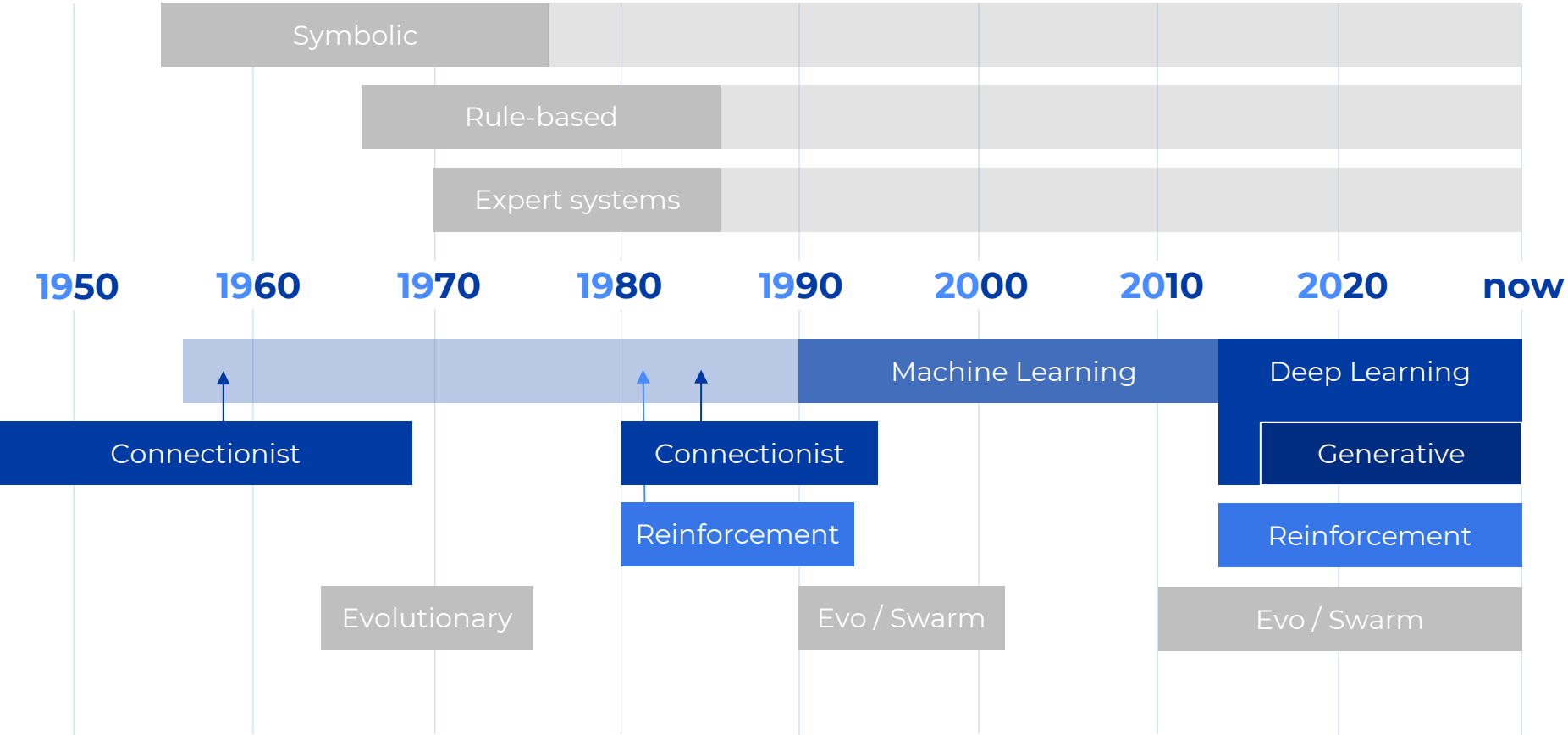
Deep Learning

Dal modello neurale alla generazione di contenuti

versione 1.0

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

Paradigmi di AI



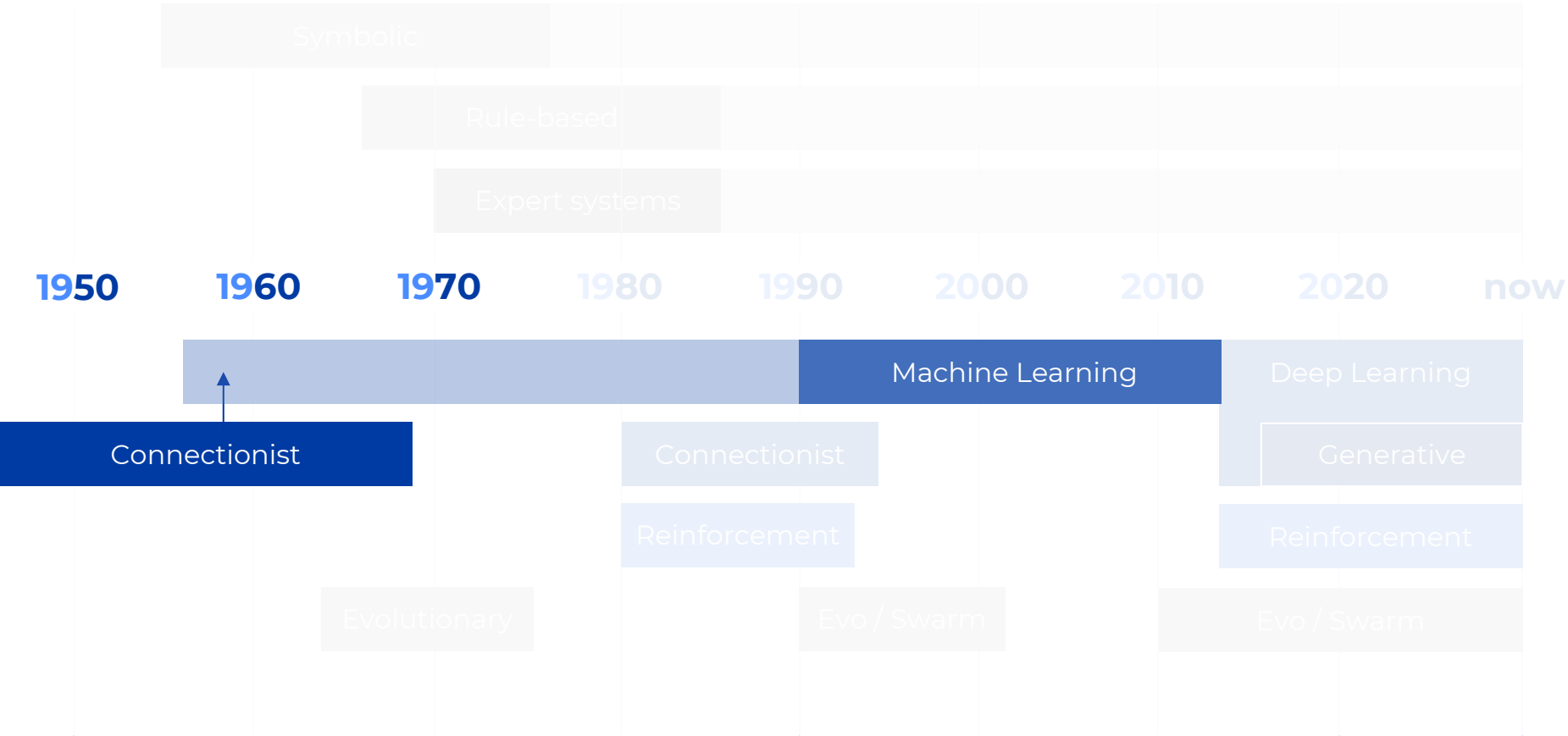
Sommario

01

Reti neurali artificiali

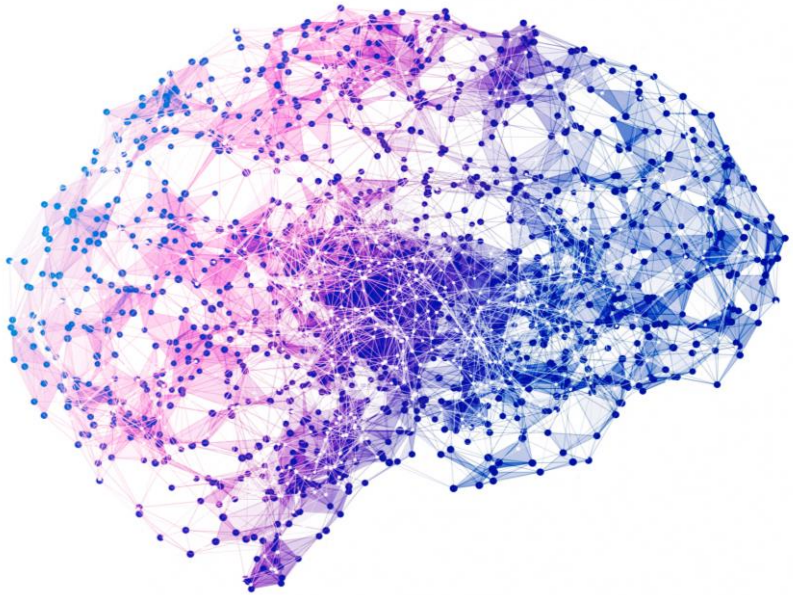
Neurone artificiale, funzione di errore, addestramento

Connessionismo (1943-1969)

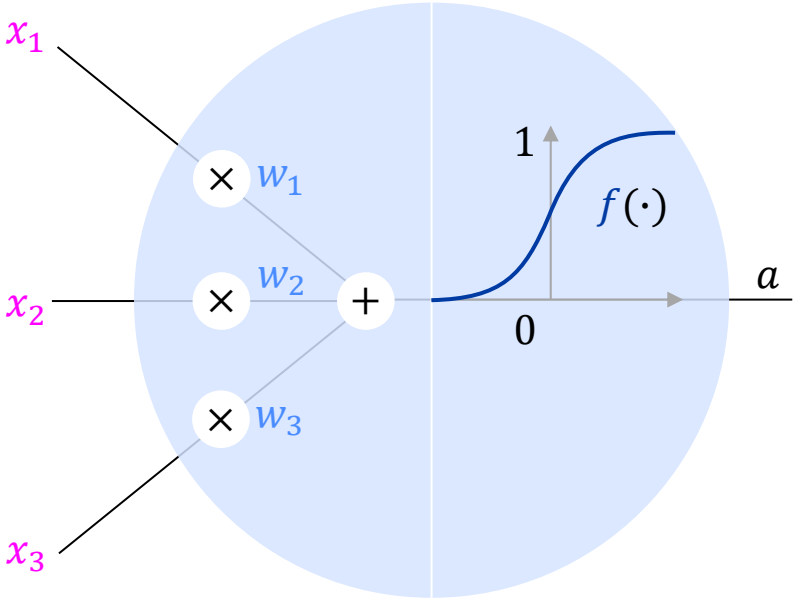


Connessionismo

- imita il funzionamento del cervello
  - «*I am my connectome*» (*Seung*: link)
- basato su neuroni artificiali interconnessi
- il paradigma AI più antico
  - confluito nel moderno Deep Learning



Neurone artificiale (1943-1958)



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

- input diversi hanno pesi w_i diversi
- i pesi sono i parametri che vengono modificati durante l'addestramento

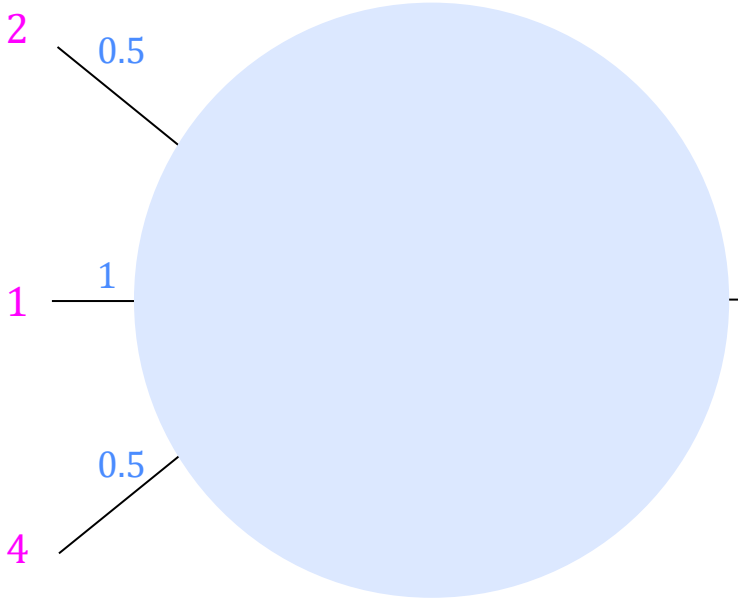
(2) gli input pesati vengono **sommati**

- così da ottenere un unico valore

(3) **attivazione**

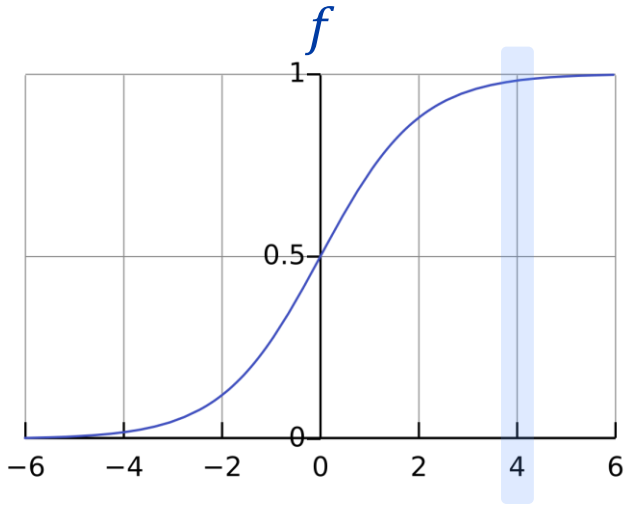
- il valore viene mappato da una funzione di attivazione f in un range tra
  - 0 (neurone spento)
  - 1 (neurone attivo)

Neurone artificiale (1943-1958)

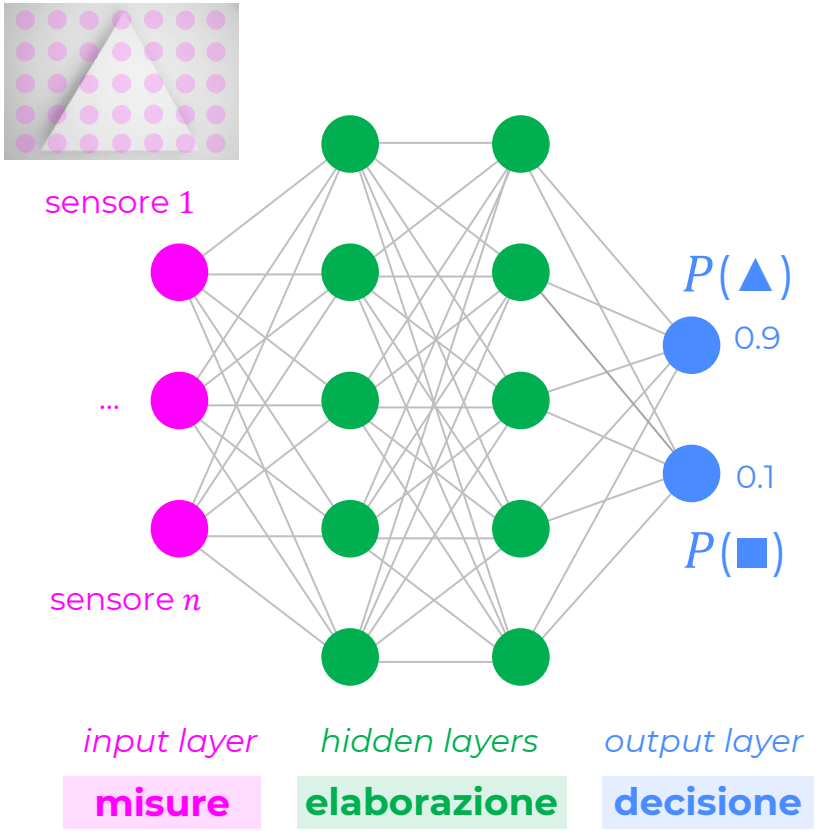


$$a = f\left(\sum w_i x_i\right)$$

$$\approx 1$$



Multi-Layer Perceptron (1958)



- costituite da stratificazioni (*layer*) di **neuroni** artificiali **interconnessi**
- lo schema delle connessioni determina l'**architettura** della rete:
  - *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 (1958)

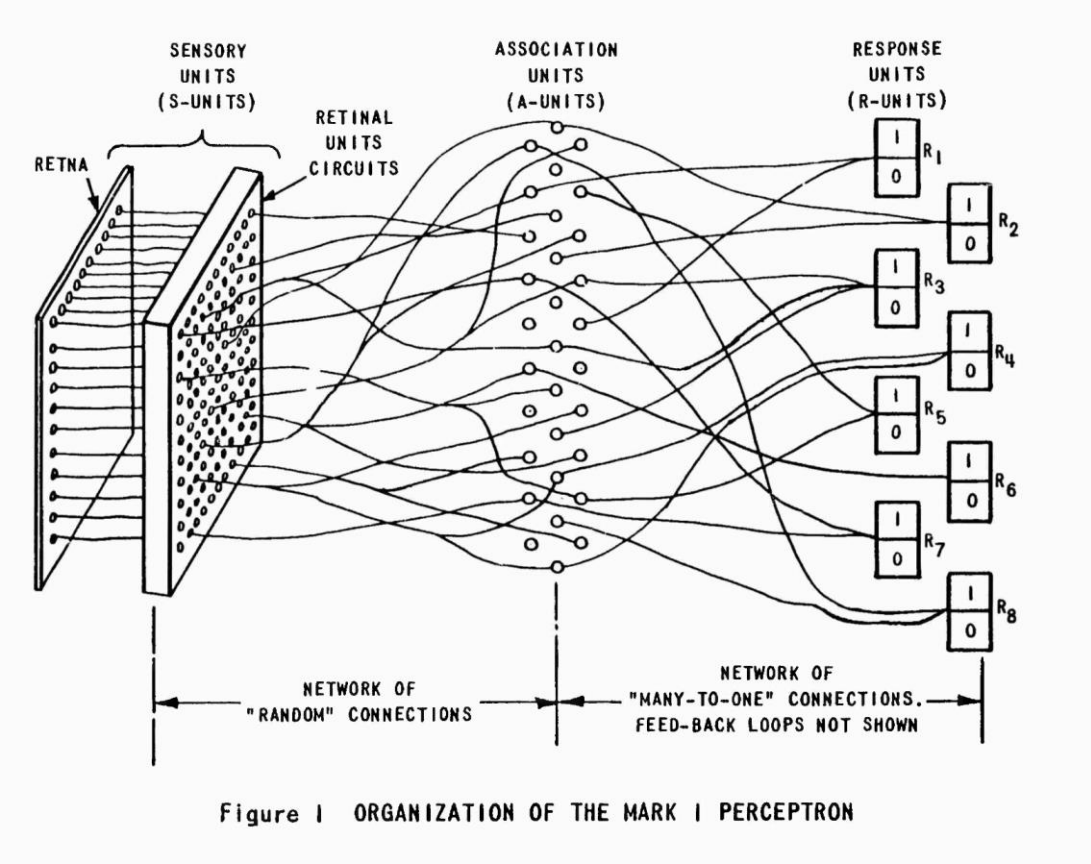
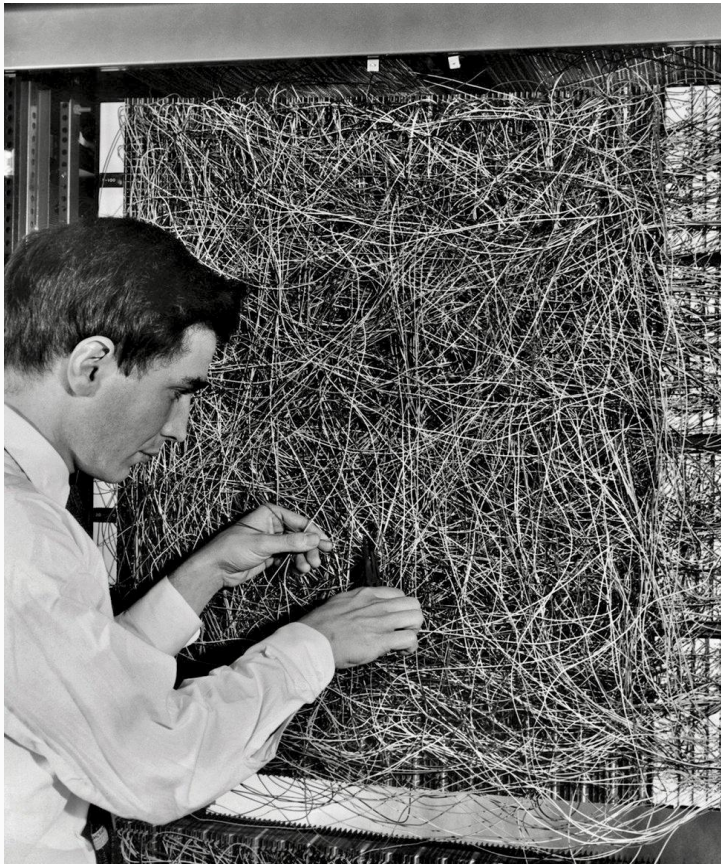
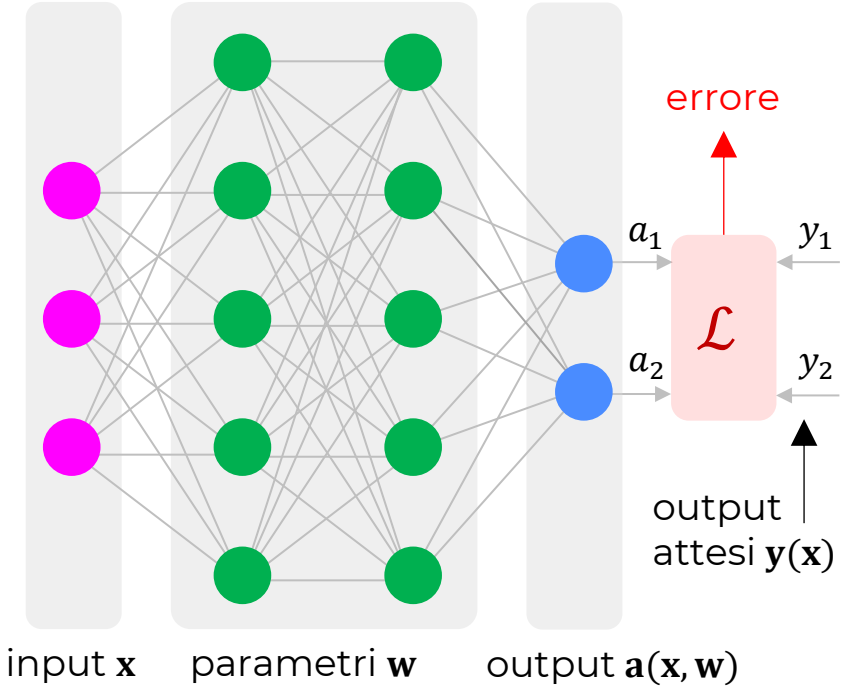


Figure 1 ORGANIZATION OF THE MARK 1 PERCEPTRON



Funzione di errore (1960)

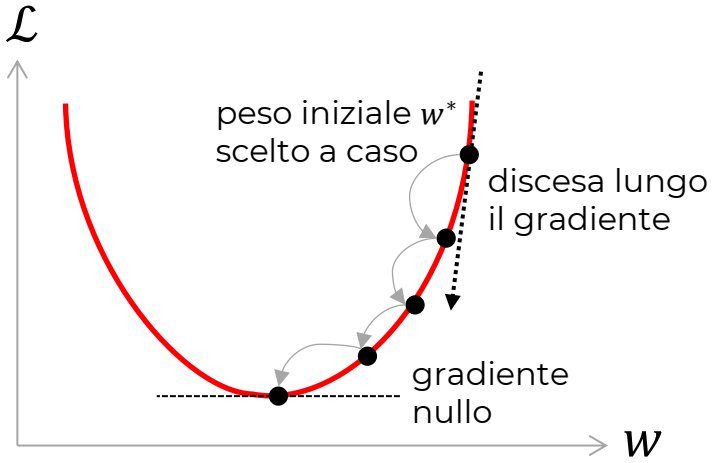


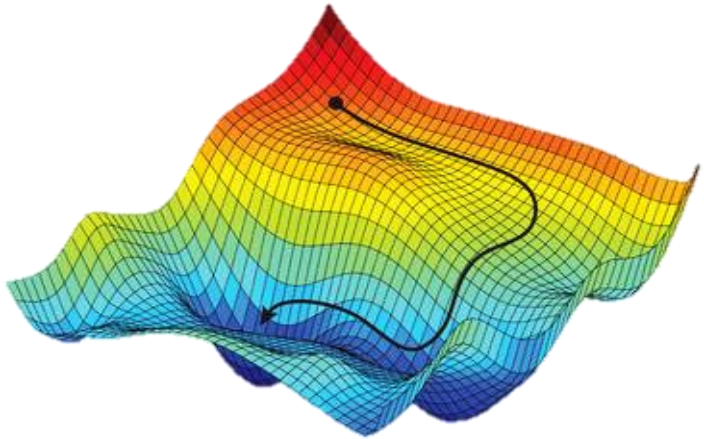
- forniamo dati di input $\mathbf{x}$ alla rete e calcoliamo gli output $\mathbf{a}(\mathbf{x}, \mathbf{w})$
- per **imparare**, ci serve quantificare *quanto simile* $\mathbf{a}(\mathbf{x}, \mathbf{w})$ è rispetto all'output atteso $\mathbf{y}(\mathbf{x})$
  - usiamo una funzione di errore (*loss function*)
- 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 di MLP (1967)

- il metodo del gradiente discendente (*Cauchy*, 1847) applicato ai MLP si basa su
  - inizializzazione *casuale* dei pesi $\mathbf{w}$
  - aggiornamenti successivi di $\mathbf{w}$ verso la discesa lungo la funzione di errore $\mathcal{L}$
  - **problema**
    - serve calcolare la **derivata** di $\mathcal{L}$ rispetto a ciascun parametro $w_1, w_2, \ldots, w_N$ della rete
    - intrattabile se la rete è complessa





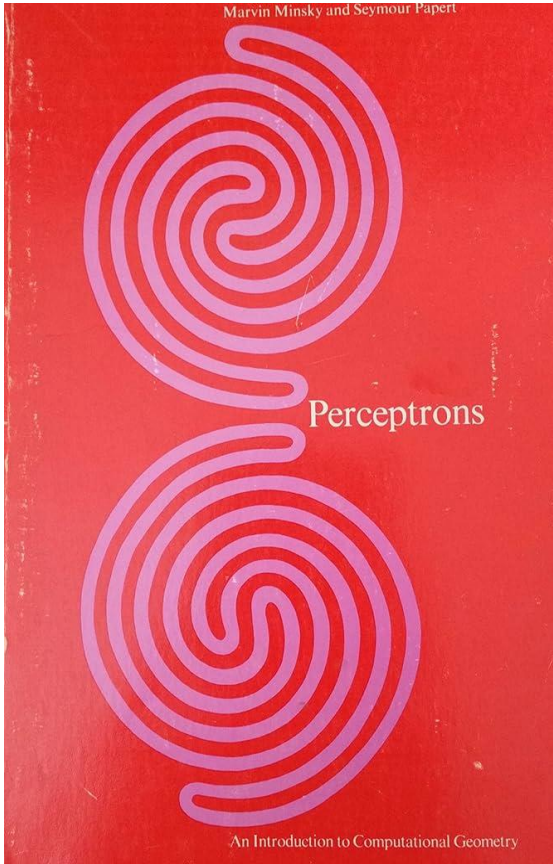
Crisi (1969-1980)

- 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

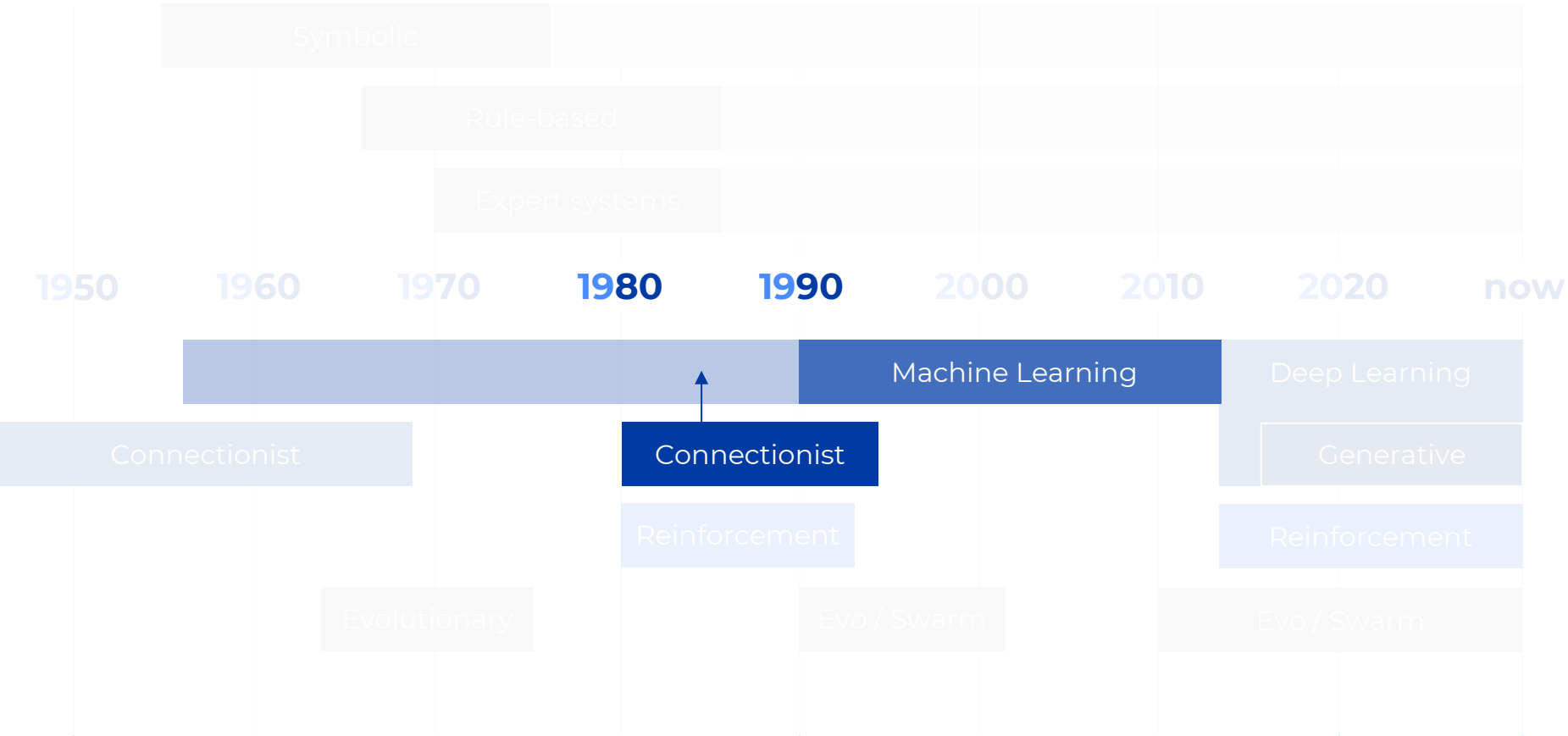
  - i ricercatori dell’epoca interpretarono il risultato come «*TUTTE le reti neurali sono fondamentalmente limitate*»

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



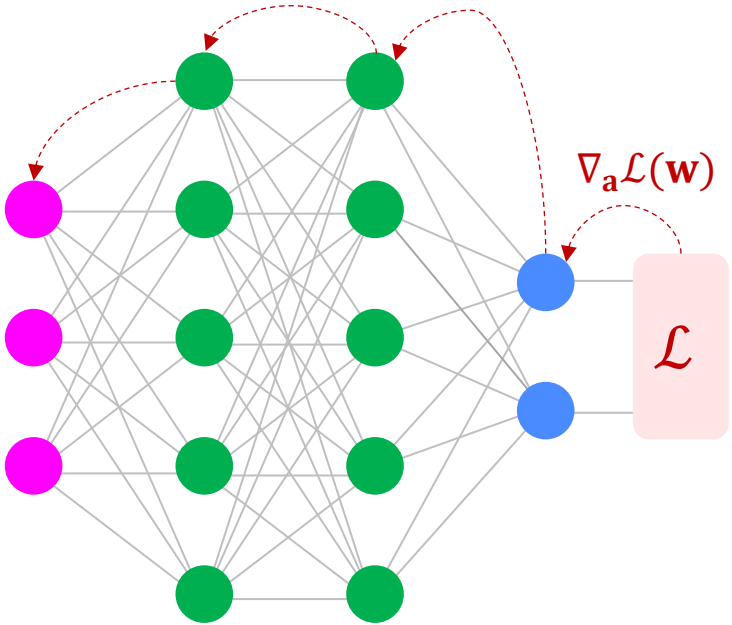
Dartmouth workshop (1956)

Connessionismo (1980-1993)

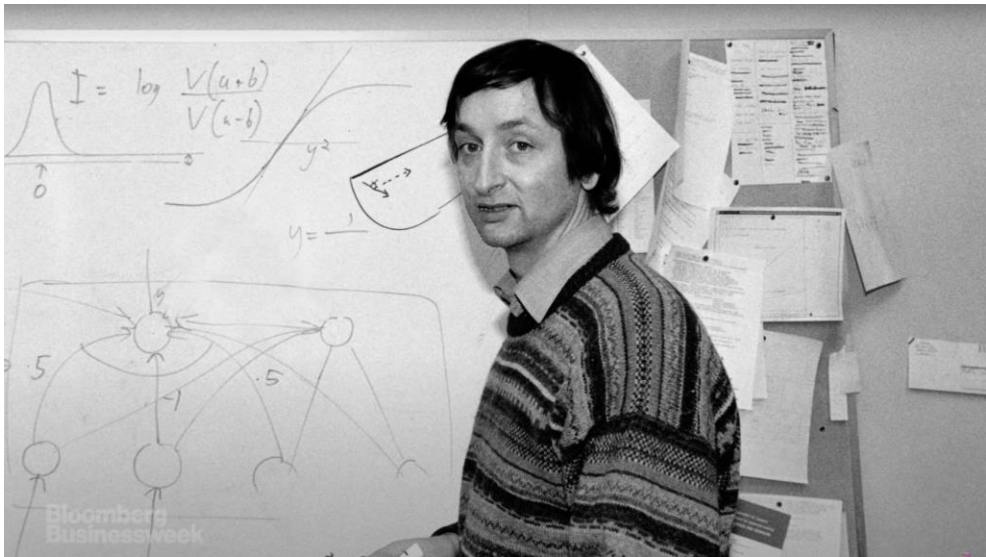


Backpropagation (1986)

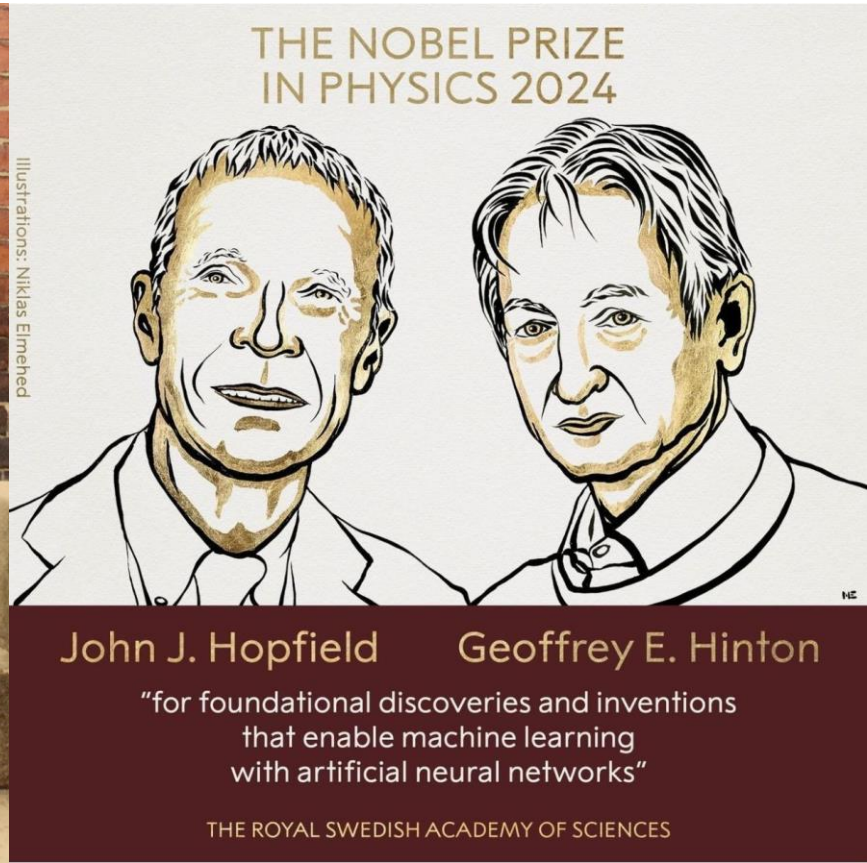
$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \left(\dots, \frac{\partial \mathcal{L}}{\partial w_j^{\ell}}, \dots \right)^T$$



- il tassello mancante
  - un algoritmo per calcolare le **derivate** della funzione di errore rispetto a ciascun parametro della rete
  - gradient descent + backpropagation



Nobel per la Fisica (2024)

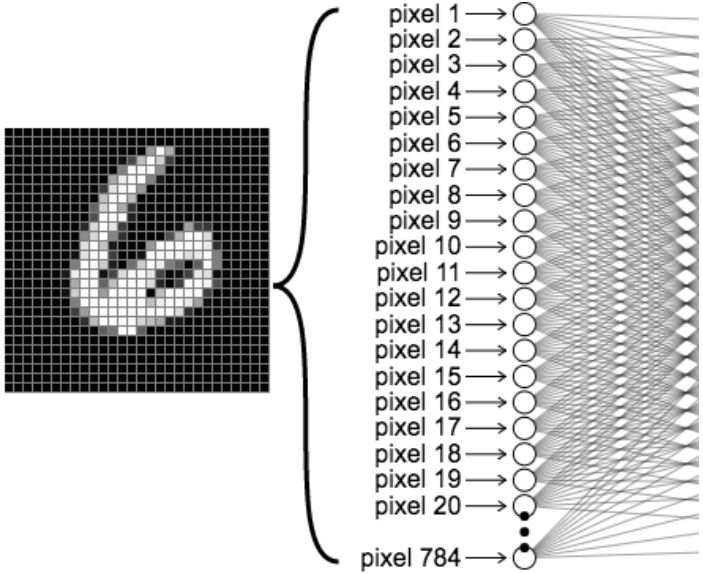


02

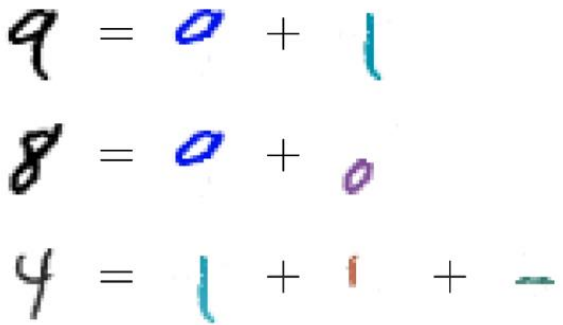
Reti convoluzionali

Filtri convoluzionali, apprendimento gerarchico, Generative Adversarial Networks (GAN)

Limiti delle reti MLP

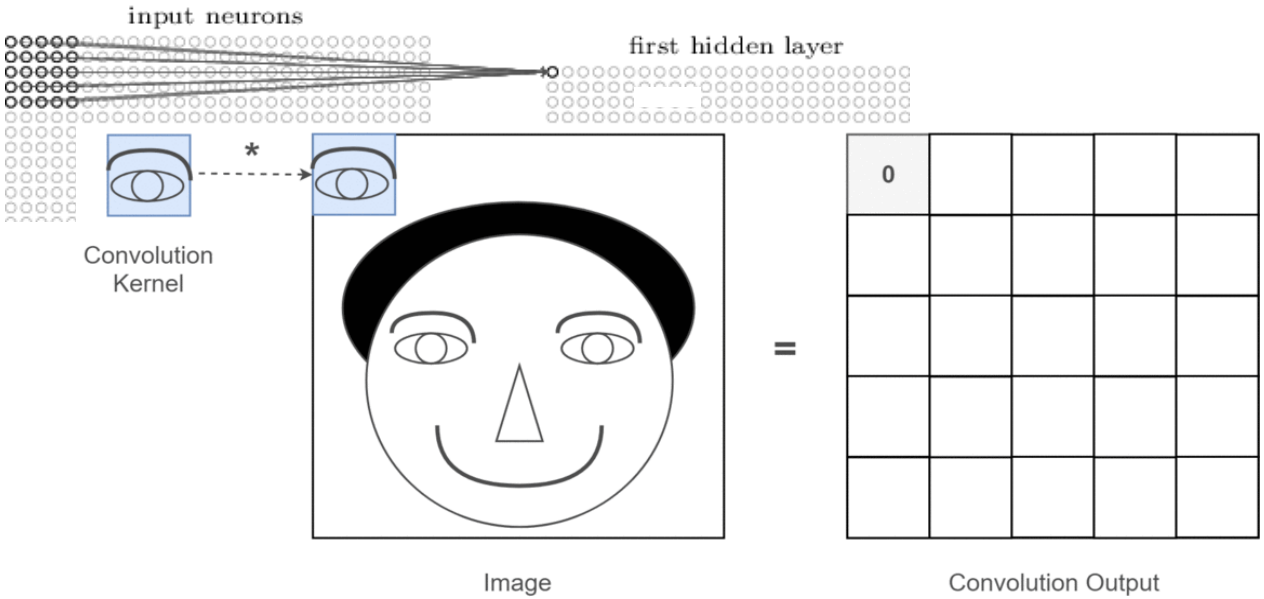


- ignorano la **struttura** del dato
- **scalano** terribilmente
- non apprendono **rappresentazioni gerarchiche**



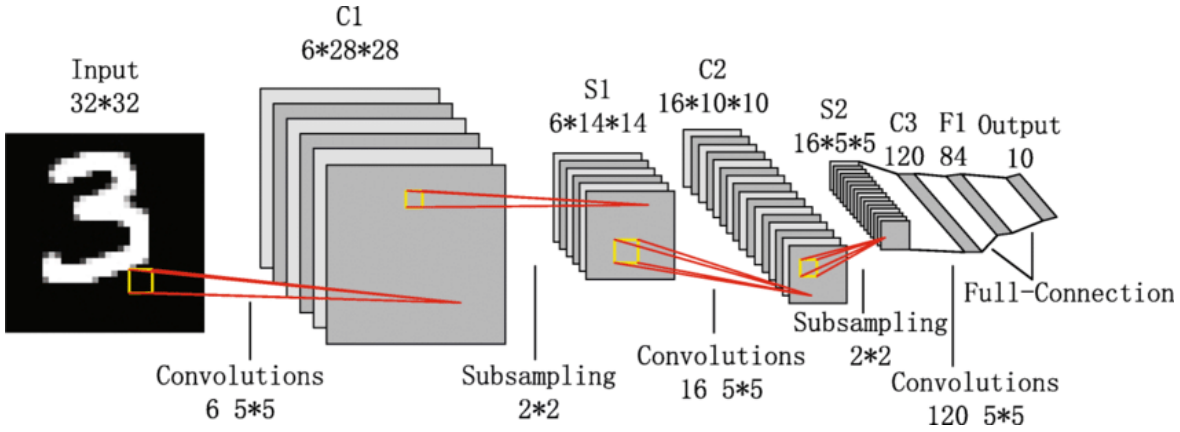
Reti Convoluzionali (CNN)

- mantengono 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**



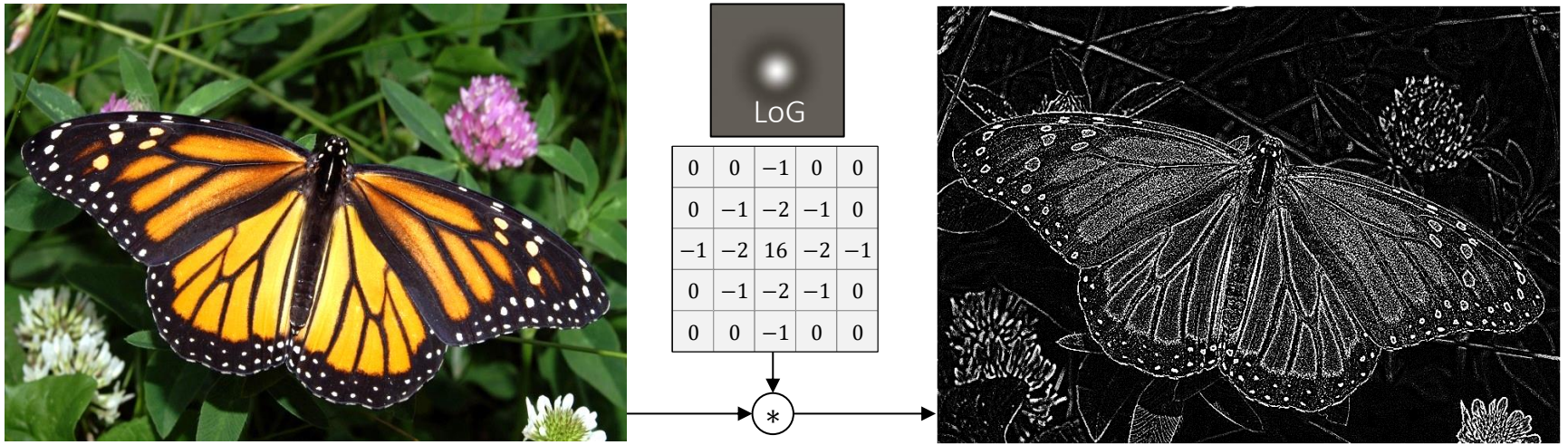
Reti Convoluzionali (1990-2000)

- sviluppate negli AT&T Bell Labs
  - riconoscimento di **immagini** (*Yann LeCun*, 1990)
  - riconoscimento del **parlato** (*Yoshua Bengio*, 1995)



Filtri convoluzionali

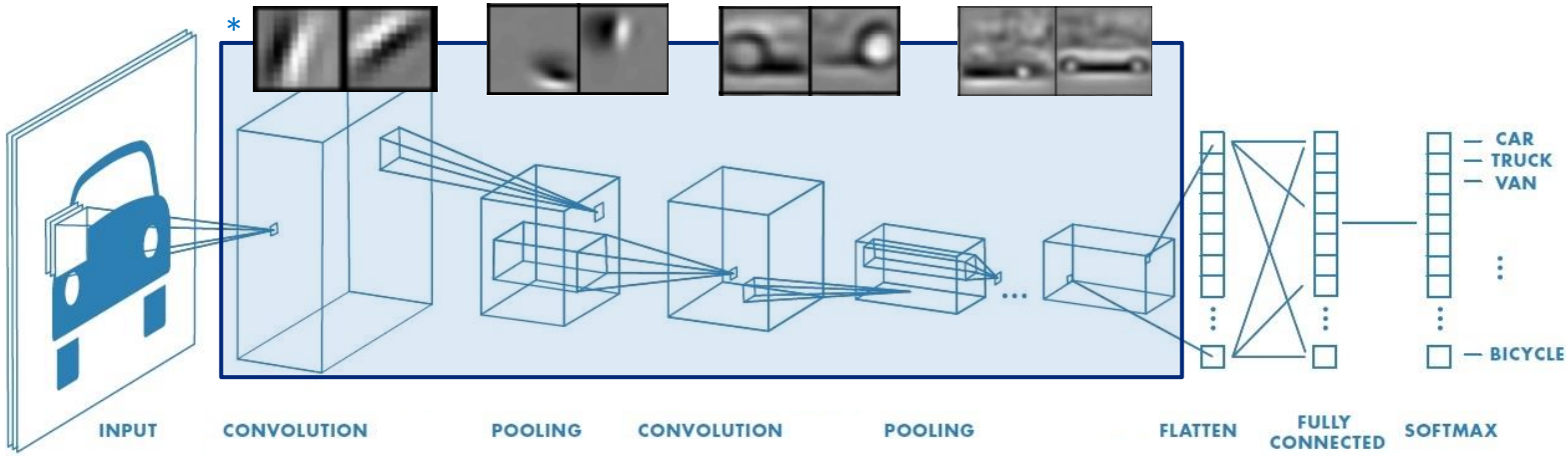
- imparano automaticamente quali misure caratteristiche estrarre
  - i **parametri** addestrabili sono i pesi W_{ij} del filtro, ovvero il filtro stesso

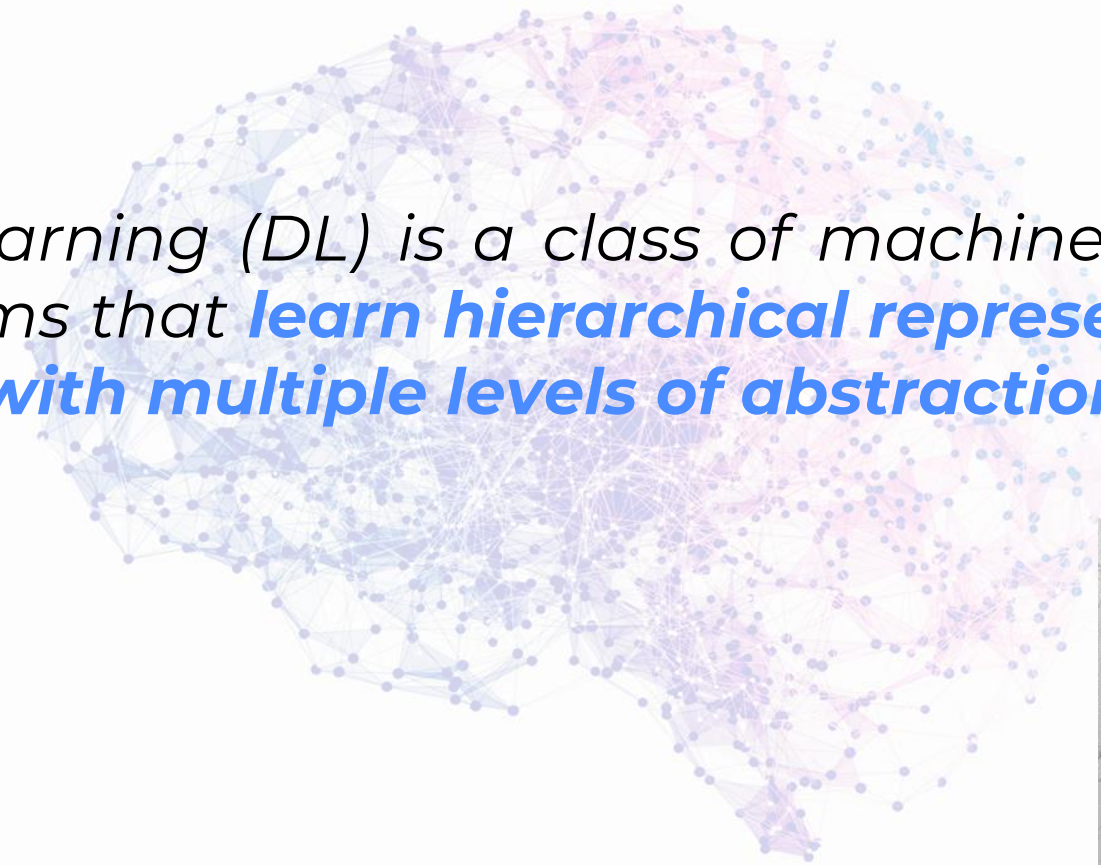


Questo filtro convoluzionale estrae i bordi, ovvero una misura di complessità della texture

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*)

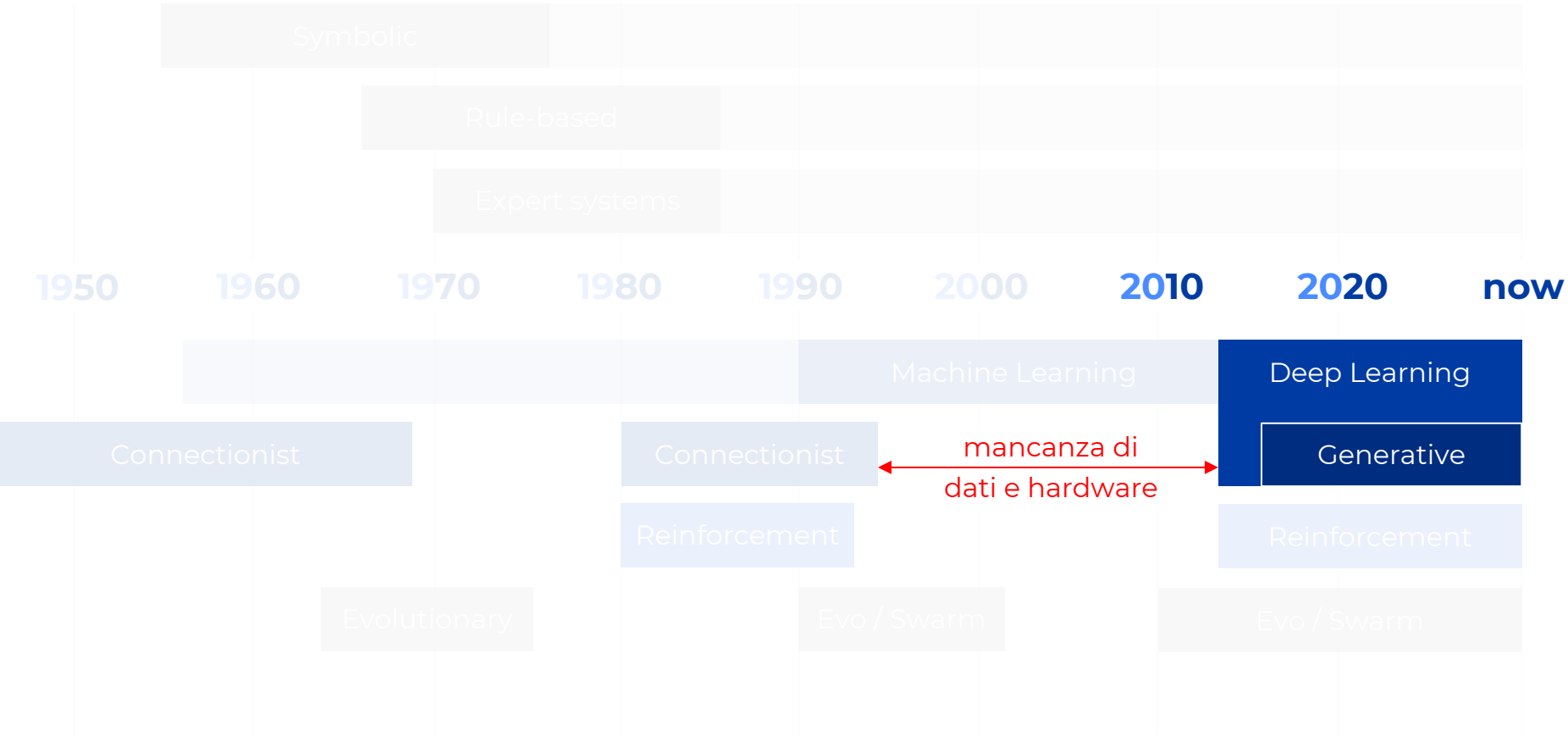




“Deep learning (DL) is a class of machine learning algorithms that ***learn hierarchical representations of data with multiple levels of abstraction****.”*

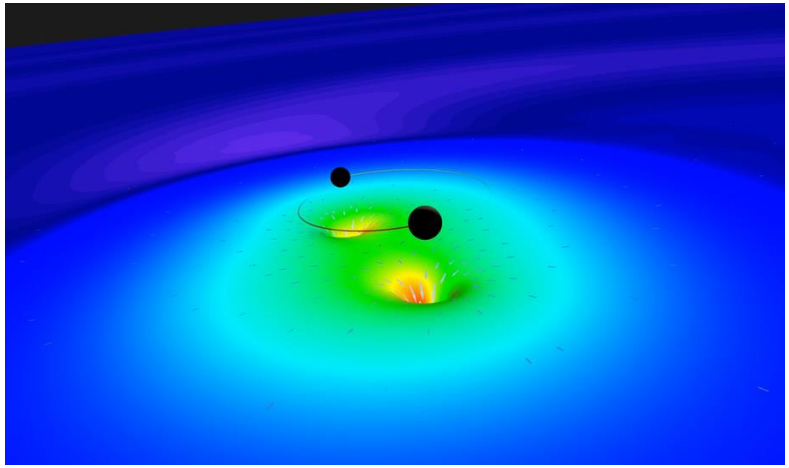
Rina Dechter, **1986**

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
- nVIDIA rilascia **CUDA** per general-purpose programming
- *Andrew Ng* (Stanford) ottiene 70 x speedup con 2 nVIDIA GeForce GTX280



NVIDIA®

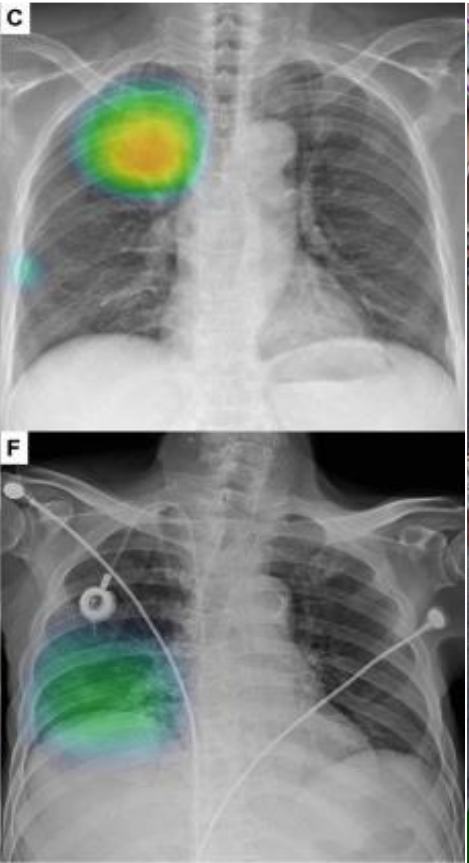
ImageNet (2009)

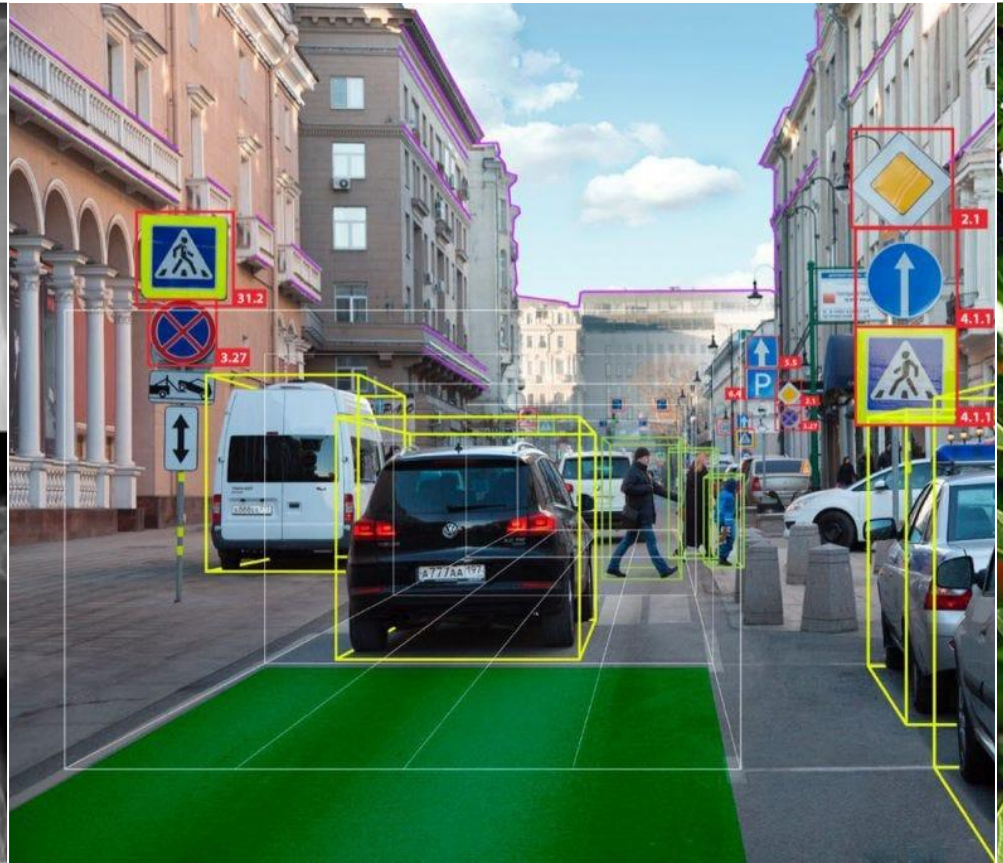
- **il più grande dataset di immagini mai creato**
  - 14.3M immagini
  - 21,841 categorie
  - realizzato in 3 anni grazie al progetto di *Fei Fei Li* (Stanford)

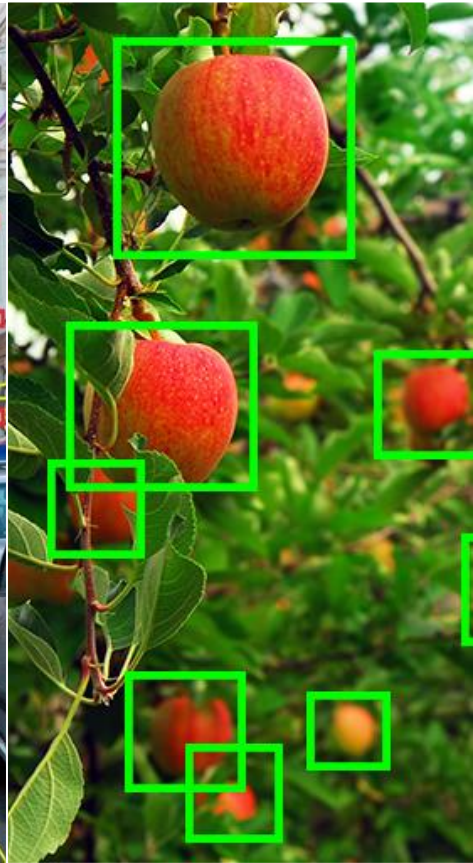
ILSVRC e AlexNet (2012)

- **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*

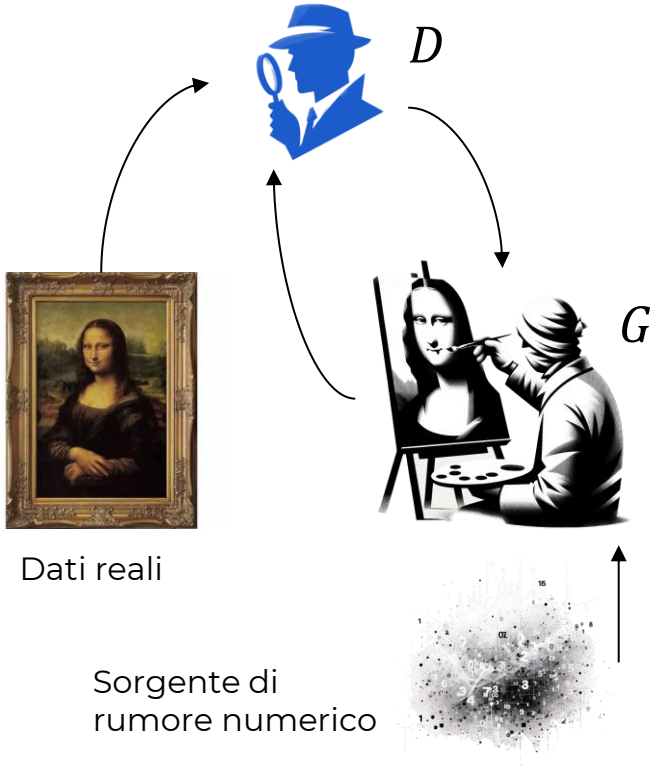
Visione artificiale (2012-now)







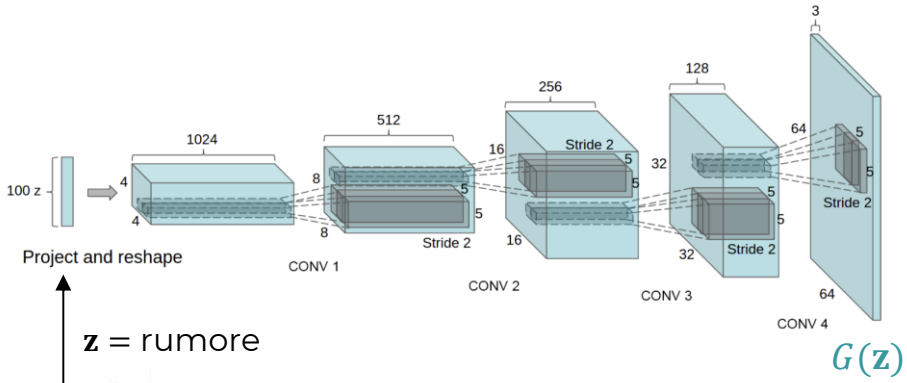
Reti generative avversariali (GAN, 2014)



- G è il **generatore**
  - una CNN che crea dati finti (*fake*), all'inizio a caso ma poi più verosimili
    - ha accesso al giudizio del discriminatore, ed è addestrata per ingannarlo
  - l'input è una sorgente di **rumore** che gioca il ruolo della *creatività*
- D è il **discriminatore**
  - una CNN che classifica un input come *genuino* o *fake*
    - ha accesso ai dati veri, ed è addestrata per riconoscere i dati veri come veri e i dati generati da G come finti

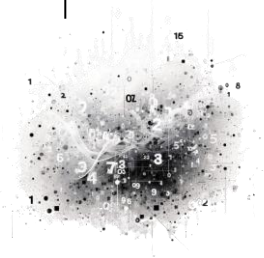
Architettura GAN

Generatore

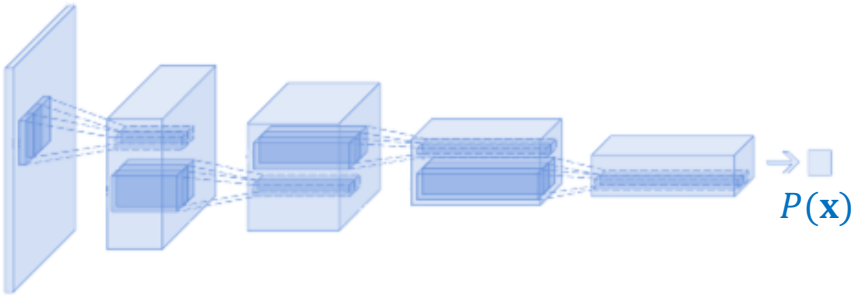


$G(\mathbf{z})$

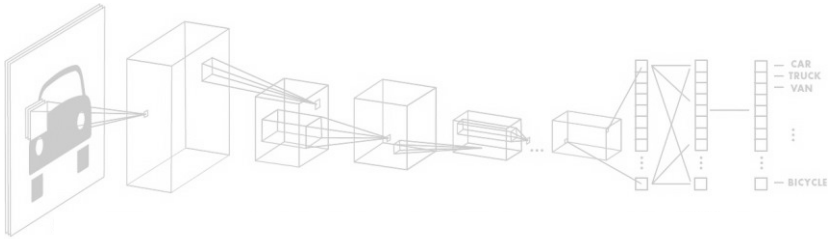
$\mathbf{z}$ = rumore



Discriminatore

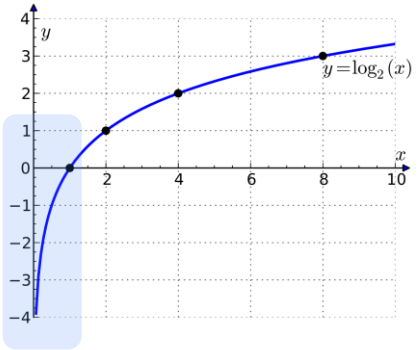


$P(\mathbf{x})$



Funzione di errore GAN

- siano
  - $\mathbf{x}$ i dati reali (ad es., volti umani)
  - $G(\mathbf{z})$ i volti finti generati da G con una CNN che prende in input rumore casuale $\mathbf{z}$
  - $D(\cdot)$ una CNN che mappa un volto nella probabilità $\in [0,1]$ che esso sia vero
- obiettivo di D:
  - massimizzare la probabilità $D(\mathbf{x})$ = probabilità di volti veri
  - minimizzare la probabilità $D(G(\mathbf{z}))$ = probabilità di volti finti
    - equivalente a massimizzare la probabilità $1 - D(G(\mathbf{z}))$
- obiettivo di G:
  - minimizzare quello che D vuole massimizzare = ingannare D



- data una probabilità P da massimizzare, meglio massimizzare $\log P$
  - l'andamento asintitotico del logaritmo vicino lo 0 penalizzerà le basse probabilità

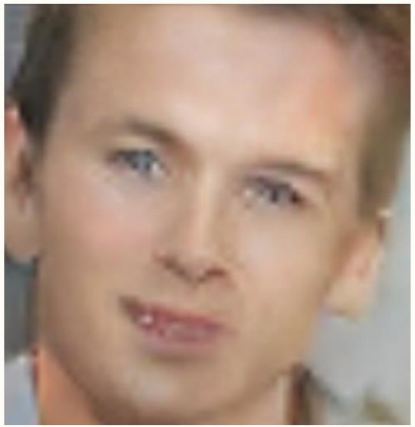
Funzione di errore GAN

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z})))].$$

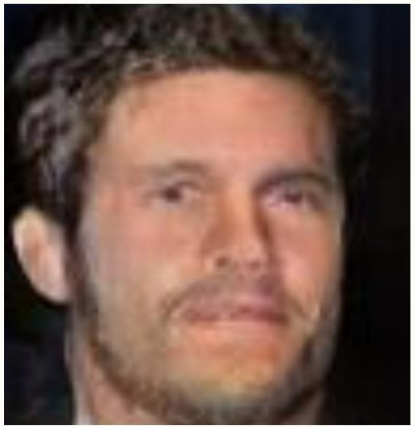
- è la firma del primo «artefatto» mai generato da un'AI, battuto all'asta per 432.500$
  - *Ritratto di Edmond Belamy* (**2018**)
- è una battaglia tra due avversari che giocano un ***minimax game***
  - metodo per minimizzare la massima perdita possibile usato nella classica intelligenza artificiale (ad es. giocatore di scacchi)
  - il valore atteso (*expected value*) $\mathbb{E}$ è la media su tutti i campioni del dataset considerato

Progressi GAN

2014



2015



2016

2017

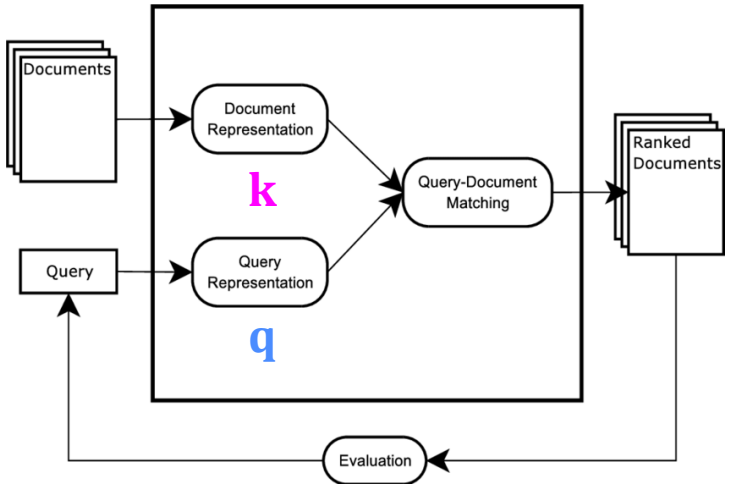
2020 - https://thispersondoesnotexist.com

03
Reti attenzionali

Meccanismi attenzionali,
Transformer, LLM

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 $\mathbf{k}$

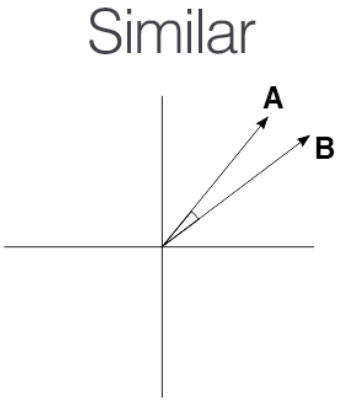


Google images

Prodotto scalare

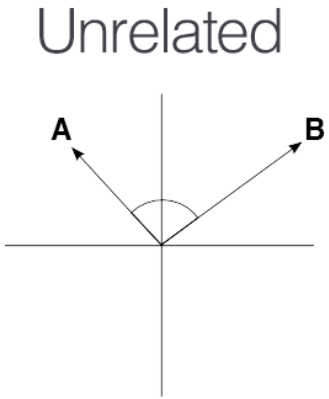
- il **prodotto scalare** $(\cdot)$ tra il vettore $\mathbf{q} = (q_1 \ \ldots q_d)^T$ e il vettore $\mathbf{k} = (k_1 \ \ldots k_d)^T$ è una misura di ***similarità***

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



- 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

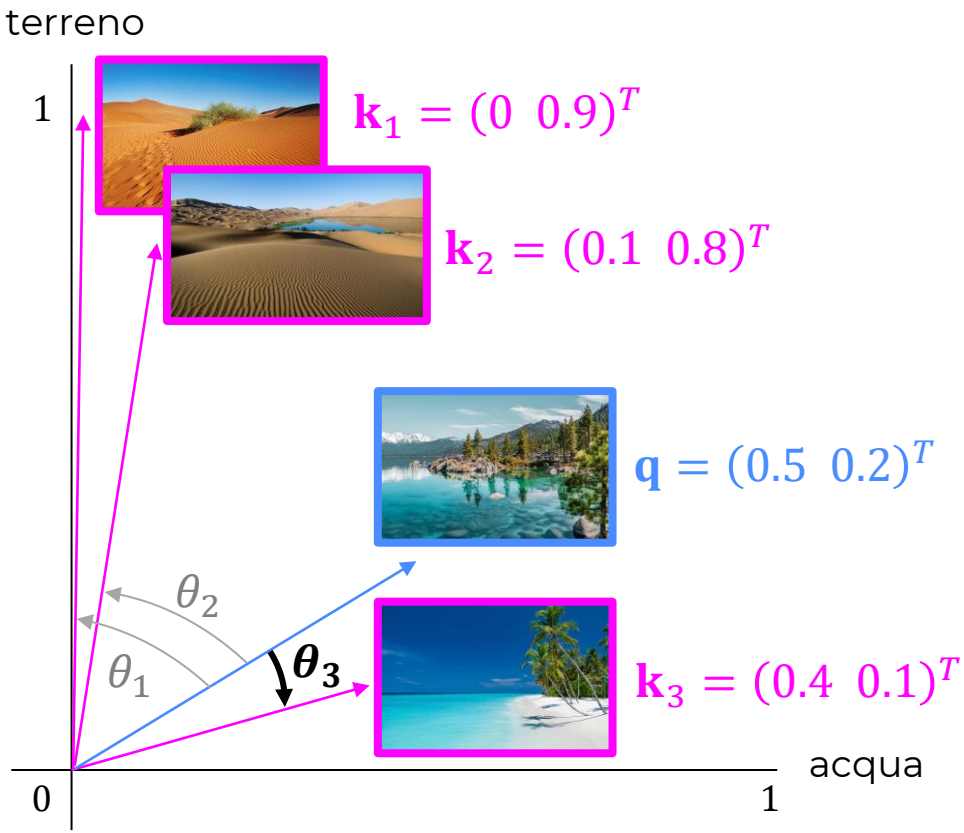
Esempio giocattolo:
Motore di ricerca di immagini
con dimensione dello spazio

$d = 2$

Esempio reale:

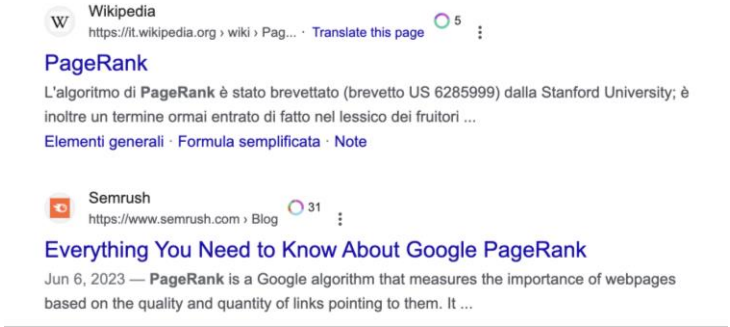
Google images

$d \gg 2$



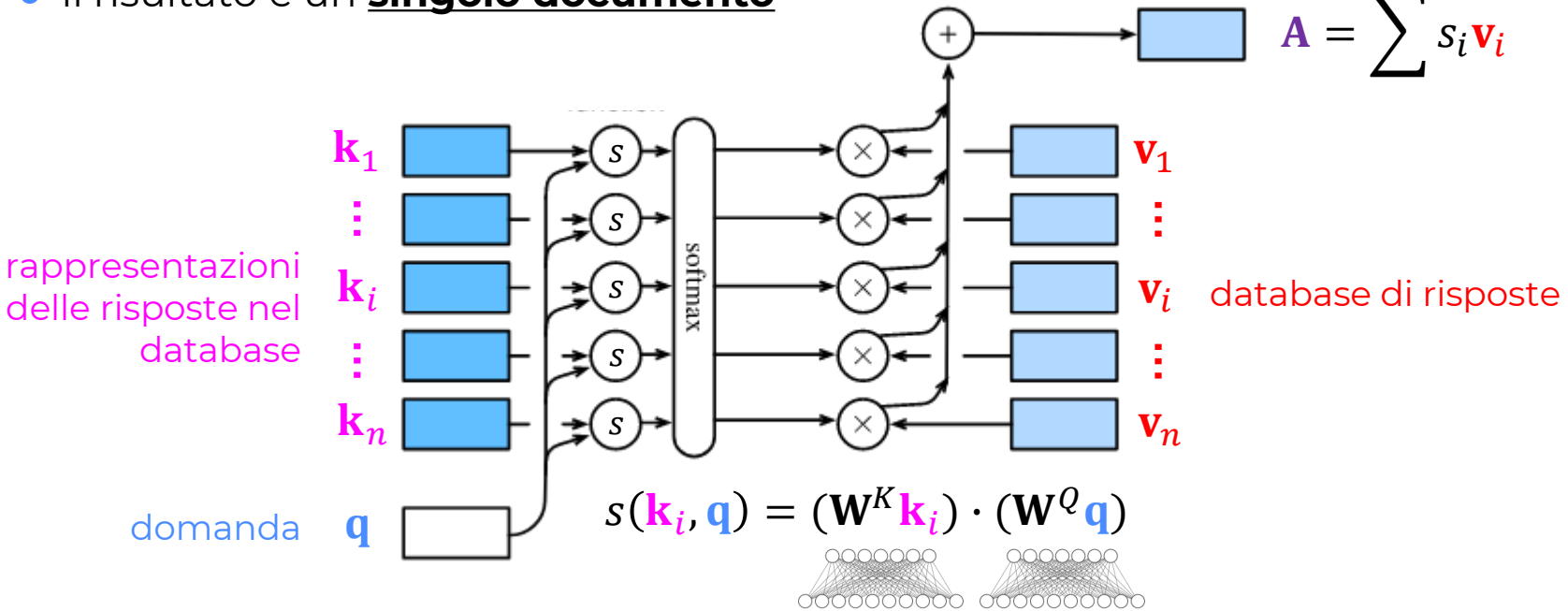
Google?

- $\mathbf{q}$ 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

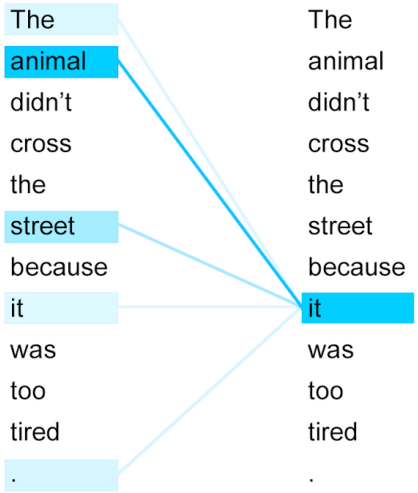
- il risultato è un **<u>singolo documento</u>**

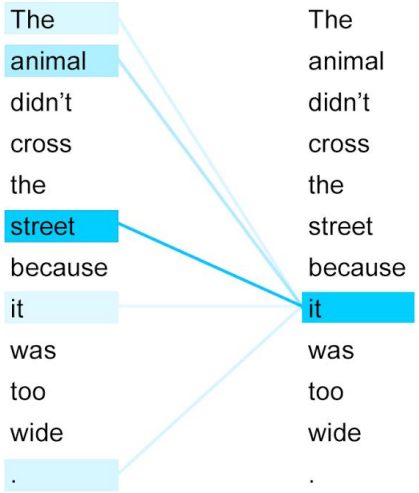


$\mathbf{W}^K$ e $\mathbf{W}^Q$ sono **trasformazioni** implementate mediante piccole **reti neurali** (2 layer) che mappano $\mathbf{k}_i$ e $\mathbf{q}$ in uno spazio dove sono meglio confrontabili

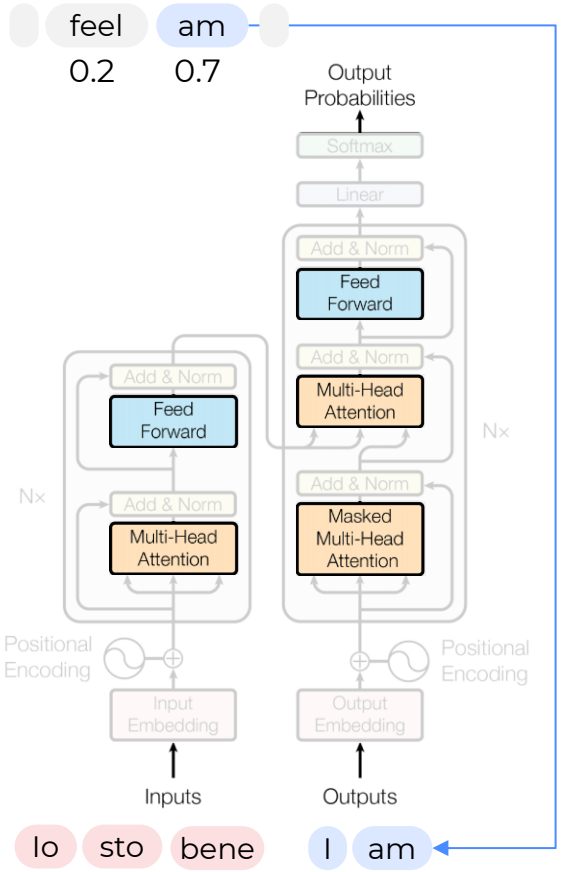
Self-attenzione

- **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!

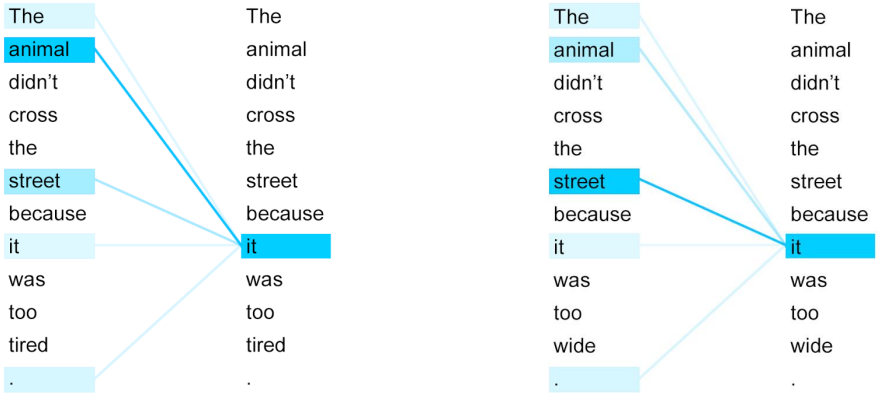




Transformer (2017)

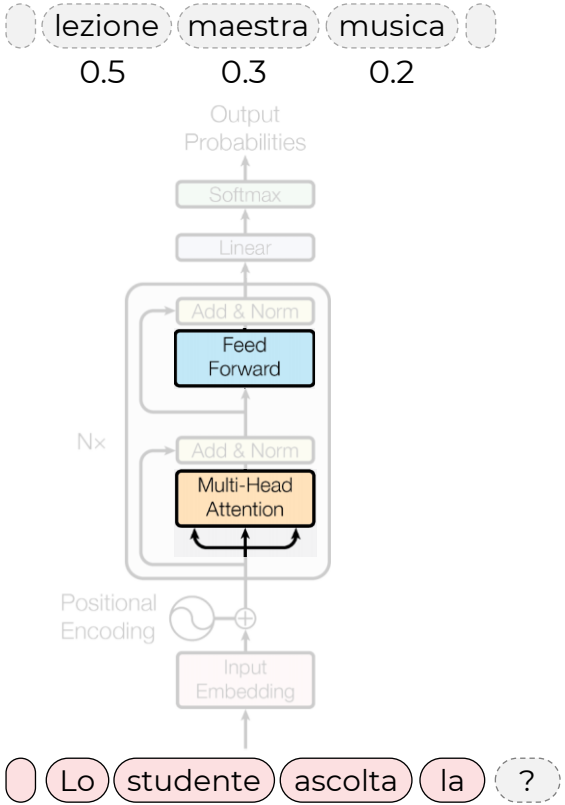


- progettato da Google per la **traduzione automatica**
  - basata sulla self-attenzione e la cross-attenzione
    - impara le correlazioni intrinseche in una frase e tra le due lingue per comprendere meglio la semantica



  - autoregressiva
    - man mano che la rete genera le parole nella nuova lingua, queste diventano l'input per predire la prossima

GPT (2018-now)



- sfrutta una parte del Transformer per **predire la prossima parola** nella frase
  - modelli sempre più grandi (**LLM**) addestrati su quantità di dati sempre più grandi ($> 100\ TB$ di testo)

OpenAI

04
Modelli di diffusione

Diffusione inversa,
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»)

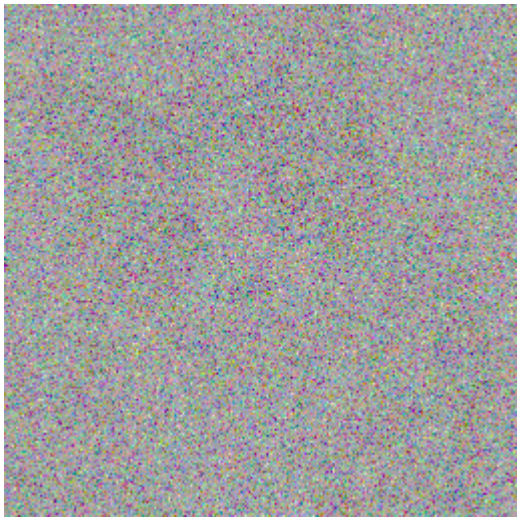
t

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)

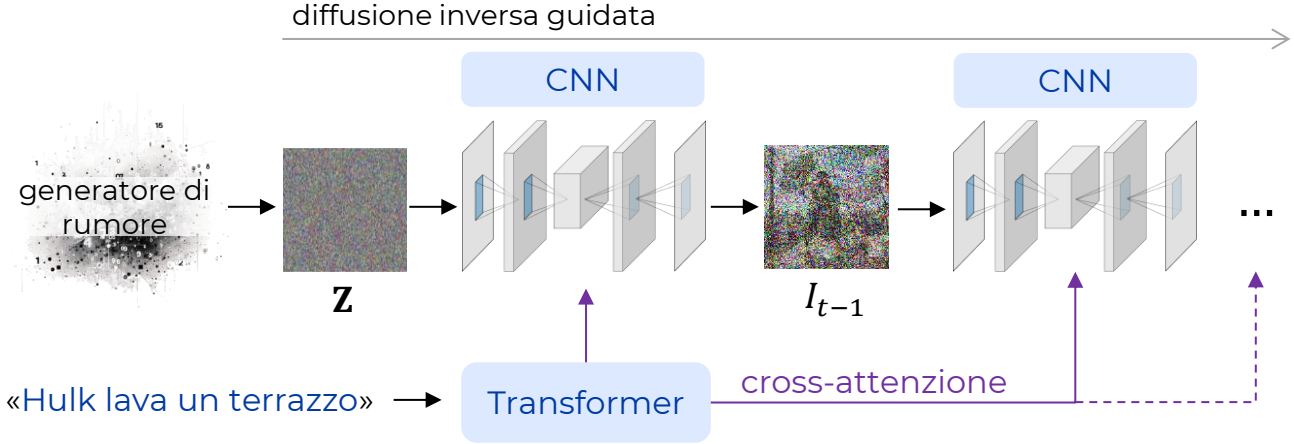
Generazione di immagini

- 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

- 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



Diffusion-based LLM?

Iterations

0

AUTOREGRESSIVE LLM

LEFT-TO-RIGHT GENERATION

Iterations

0

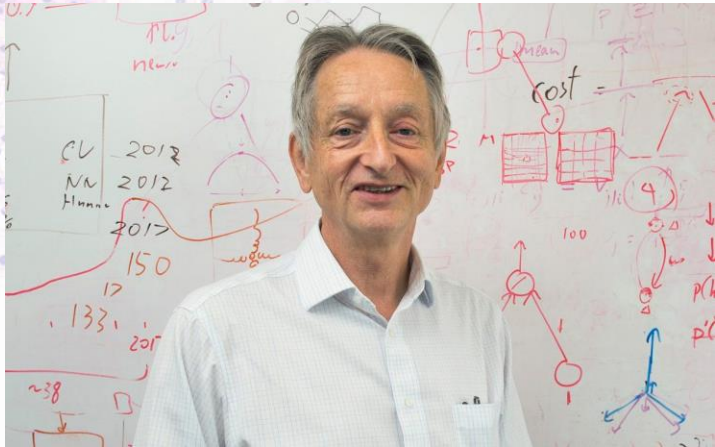
INCEPTION DIFFUSION LLM

COARSE-TO-FINE GENERATION

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***.”

ChatGPT4o, **2025**





ABSOUTELY
MAIMUM TOPE



*“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**