepiscono?

Reti convoluzionali

Limiti delle reti MLP

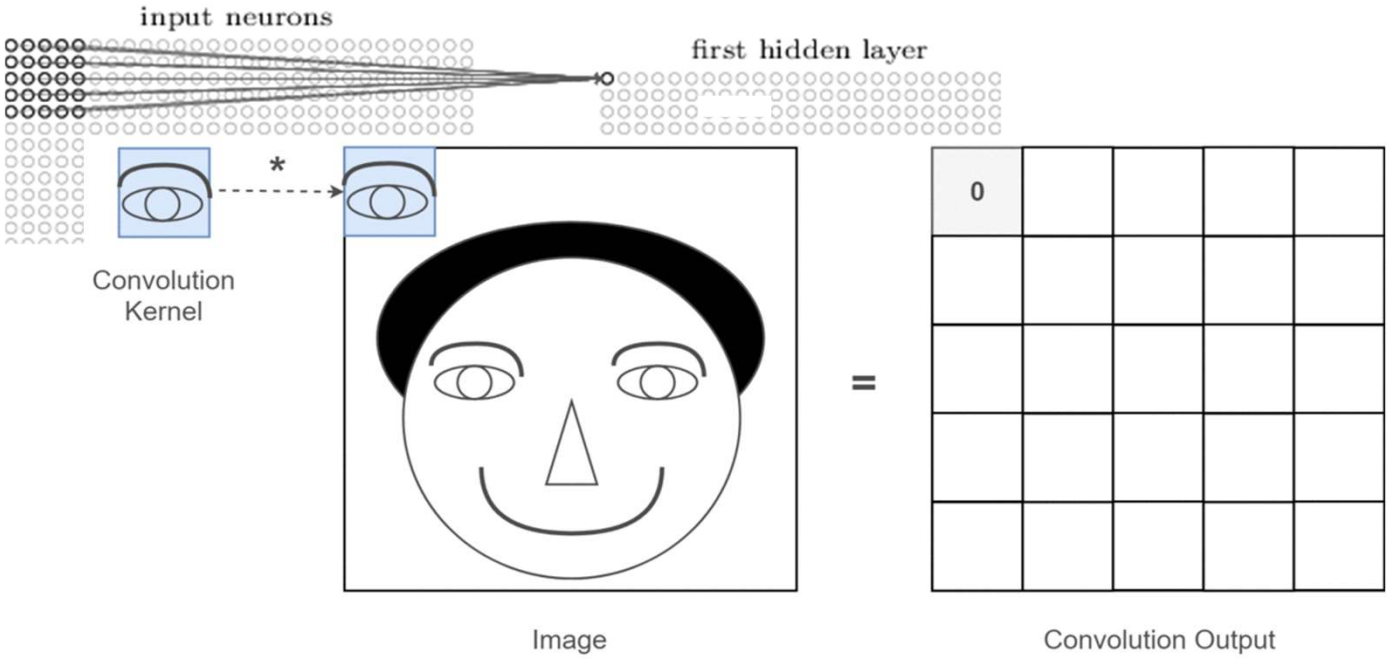


- ignorano la **struttura** del dato
- **scalano** terribilmente
- non apprendono **rappresentazioni gerarchiche**



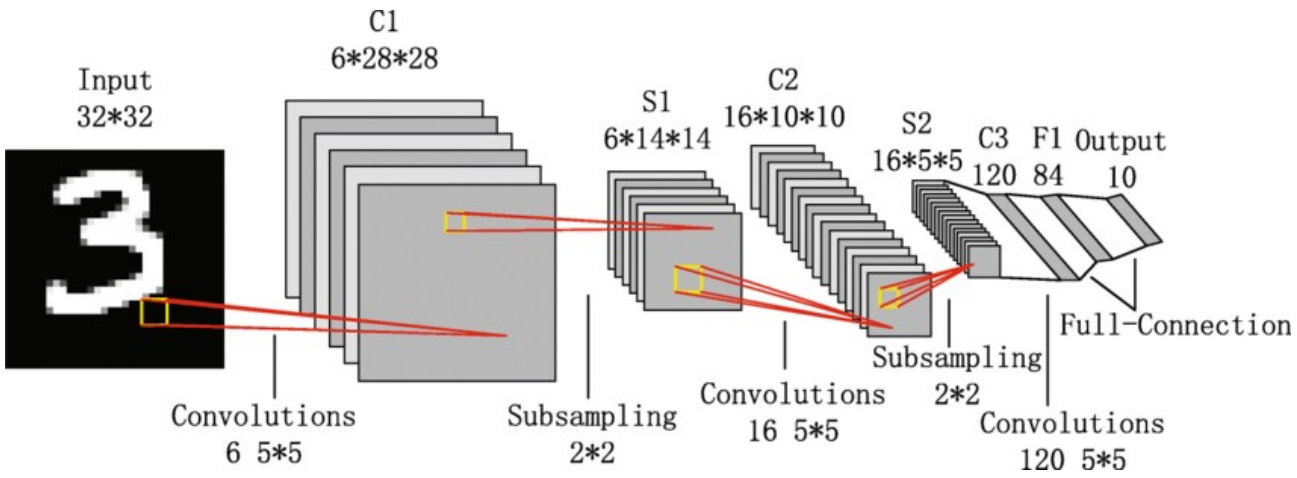
Reti Convoluzionali (CNN)

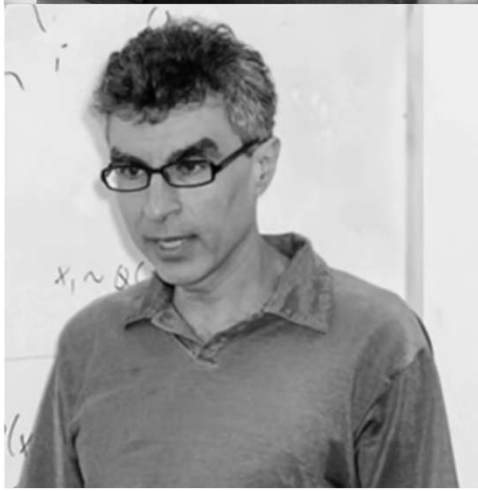
- sfruttano la **struttura** dei dati (ad es. **immagini**)
  - le connessioni e dunque l'elaborazione non avvengono più a livello del singolo neurone, ma "*a blocchi*" attraverso **filtri convoluzionali**



Reti Convoluzionali (1990-2000)

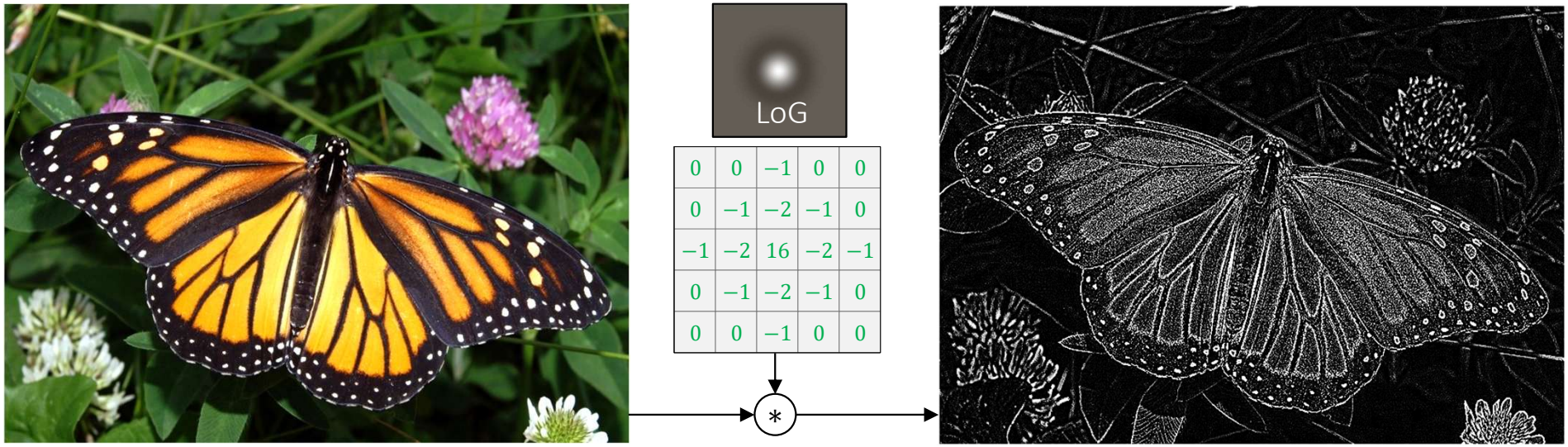
- sfruttano la **struttura** spaziale (o temporale) dei dati
  - riconoscimento di **immagini** (*LeCun & Bengio*, 1990)
    - prima applicazione: cifre scritte a mano su assegni bancari
  - riconoscimento del **parlato** (*LeCun & Bengio*, 1995)





Filtri convoluzionali (1/n)

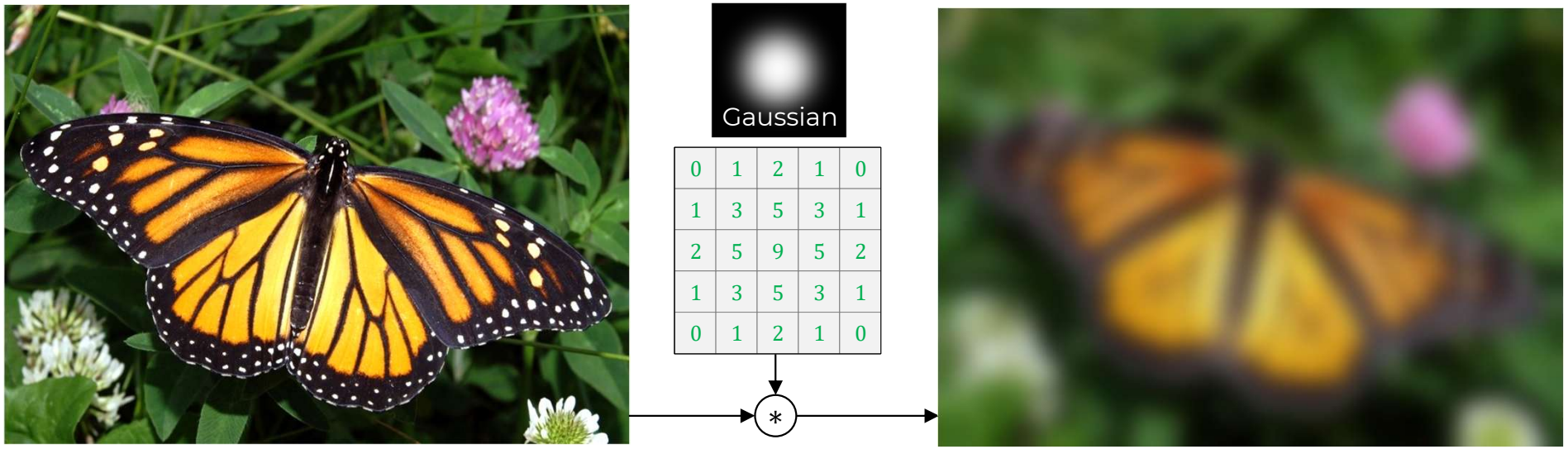
- imparano automaticamente quali misure caratteristiche estrarre
  - i **parametri** addestrabili sono i pesi W_{ij} del filtro, ovvero il filtro stesso



Questo filtro convoluzionale estrae i bordi, ovvero una misura di complessità della texture

Filtri convoluzionali (2/n)

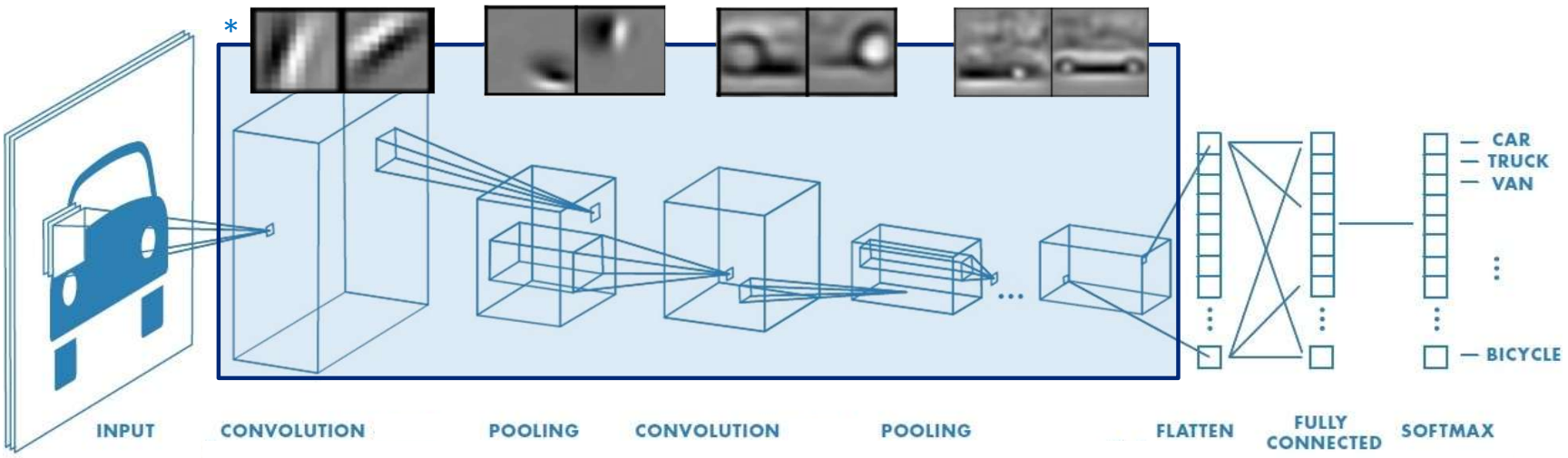
- imparano automaticamente quali misure caratteristiche estrarre
  - i **parametri** addestrabili sono i pesi W_{ij} del filtro, ovvero il filtro stesso



Questo filtro convoluzionale sfoca l'immagine, ovvero estrae i colori dominanti

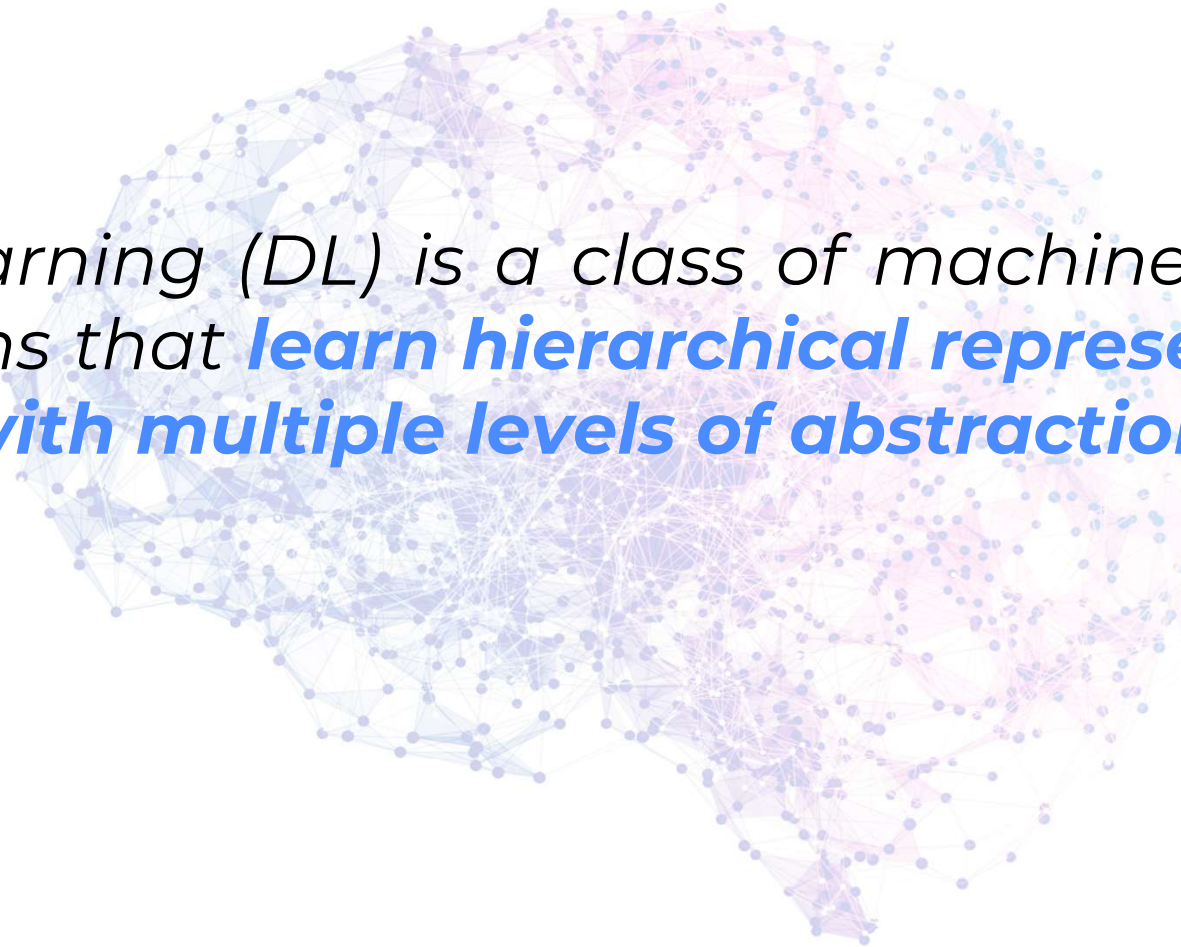
Apprendimento gerarchico

- una CNN che individua **veicoli** su immagini impara, nell'ordine, i "concetti" di *bordi*, *curve*, *parti di veicoli*, *tipi di veicoli*, ecc.
  - l'apprendimento **gerarchico** è reso possibile attraverso riduzioni di scala (*pooling*)



estrazione gerarchica di misure *caratteristiche*

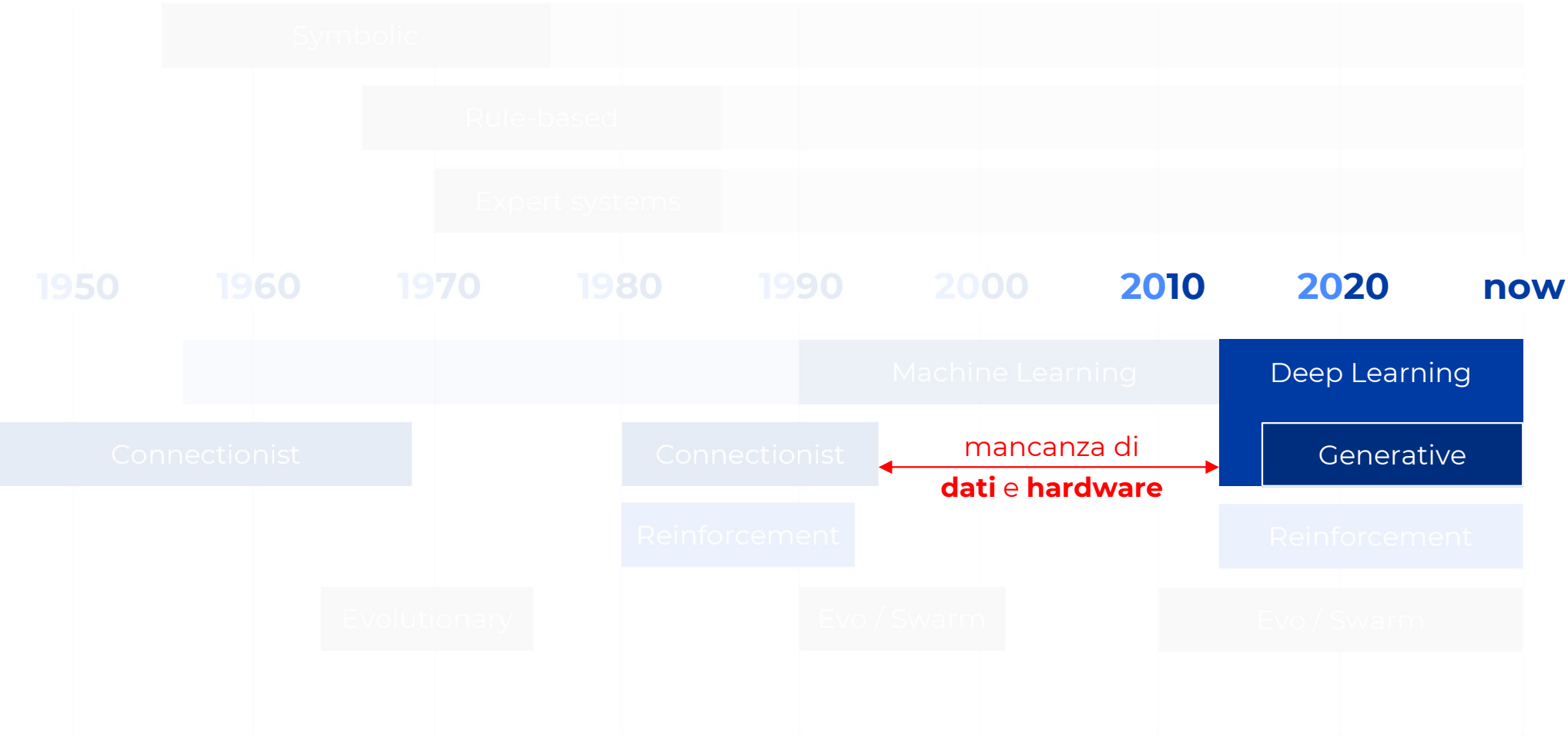
decisione



"Deep learning (DL) is a class of machine learning algorithms that ***learn hierarchical representations of data with multiple levels of abstraction****."*

Rina Dechter, **1986**

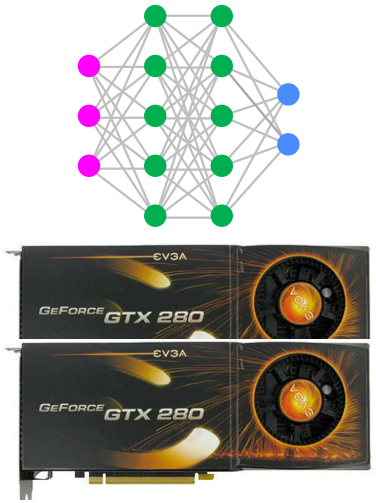
Deep Learning (2012-now)



Accelerazione GPU (2007-2009)

- *Gaurav Khanna* (MIT) utilizza 16 **PlayStation 3** (con chip grafico nVIDIA RSX) per simulazioni di collisioni tra buchi neri
- *Jensen Huang* (nVIDIA) rilascia **CUDA** per general-purpose programming
- *Andrew Ng* (Stanford) ottiene 70 x speedup con 2 nVIDIA GeForce GTX280





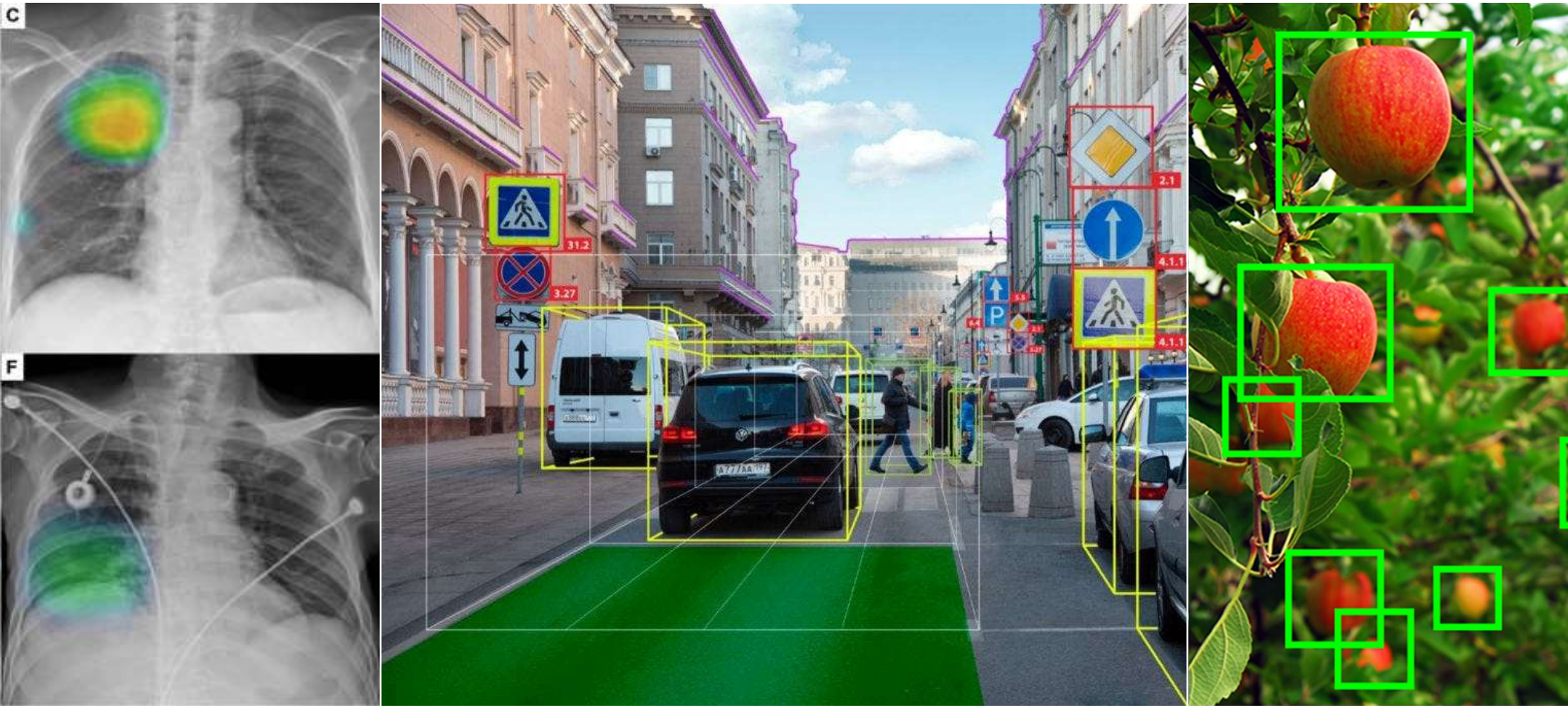
ImageNet (2009)

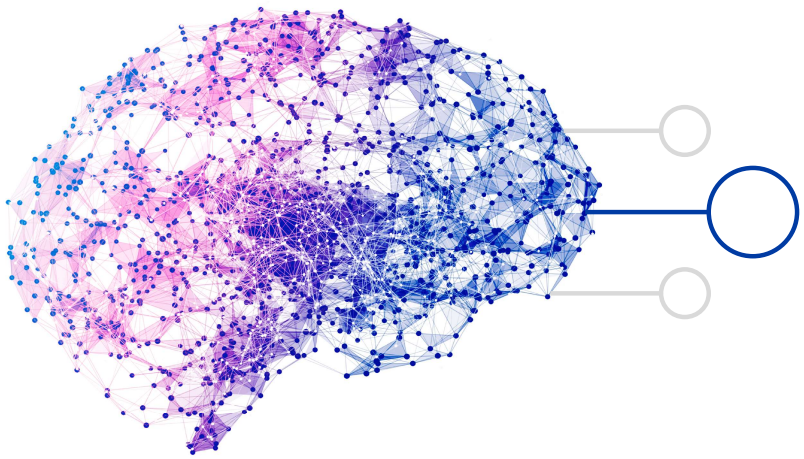
- **il più grande dataset di immagini mai creato**
  - 21,841 categorie e 14.3M immagini
  - realizzato in 3 anni grazie al progetto di *Fei Fei Li* (Stanford)

ILSVRC e AlexNet (2012)

- **ImageNet Large Scale Visual Recognition Challenge**
  - i risultati dell'edizione 2012 furono presentati a **Firenze** all'interno della European Conference for Computer Vision (**ECCV**)
  - il team di *Hinton* «SuperVision» vinse con un incremento di oltre il 63% rispetto alla prestazione ottenuta dai vincitori dell'edizione precedente
    - SuperVision = *Alex Krizhevsky* + *Ilya Sutskever* + *Geoffrey Hinton*

Visione artificiale (2012-now)





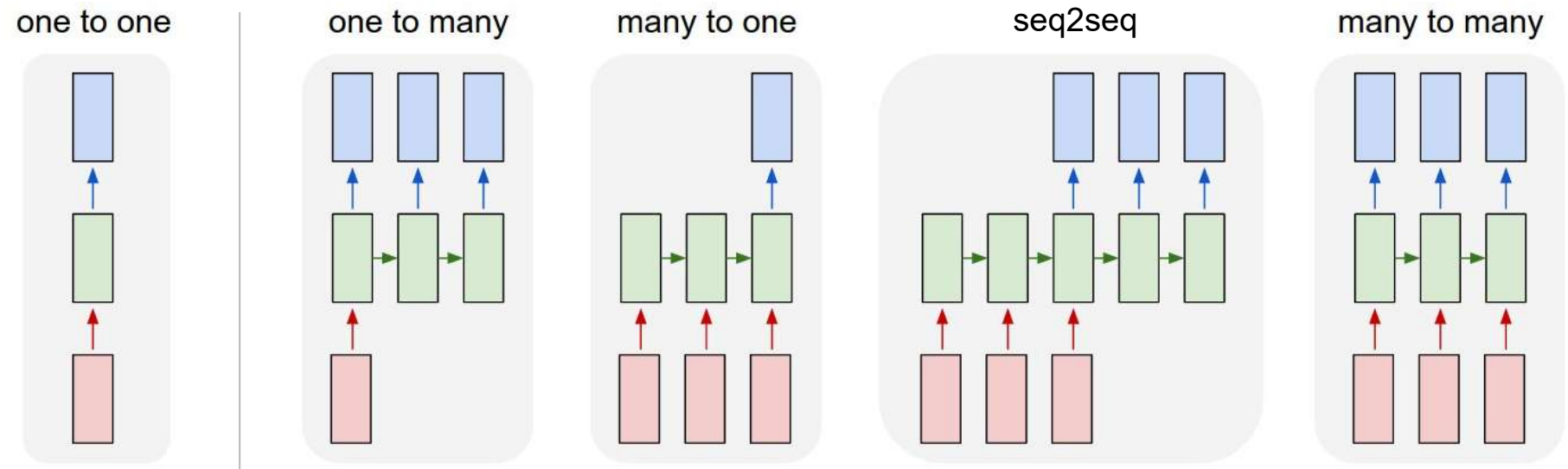
03

Come capiscono il linguaggio?

Reti attenzionali

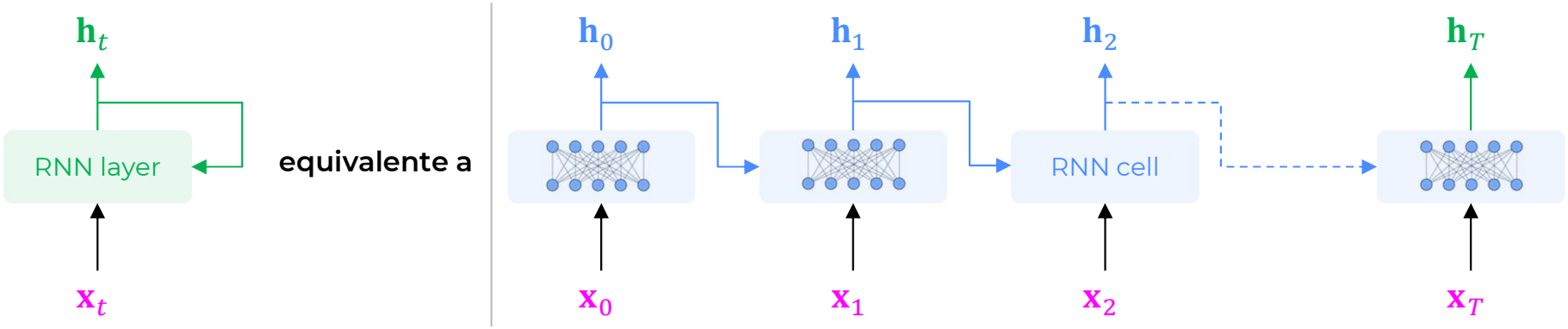
Limiti delle reti non sequenziali

- MLP e CNN sono progettate per input/output di **dimensioni fisse**
  - innumerevoli applicazioni si basano su sequenze di dimensioni variabili



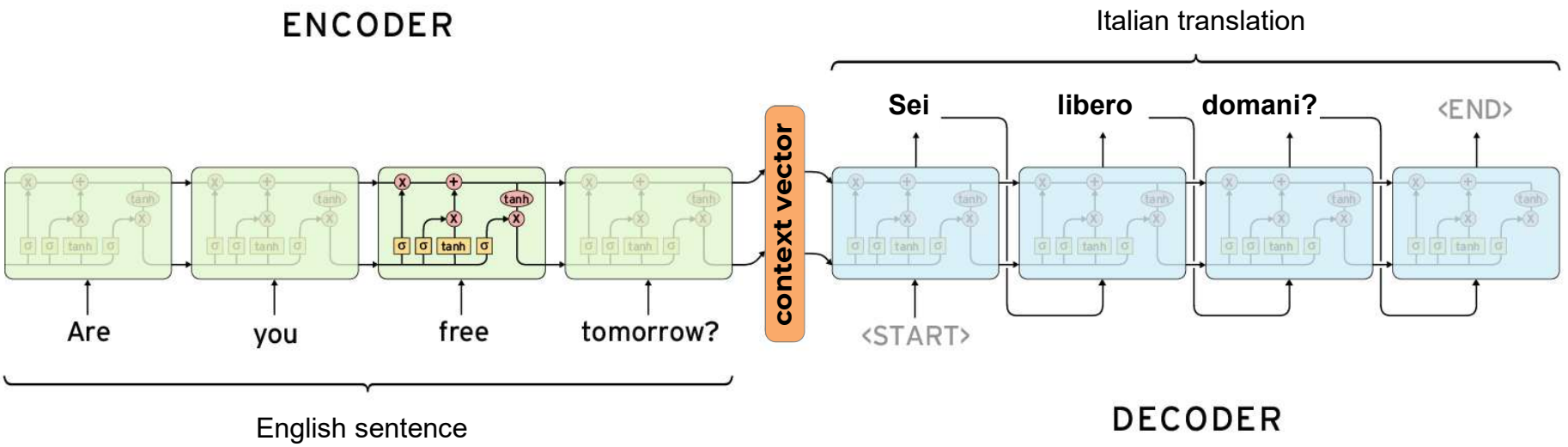
Reti Neurali Ricorrenti (RNN) (1997)

- la soluzione di riferimento fino al 2017
- **una rete neurale che si ripete** (ricorrente)
  - collegata a se stessa
    - dotata di un secondo input = rappresentazione interna $\mathbf{h}_t$ generata in output dalla stessa rete che ha processato l'elemento precedente
  - una sorta di «MLP sequenziale» che ripete e condivide se stessa per tutta la lunghezza T della sequenza

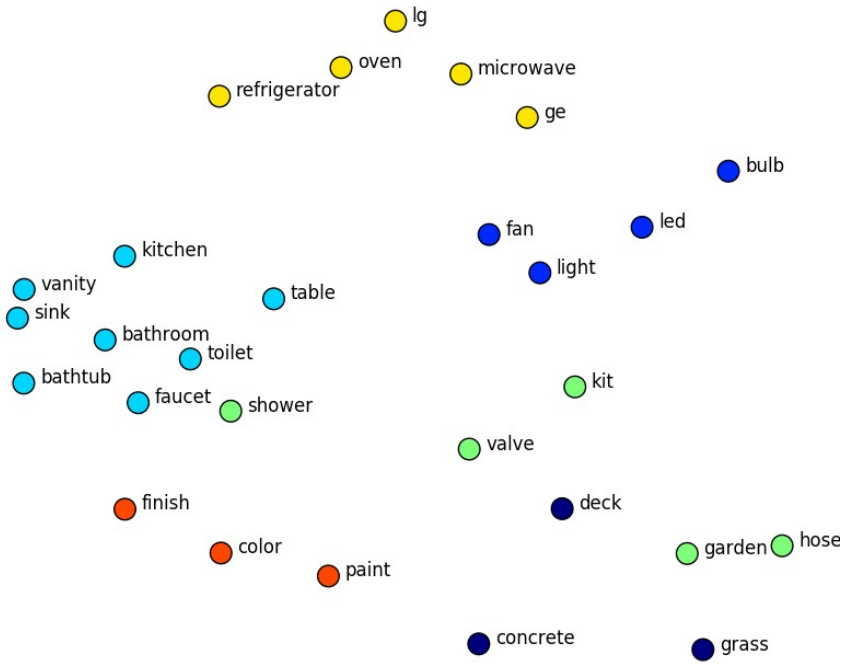


Encoder-decoder RNN

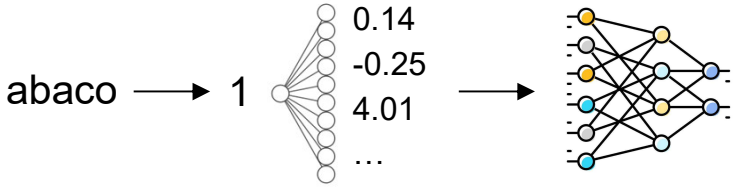
- noto anche come **seq2seq** (*Sutskever*, 2014)
  - elaborazioni sequenziali di tipo ***many-to-many***
    - traduzione automatica, sintesi, riconoscimento del parlato
  - Google Translate (2016)



Embedding di parole



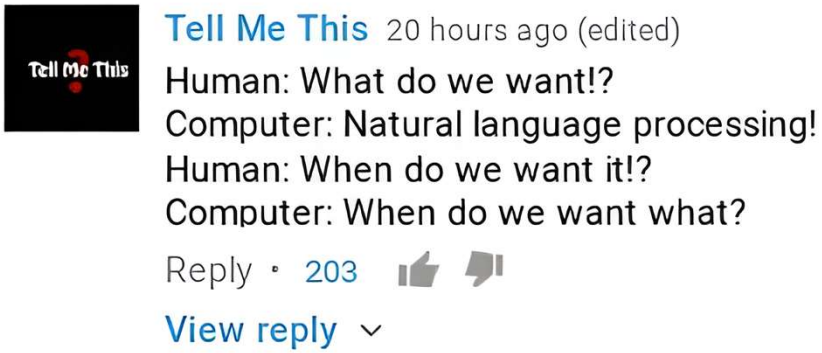
- trasformano **parole** in punti di uno **spazio multidimensionale**
  - parole che hanno una connessione semantica sono vicine nello spazio
    - dimensione $100 \sim 10000$ (LLM)
  - implementato mediante:
    - uno strato di neuroni addestrato assieme agli altri strati (Transformer, 2017)



    - modello neurale preaddestrato ad hoc
      - word2vec (*Mikolov*, 2013)
      - BERT (*Google*, 2019)

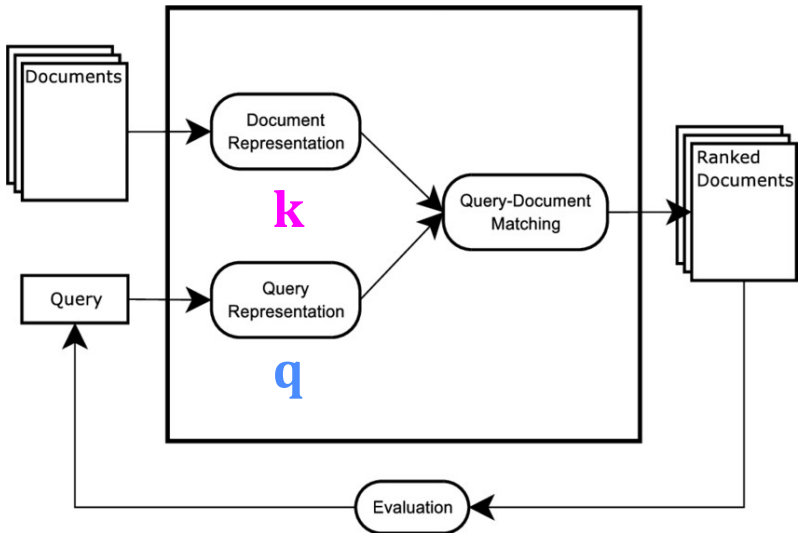
Limiti delle RNN

- hanno una **memoria a breve termine**
  - l'output $\mathbf{h}_t$ dell'ultima cella RNN della sequenza contiene solo le informazioni estratte dagli ultimi $20 \sim 40$ elementi
  - in altri termini, fanno fatica ad apprendere rappresentazioni del contesto



Information Retrieval (IR, 1970)

- i sistemi IR, a partire da una ***query*** di ricerca, producono una lista di ***documenti*** ordinati per rilevanza
  - i ***documenti*** sono rappresentati con un vettore $\mathbf{k}$ (***key***) di numeri $\in [0,1]$ che modellano la presenza di concetti
    - ad es. terra, mare, montagna, palazzi, ...
    - analoghi alle misure caratteristiche
  - le ***query*** sono rappresentate con un vettore $\mathbf{q}$ costruito allo stesso modo
  - questo consente di calcolare una misura di **similarità** tra $\mathbf{q}$ e $\mathbf{k}$

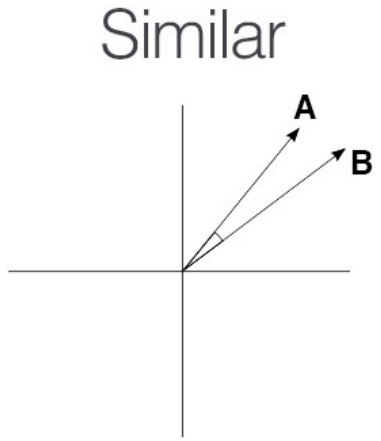


Google images

Prodotto scalare

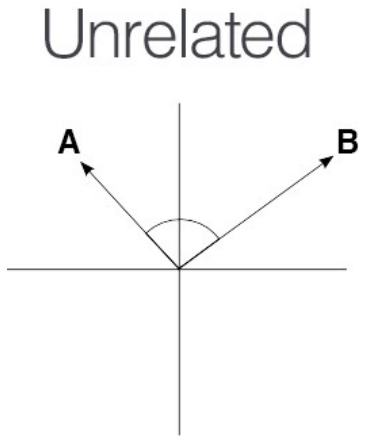
- il **prodotto scalare** $(\cdot)$ tra il vettore $\mathbf{q} = (q_1 \ \ldots q_d)^T$ e il vettore $\mathbf{k} = (k_1 \ \ldots k_d)^T$ è una misura di ***similarità***

$$\mathbf{q} \cdot \mathbf{k} \stackrel{\text{def}}{=} \sum_{j=1}^{d} q_j k_j = \mathbf{q}^T \mathbf{k} = (q_1 \ \ldots q_d) \begin{pmatrix} k_1 \\ \ldots \\ k_d \end{pmatrix}$$



- nello spazio euclideo, il prodotto scalare è proporzionale al **coseno** dell'angolo θ tra i due vettori

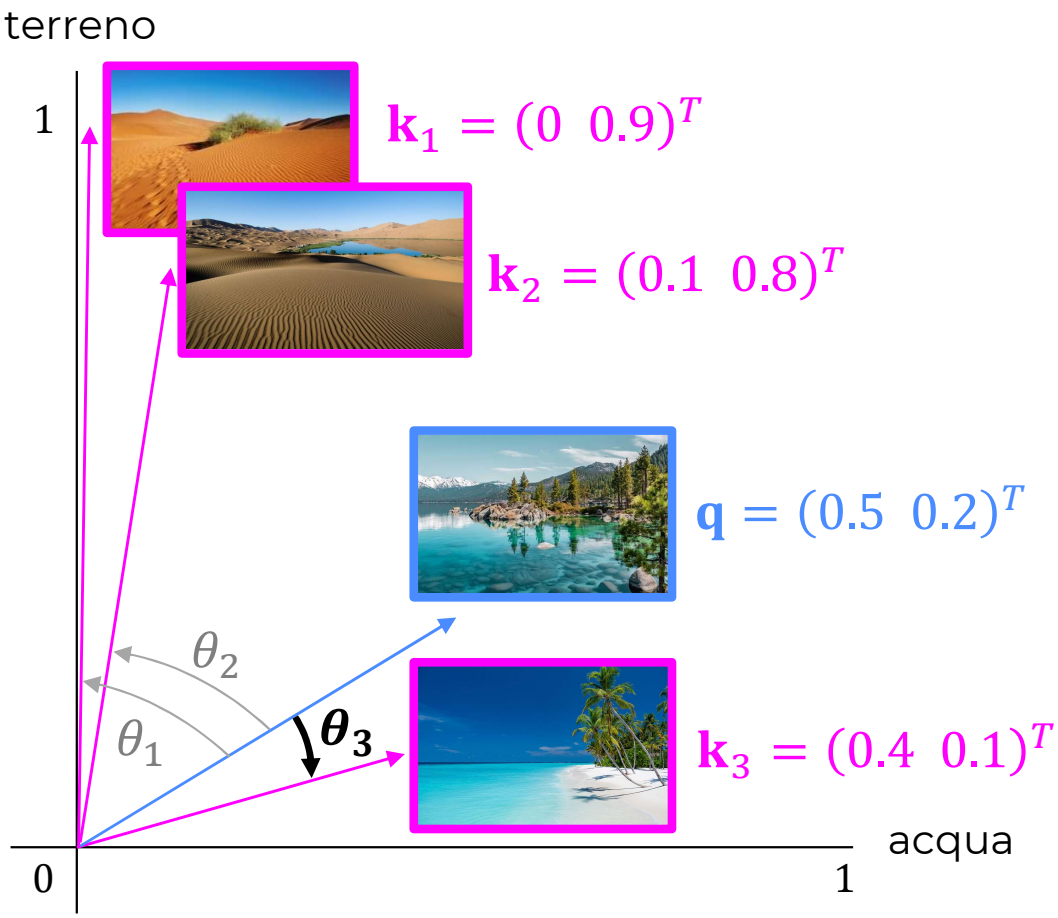
$$\mathbf{q} \cdot \mathbf{k} \stackrel{\text{def}}{=} \|\mathbf{q}\| \, \|\mathbf{k}\| \cos(\theta)$$



Prodotto scalare

Esempio giocattolo:
Motore di ricerca di immagini
con dimensione dello spazio

$d = 2$



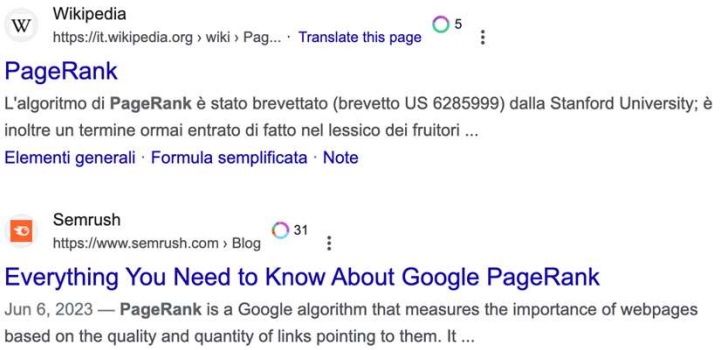
Esempio reale:

Google images

$d \gg 2$

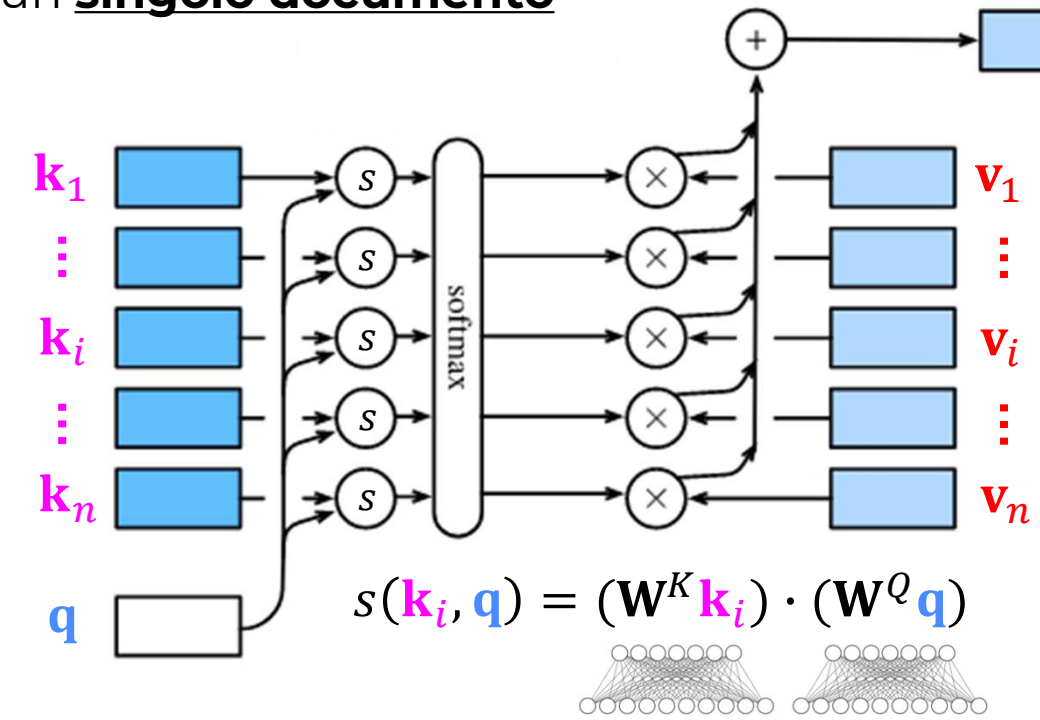
Google?

- $\mathbf{q}$ è la ***query***, il testo che l'utente digita nella barra di ricerca
- i ***documenti*** $\mathbf{v}_i$ (*values*) sono le pagine web nel database, ciascuna rappresentata con $\mathbf{k}_i$
  - $\mathbf{k}_i$ contiene misure caratteristiche, ad es. presenza di concetti e parole chiave
- $\mathbf{q}$ viene confrontato con ciascuna $\mathbf{k}_i$ producendo una misura di **similarità** s_i
- il risultato è una **lista di documenti** (pagine) ordinate per $s_i \downarrow$ (*rilevanza*)



Meccanismo attenzionale

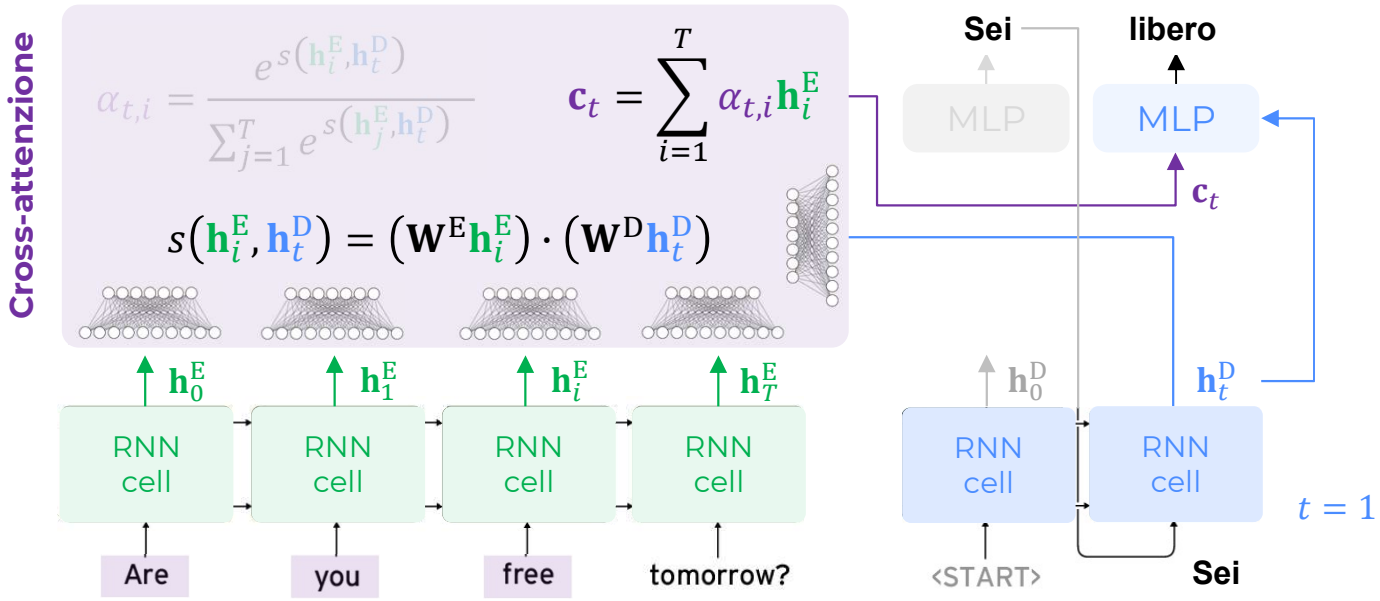
- il risultato è un **singolo documento**



$\mathbf{W}^K$ e $\mathbf{W}^Q$ sono **trasformazioni** implementate mediante piccole **reti neurali** (2 layer) che mappano $\mathbf{k}_i$ e $\mathbf{q}$ in uno spazio dove sono meglio confrontabili

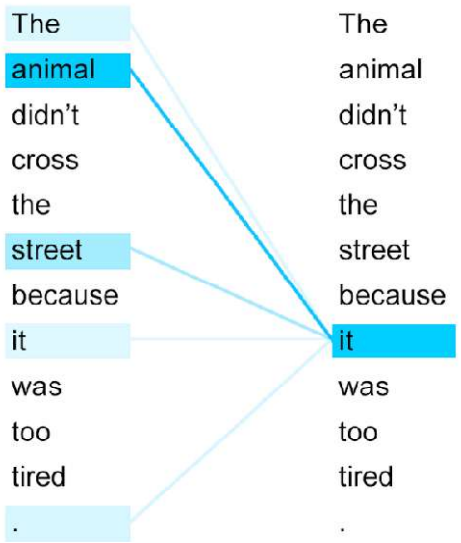
Cross-attenzione

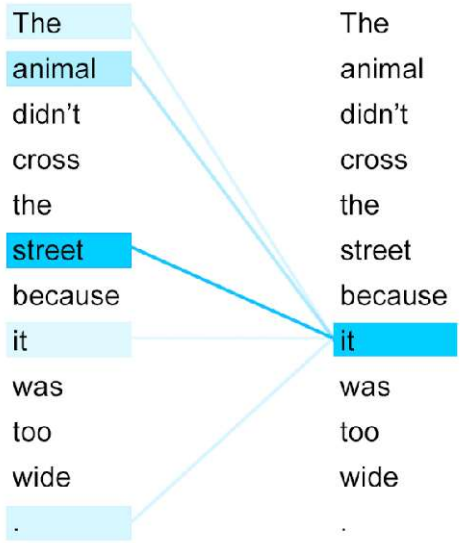
- per **attenzionare *una*** sequenza B mentre si elabora ***un'altra*** sequenza A
  - **keys** e **values** da B
    - ad es. rappresentazioni $\mathbf{h}_i^{\mathrm{E}}$ estratte da un Encoder RNN dal linguaggio sorgente
  - **query** da A
    - ad es. rappresentazioni $\mathbf{h}_t^{\mathrm{D}}$ estratte da un Decoder RNN dal linguaggio target



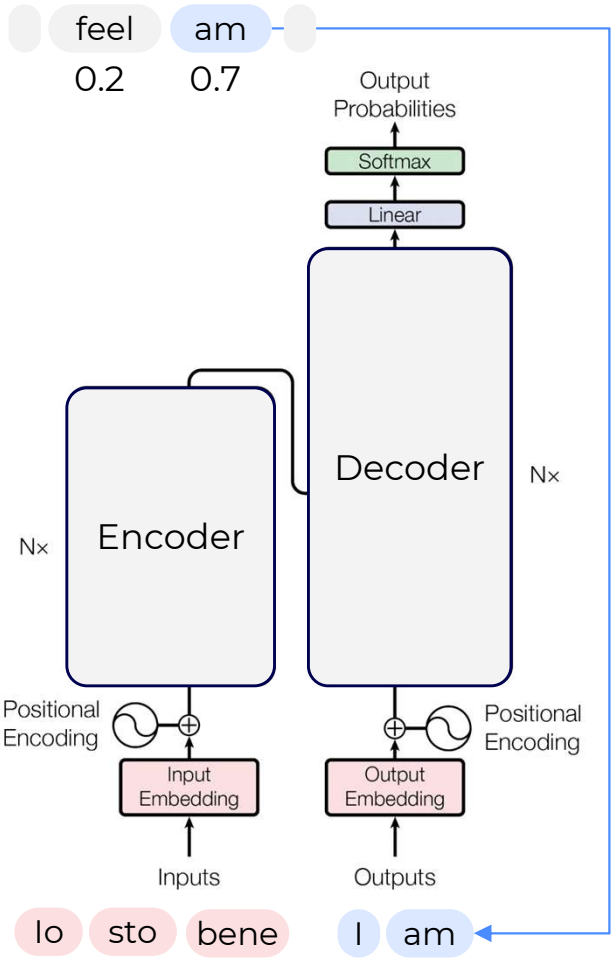
Self-attenzione

- **query** ≡ **key** ≡ **value**
  - chiedo al sistema di trovare similarità (**correlazioni**) tra un dato e...se stesso?
  - se il dato è testuale, il sistema impara a...comprendere il linguaggio!



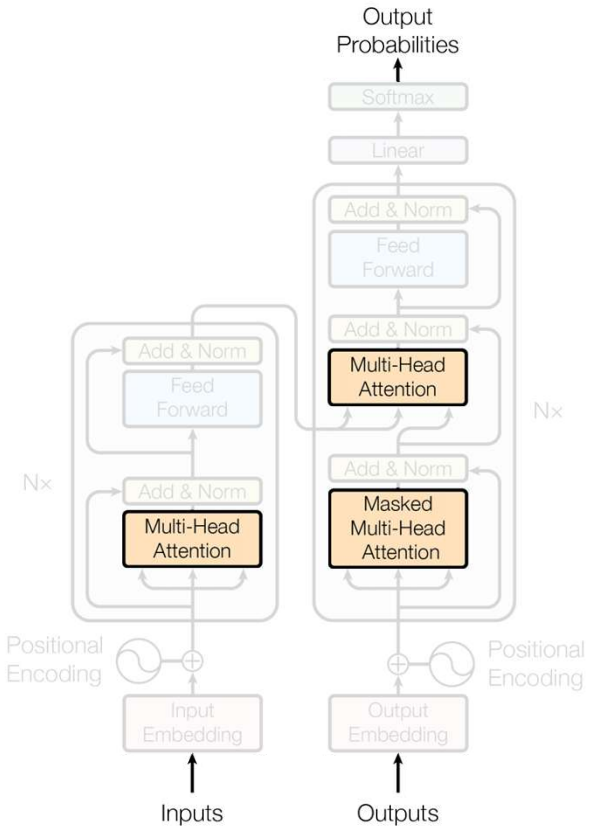


Transformer (1/3)

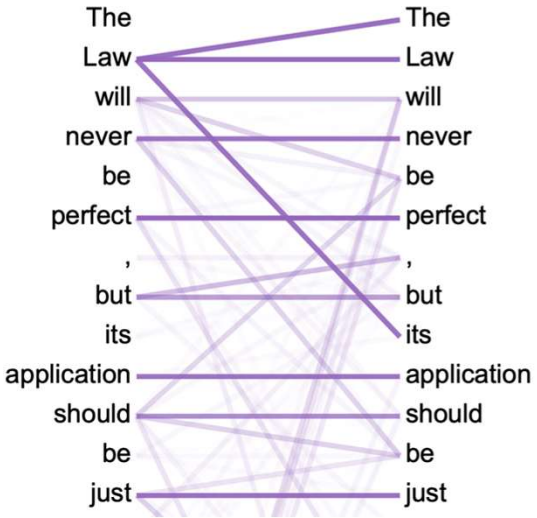


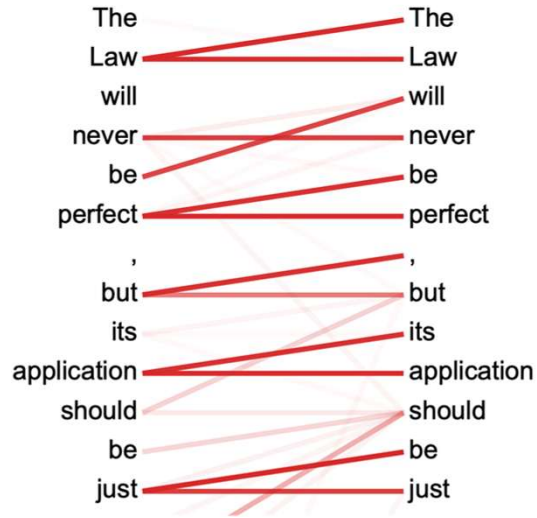
- modello per la **traduzione** (Google , 2017)
  - autoregressivo
    - man mano che la rete genera le parole nella nuova lingua, queste diventano l'input per predire la prossima
  - predittivo
    - l'output layer ha tanti neuroni quante sono le possibili (sotto)parole (***token***)
- encoder e decoder usano la **self-attenzione**
  - imparano le correlazioni intrinseche in una frase per catturare le misure caratteristiche di quella lingua
- il decoder usa la **cross-attenzione**
  - impara le correlazioni tra le due lingue per catturare le misure caratteristiche di trasformazione da una lingua all'altra

Transformer (2/3)

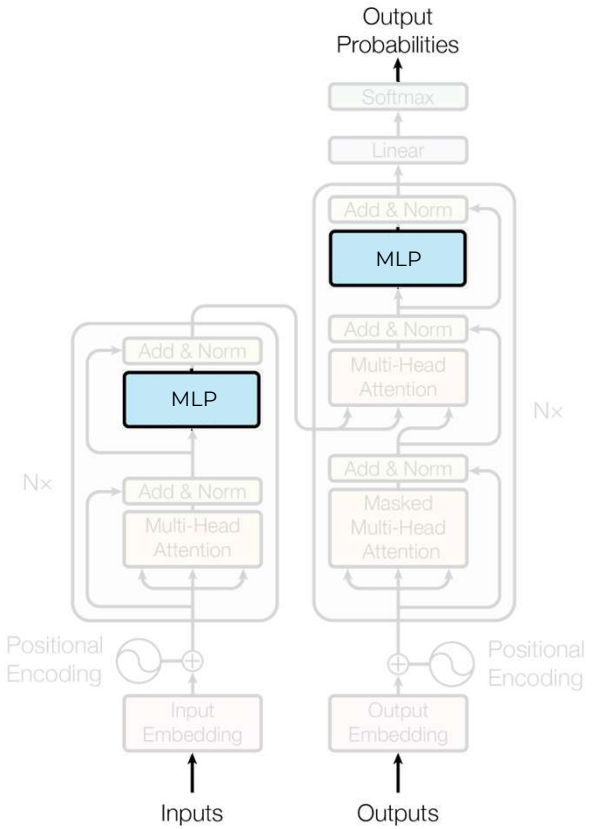


- i meccanismi attenzionali sono **multi-head**
  - apprendono in parallelo diverse trasformazioni $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$ per estrarre **diversi tipi** di misure caratteristiche
    - come le CNN!

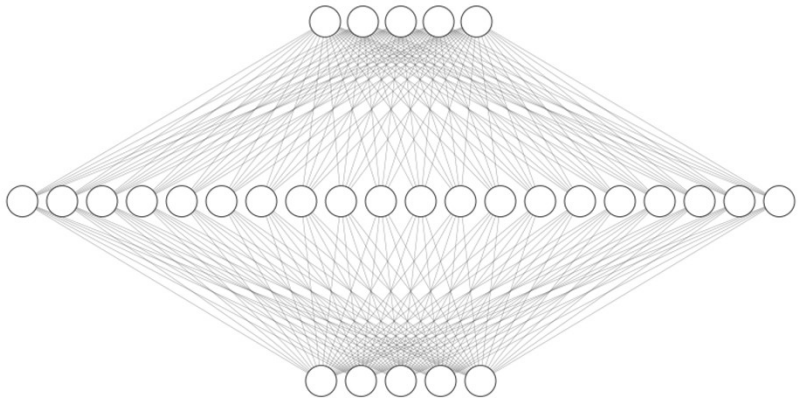




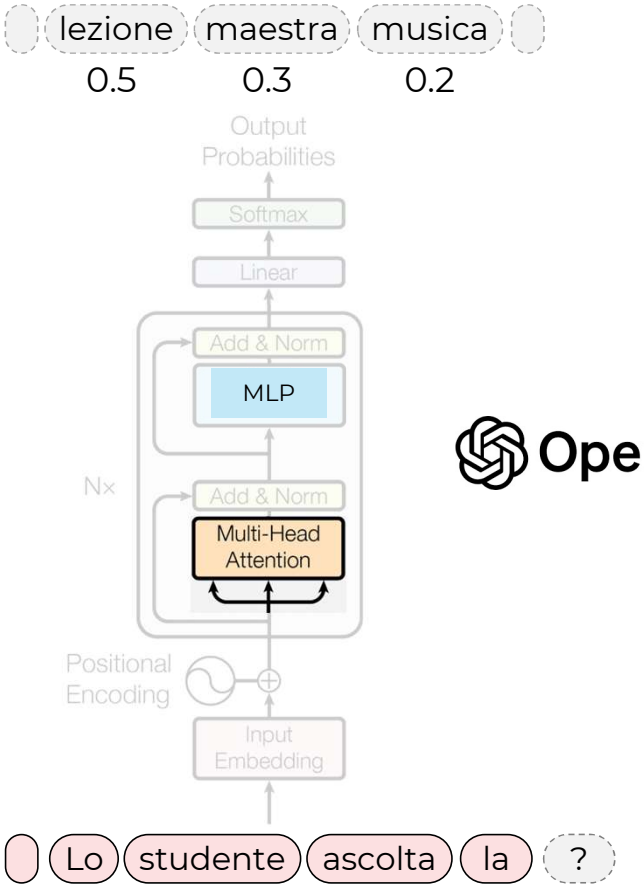
Transformer (3/3)



- gli **MLP** introducono ulteriori trasformazioni per aumentare la capacità della rete
  - tipicamente 3 layer con fattore di espansione 4



GPT (2018)

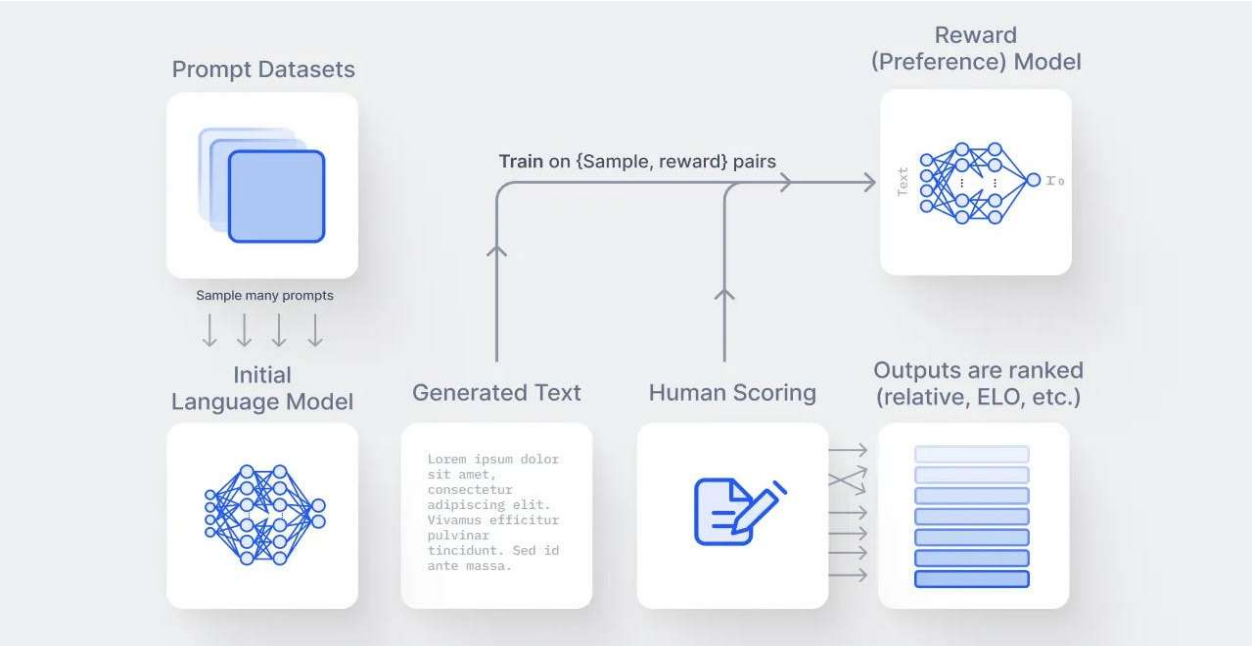


- sfrutta una parte del Transformer per **predire la prossima parola** nella frase
  - GPT 1 addestrato su *BookCorpus* (≈ 7.000 libri)
  - GPT 2 addestrato su *WebText* (≈ 8M di pagine web)
  - GPT 3 addestrato su *CommonCrawl* (≈ 45 TB di testo dal web) + *WikipediaEN* + *BookCorpus*

OpenAI

Large Language Model (LLM)

- Transformer addestrati su **centinaia di Terabyte di testi**
  - ~ 5% di tutto il testo disponibile (100% entro il 2028)
- *allineati* ovvero resi coerenti **valori e preferenze umane**
  - ad es. mediante Reinforcement Learning from Human Feedback



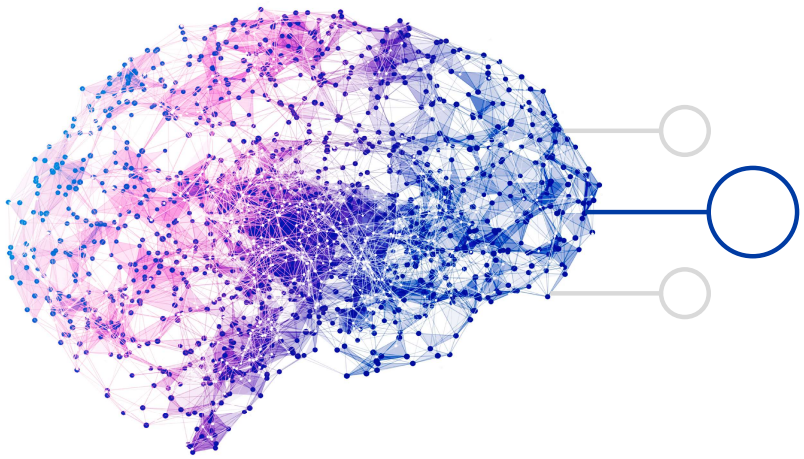
ChatGPT

Claude

Gemini

Grok

deepseek



04

Come generano contenuti?

Diffusione guidata

Diffusione termodinamica

- *sistema fisico che parte da uno stato ordinato e a **bassa entropia**, evolvendo gradualmente verso un equilibrio disordinato e ad **alta entropia***

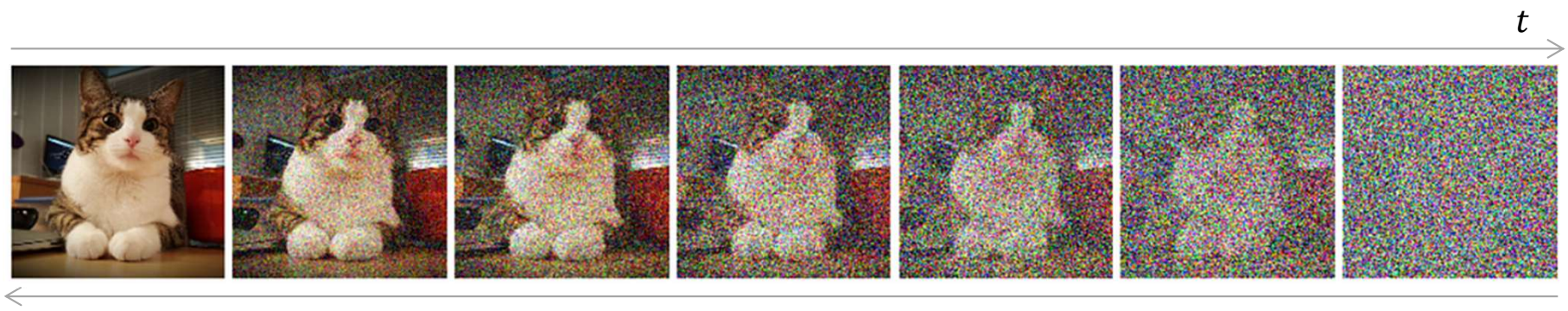
- ordine ⟶ caos
  - col tempo, tutta l'acqua apparirà blu e non potremo più distinguere «liquido naturalmente blu» da «acqua in cui è caduta una goccia di inchiostro blu»

Diffusione inversa

- è teoricamente possibile (ma difficile) **invertire** questo processo
  - caos ⟶ ordine
    - richiederebbe la conoscenza degli stati microscopici e delle interazioni molecolari
  - a che scopo?
    - stimare lo stato iniziale («liquido blu» vs. «acqua in cui è caduta una goccia blu»)

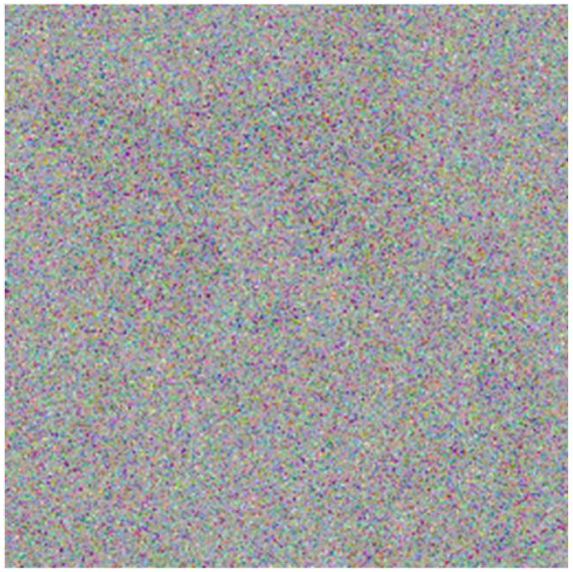
Diffusione su immagini

- aumentare l'entropia di un'immagine = aggiungere rumore
- **idea (addestramento)**
  - **aggiungo progressivamente rumore** ad un'immagine per ottenere tutti gli stati
    - compreso lo stato di «caos totale» ovvero puro rumore $\mathbf{z}$ («liquido blu»)
  - addestro una **rete** CNN per **invertire il processo** di diffusione ovvero
    - data l'immagine I_t, la CNN predice il rumore da sottrarre a I_t per ottenere I_{t-1} $\forall t$
  - ripeto il processo per tante (milioni) di immagini (ad es. di gatti) del training set
    - la rete neurale imparerà l'essenza stessa dei dati di addestramento (ad es. gatti)



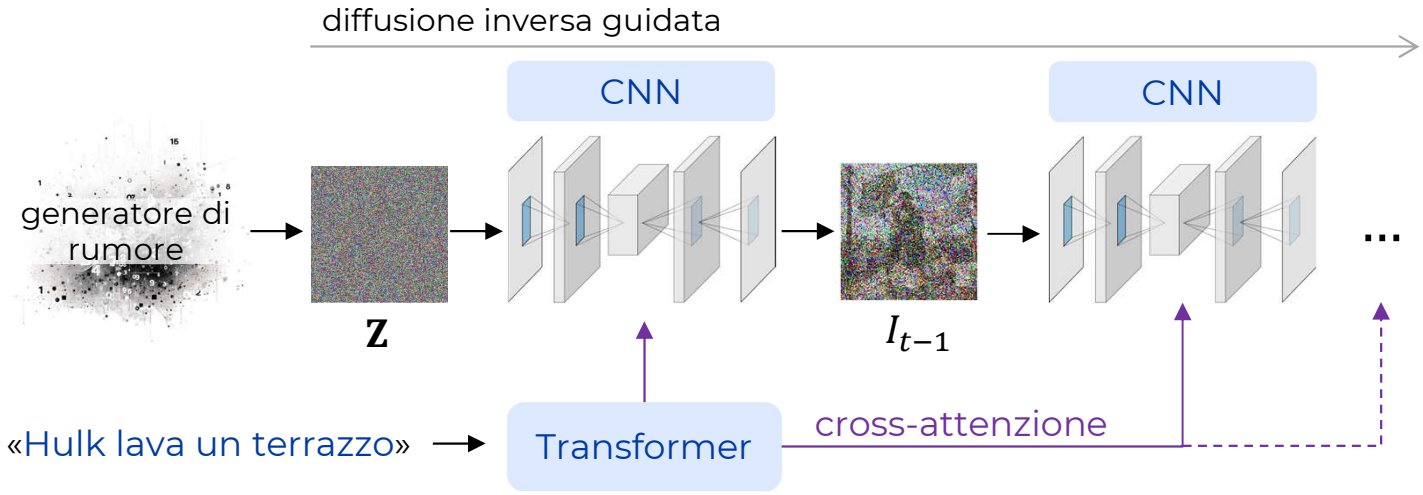
Generazione di immagini

- per generare un'immagine, fornisco puro rumore $\mathbf{z}$ e inverto la diffusione mediante la CNN precedentemente addestrata
  - ciascuna immagine di puro rumore $\mathbf{z}$ genererà un'immagine unica
  - non è possibile controllare *cosa* verrà generato, dipende dai dati di addestramento



Diffusione guidata (2023-oggi)

- in **addestramento** le immagini sono accompagnate da **descrizioni testuali**
  - la CNN usa un Transformer per *fondere* mediante meccanismi attenzionali il significato del testo con il contenuto dell'immagine
- in fase di generazione la descrizione testuale *guida* la CNN a trasformare il rumore puro $\mathbf{z}$ in un'immagine allineata al testo fornito dall'utente



2023


2024

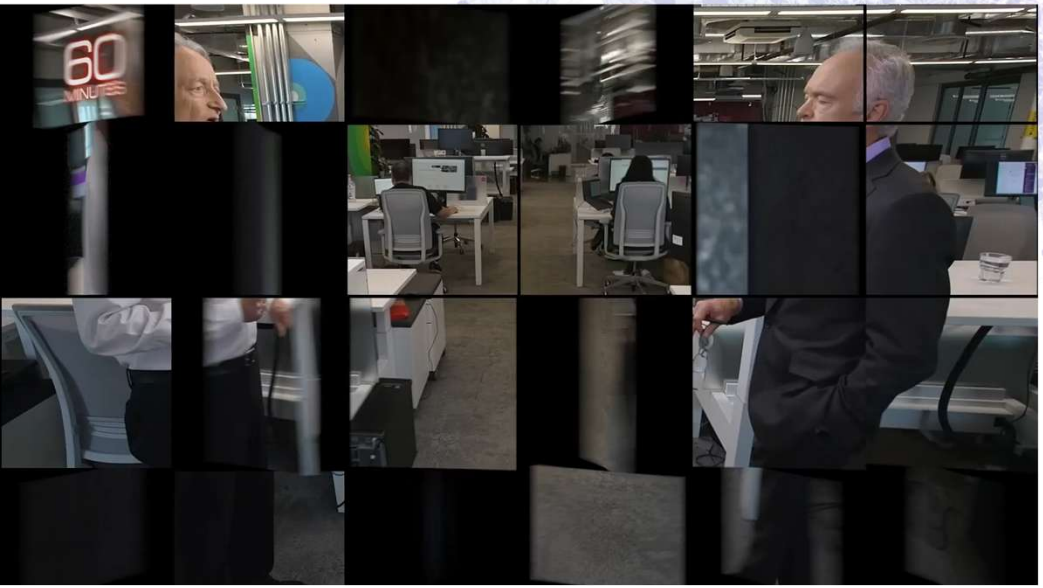
2025

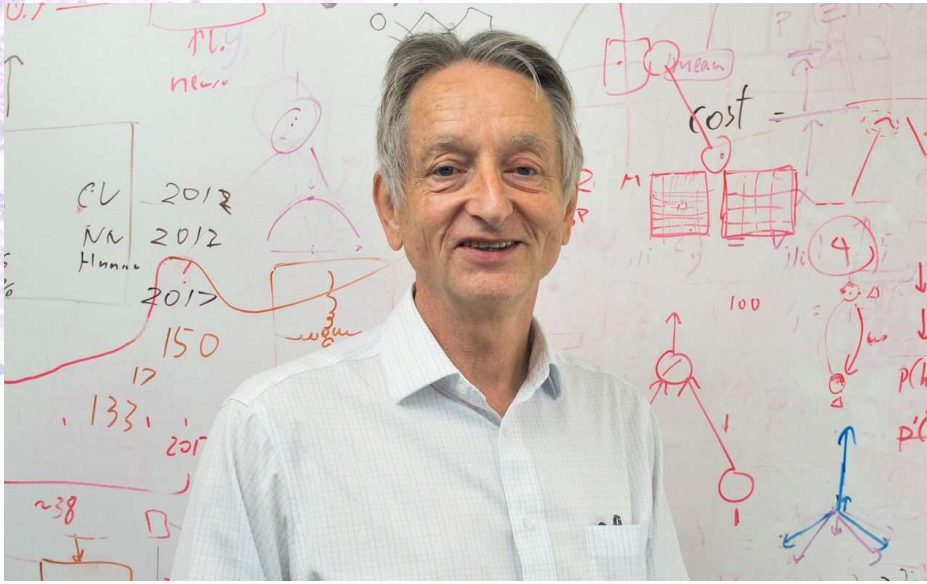

2026
AI generated

Conclusioni e riflessioni

“Per predire la prossima parola serve comprendere l’intera frase. ***Pensare che un sistema capace di farlo sia privo di intelligenza è un’idea folle****.”*

Geoffrey Hinton, **2023**








“*Genera un'immagine di un* ***bicchiere di vino pieno fino all’orlo***.” ChatGPT 4o




“*Genera un'immagine di un* ***bicchiere di vino pieno fino all’orlo***.”
ChatGPT 5

*“Nessuna **idea** può sorgere nella mente se non è preceduta da un’**impressione sensibile**, poiché ogni pensiero è solo una copia sbiadita della nostra **esperienza**”*

David Hume, **1740**