

AI Literacy

Introduzione e fondamenti di Intelligenza Artificiale

Lorenzo Babini



Sommario

- INTRO

Definizioni e storia / Approccio simbolico e connessionista / Ambiti / Reti neurali artificiali / Percettrone semplice

- DEEP LEARNING

Basi di machine learning / Regole di apprendimento / Reti neurali multistrato / Tipi di apprendimento

- WORD EMBEDDING

Esempio di Word2Vec / Word Embeddings come rappresentazioni / Language models

- RETI NEURALI FONDAMENTALI

Convoluzionali / Ricorrenti / Long Short Term Memory / Transformer e Large Language Models

- BIBLIOGRAFIA

Cos'è l'intelligenza artificiale?

“ Il problema dell'IA è quello di costruire una macchina che agisca in modo che si definirebbe intelligente se un umano si comportasse allo stesso modo ”

J. MCCARTHY 1956

“ L'IA è quella disciplina, appartenente all'informatica, che studia i fondamenti teorici, le metodologie e le tecniche che permettono di progettare sistemi digitali (*hardware*) e sistemi di programmi (*software*) capaci di fornire all'elaboratore elettronico prestazioni che, a un osservatore comune, sembrerebbero di pertinenza esclusiva dell'intelligenza umana. ”

M. SOMALVICO 1987

“ La IA è la delucidazione del processo umano di apprendimento, la quantificazione del processo del pensiero umano, l'esplicazione del comportamento umano e la comprensione di ciò che rende possibile l'intelligenza. È il passo finale dell'umanità lungo il cammino della comprensione di se stessa. ”

KAI-FU LEE 1988

“ L'IA è una disciplina con l'obiettivo non solo di comprendere, ma anche di costruire entità intelligenti, cioè macchine in grado di calcolare come agire in modo efficace e sicuro in un'ampia varietà di situazioni nuove. ”

S. RUSSEL, P. NORVING 2021

“ IA designa innanzitutto un oggetto che si cerca di creare (un sistema dotato di una certa proprietà), ma designa anche la disciplina e, in senso più ampio, l'istituzione che s'incarica di concepire tale oggetto. [...] La confusione che circonda il termine è nota, al punto che alcuni consigliano, in modo puro e semplice, di abbandonarlo. ”

D. ANDLER 2023

Le macchine possono pensare?



ALAN TURING (1912-1954)

«Il solo modo per cui si potrebbe essere sicuri che una macchina pensa è quello di essere la macchina e di sentire sé stessi pensare. [...] Secondo questa opinione la sola via per sapere se un uomo pensa è quella di essere quell'uomo in particolare. [...] Invece di discutere in continuazione su questo punto, è normale attenersi all'educata convenzione che ognuno pensi»

TEST DI TURING (1950)



JOHN SEARLE (1932)

IA forte = le macchine possono agire pensando in modo cosciente e non limitandosi a simularlo
IA debole = le macchine possono agire *come* se fossero intelligenti, ma non lo sono

STANZA CINESE (1990)

EDSGER DIJKSTRA (1930-2002)

«Chiedersi se le macchine possano pensare è importante più o meno quanto chiedersi se i sottomarini possano nuotare» (1984)

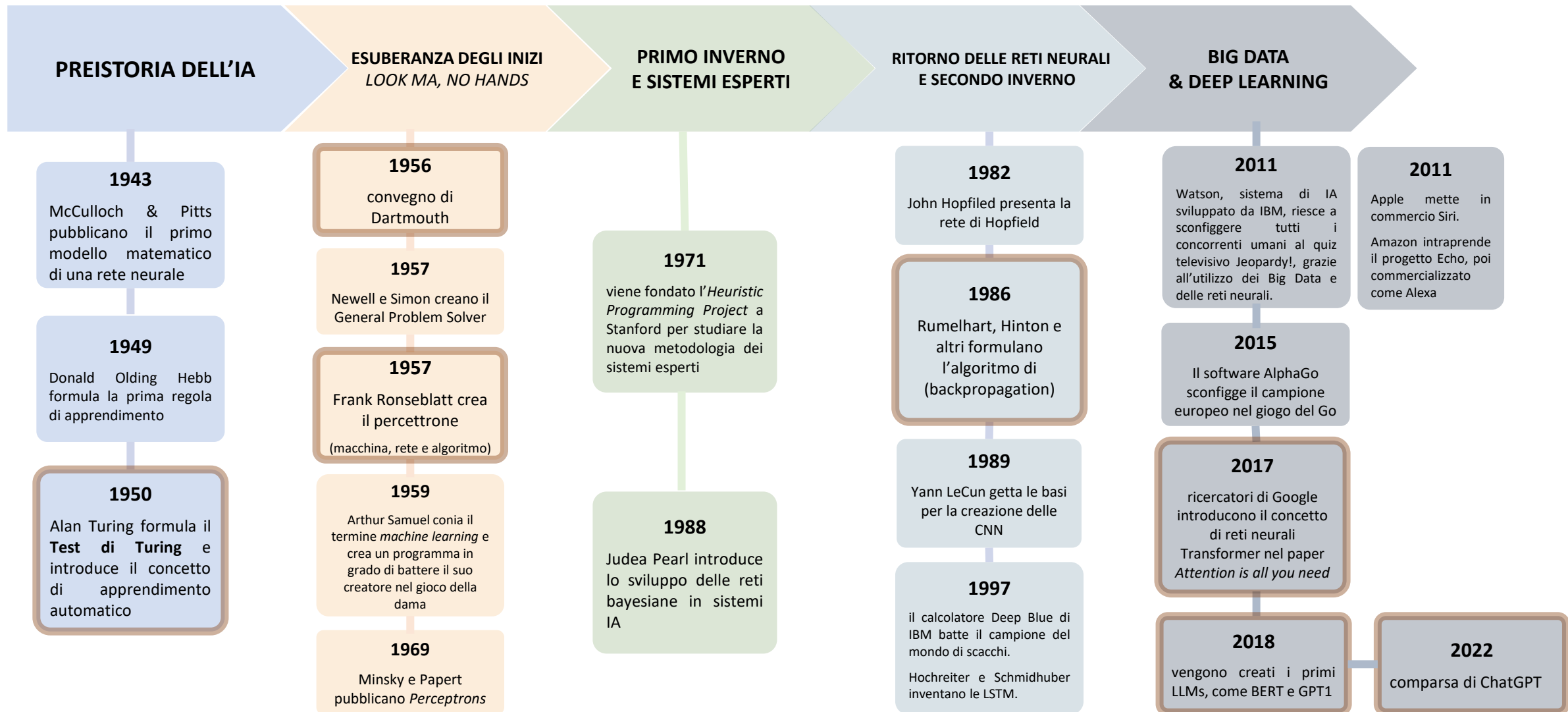
YOSHUA BENGIO (1964-)

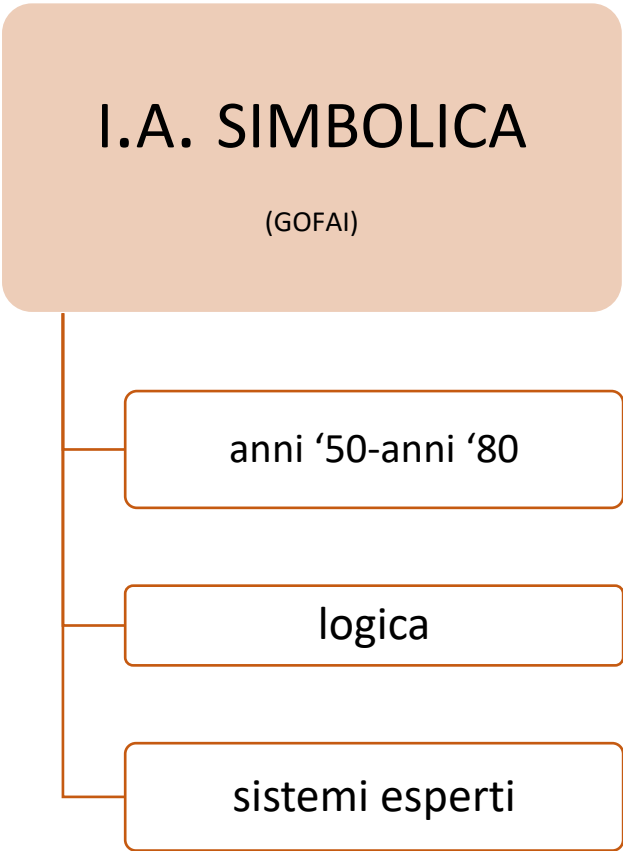
«Le nostre macchine sono ottuse, e noi non facciamo che cercare di renderle meno ottuse» (2016)

LLMs

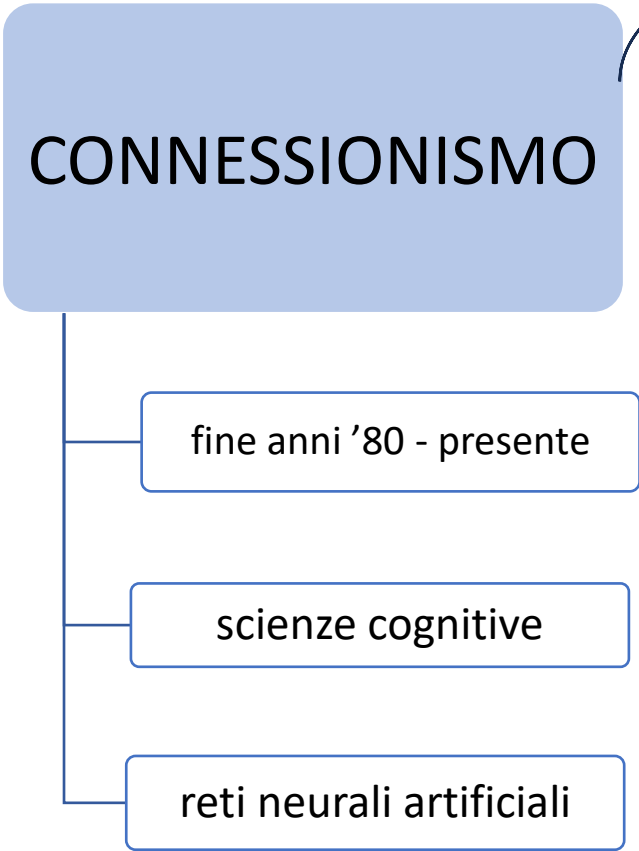
- D. J. CHALMERS, [*Could a Large Language Model be Conscious?*](#), "Boston Review", 2023, in arXiv:2303.07103.
- M. BENASAYAG, A. PENNISI, *ChatGPT non pensa (e il cervello neppure)*, Jaca Book, Milano 2024.

Storia dell'IA - sintesi





Programmi basati su rappresentazioni e linguaggi simbolici, regole e relazioni logiche per risolvere determinati problemi. Sono sempre in grado di esplicitare le regole logiche che hanno portato alla soluzione. Sono formati da una base di conoscenze (*knowledge base*); un motore inferenziale; un'interfaccia utente.



Formano concetti interni in modo più fluido e impreciso, più adatto alla confusione del mondo reale, rispetto ai pacchetti di istruzioni confezionate da un essere umano.



DEEP LEARNING

- costruzione di reti neurali profonde
- incremento della potenza di calcolo dei computer
- disponibilità di Big Data

"Abbiamo fatto una sola grande scoperta in oltre sessant'anni"

KAI-FU LEE, CHEN QIUFAN, *AI 2041. Scenari dal futuro dell'intelligenza artificiale*, Luiss University Press, Roma 2023.



Yann Le Cun, Geoffrey Hinton, Yoshua Bengio
Turing Award 2018

Principali campi dell'IA



Machine Learning



Computer Vision



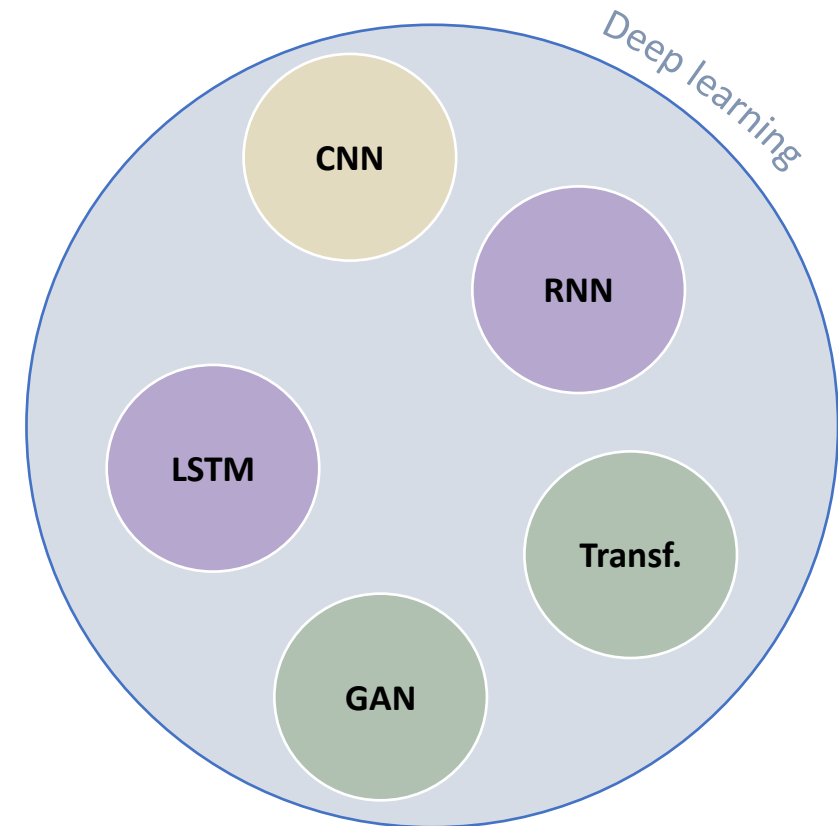
Natural Language Processing



Generative IA e AGI



Robotica e IA integrata



Reti neurali artificiali

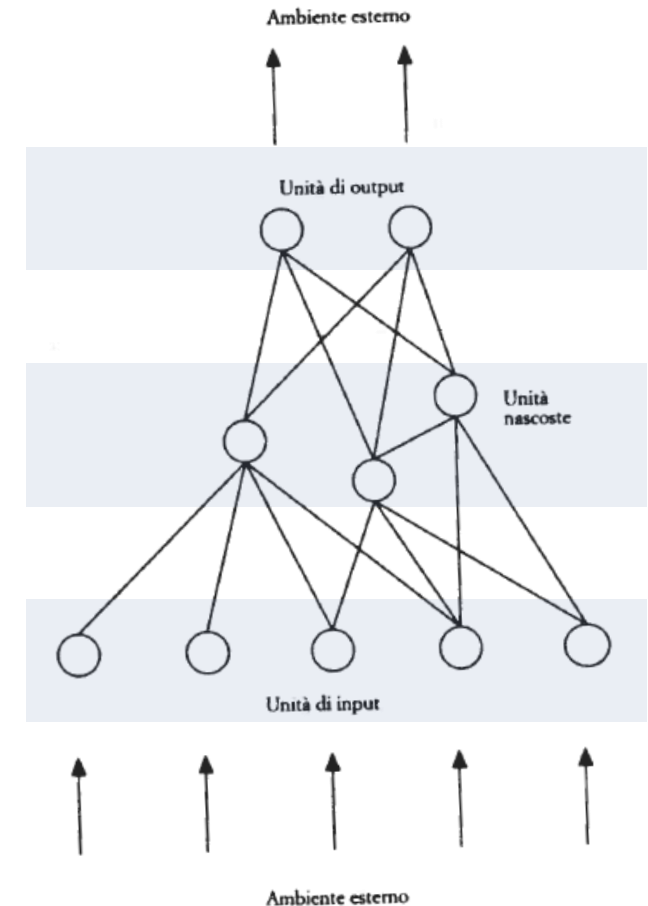
Sono dei modelli computazionali, ispirati alle più complesse reti neurali dei sistemi nervosi biologici, nati nell'ambito delle neuroscienze e della psicologia cognitiva.

I neuroni sono unità di calcolo collegate tra loro mediante connessioni. Ogni unità di calcolo che viene attivata, compie un calcolo e trasmette il risultato alle unità a cui è connessa.

Il successo dei modelli a rete neurale è dovuto a:

- autoapprendimento
- generalizzazione
- robustezza
- flessibilità

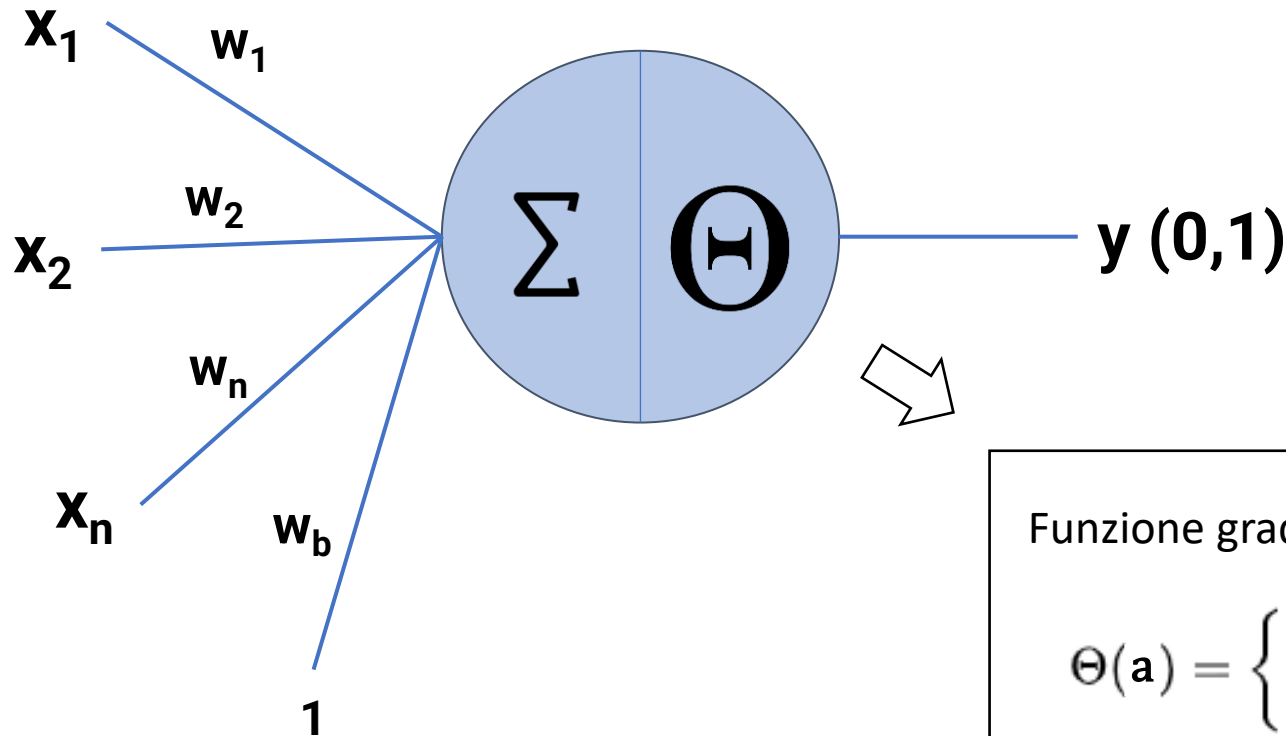
Criticità (funzionamento a *black box*, natura probabilistica degli output, quantità di dati)



Percettrone semplice

(Rosenblatt 1958; Minsky-Papert 1969)

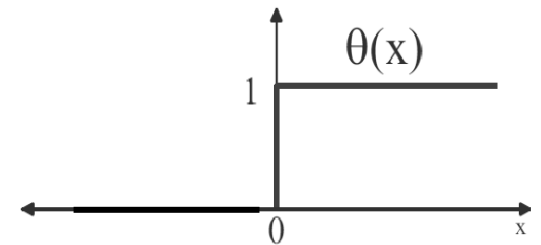
- l'unità minima delle prime reti neurali
- un **algoritmo di classificazione binaria**, che trasforma valori di input in un valore binario.



$$y = \Theta \sum_{i=1}^n (x_i w_i)$$

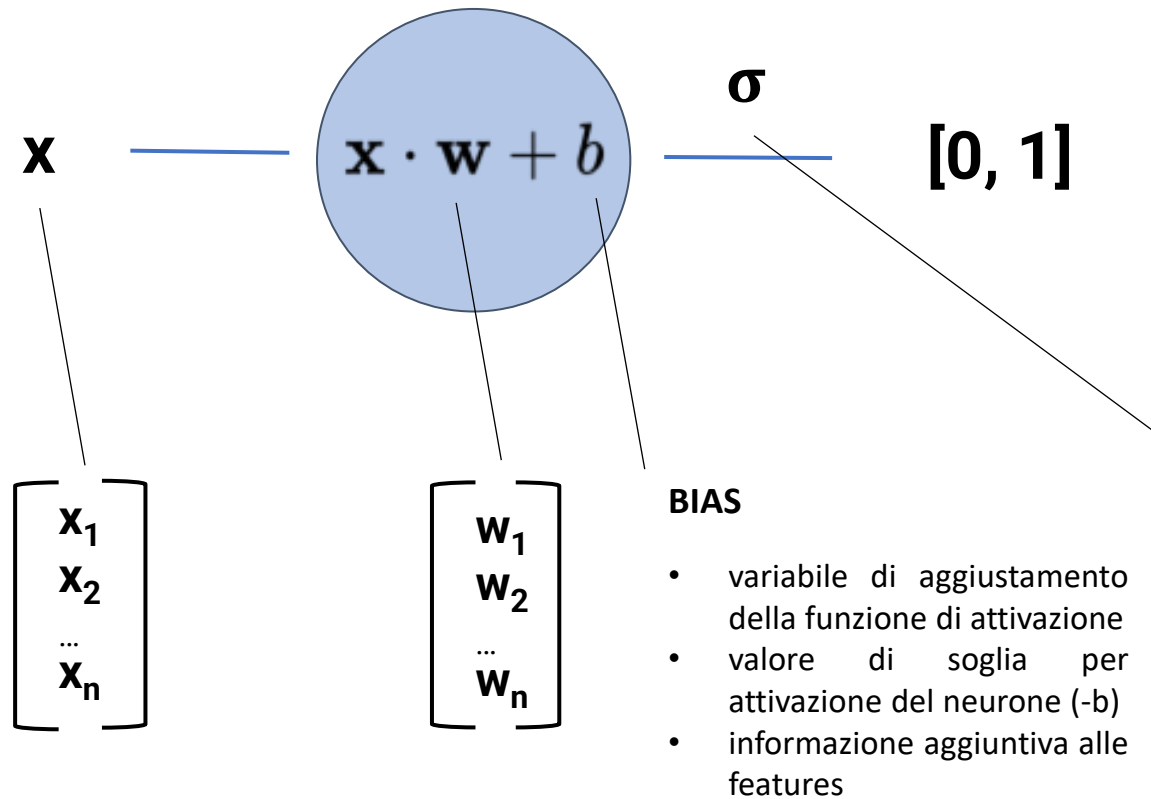
Funzione gradino di Heaviside

$$\Theta(a) = \begin{cases} 0, & a < 0 \\ 1, & a \geq 0 \end{cases}$$



Rete neurale: singolo neurone

Modello di rete neurale artificiale basato su un solo strato e su un **singolo neurone**:



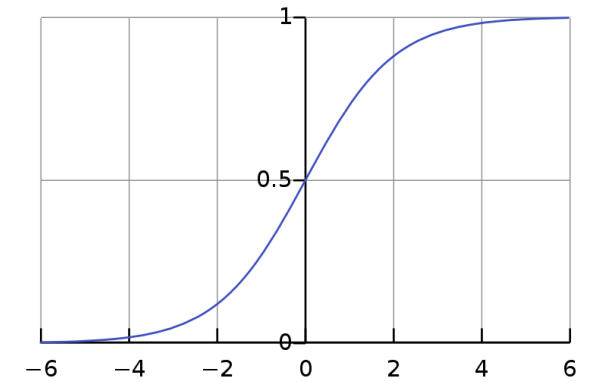
$$f(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

SIGMOID FUNCTION

Funzione che compatta un valore $y \in (-\infty, +\infty)$ in un intervallo da 0 a 1

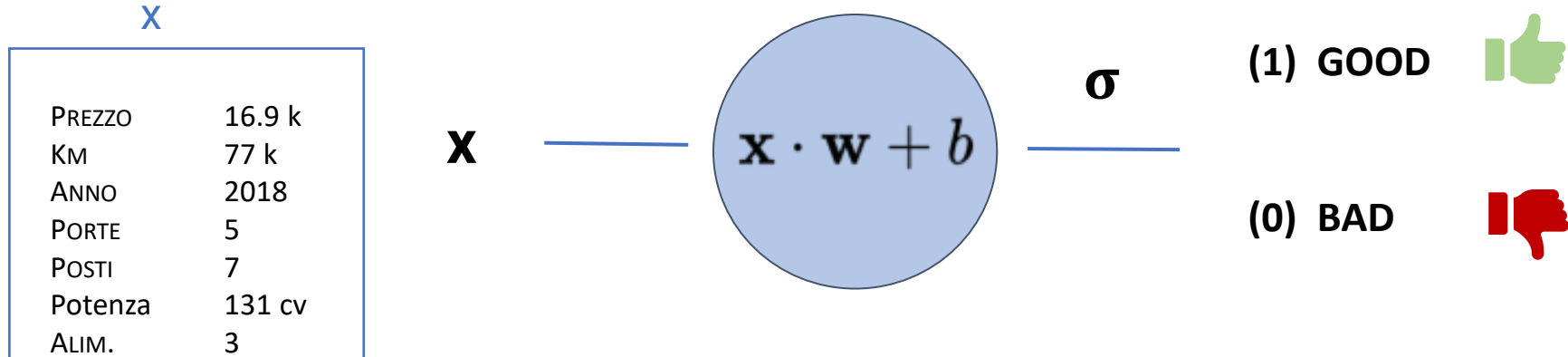
$$\sigma(y) \in (0, 1)$$

$$\sigma(y) = \frac{1}{1 + e^{-y}}$$



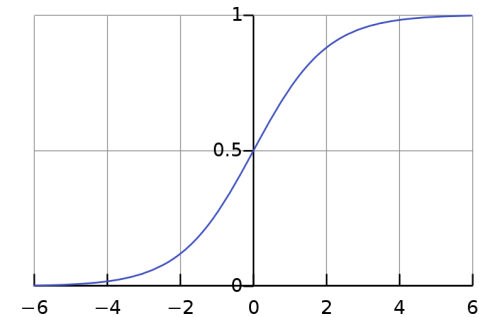
Esempio


Rete neurale già addestrata a riconoscere la convenienza di acquisto di una macchina usata



$$\begin{bmatrix} 16.9 & 77 & 18 & 5 & 7 & 131 & 3 \end{bmatrix} \cdot \begin{bmatrix} -2.6 \\ -2 \\ 3 \\ 3 \\ 3 \\ 0.9 \\ -6 \end{bmatrix} + \begin{bmatrix} 8 \end{bmatrix} = -0.04$$

[sigmoid-function-calculator](#)

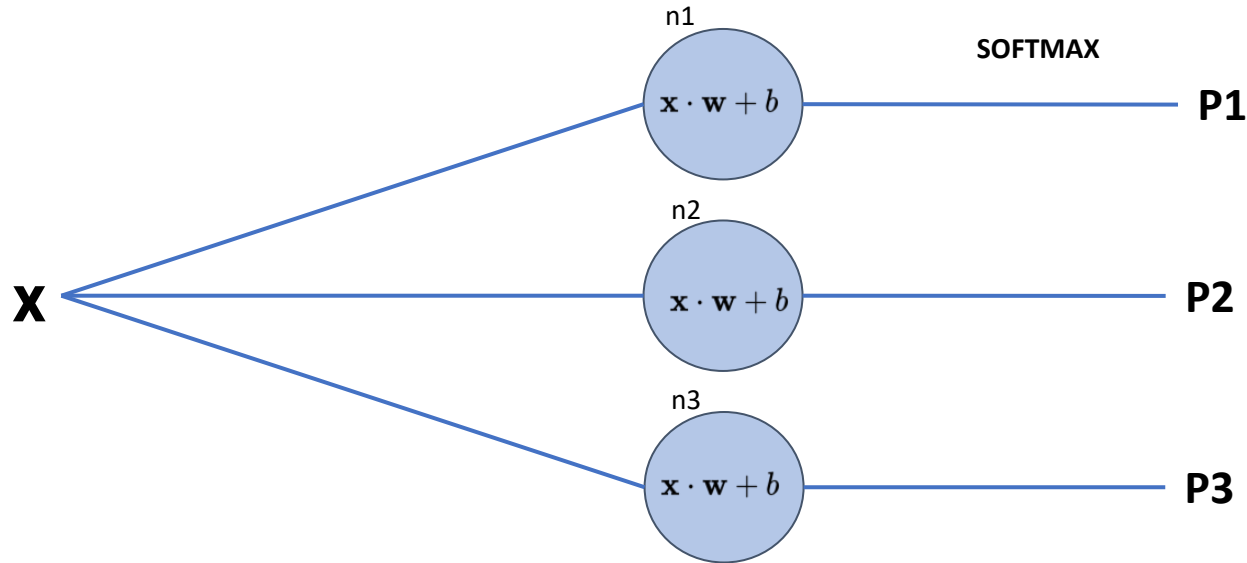


$$\sigma(-0.04) = 0.490$$


Rete neurale: più neuroni, un livello

Percettroni (Rosenblatt 1958; Minsky-Papert 1969)

Rete neurale addestrata per una classificazione multipla.



$$[x_1 \ x_2 \ x_3 \ x_4] \cdot \begin{pmatrix} w_1^{n1} & w_1^{n2} & w_1^{n3} \\ w_2^{n1} & w_2^{n2} & w_2^{n3} \\ w_3^{n1} & w_3^{n2} & w_3^{n3} \\ w_4^{n1} & w_4^{n2} & w_4^{n3} \end{pmatrix} + [b^{n1} \ b^{n2} \ b^{n3}] = [y^{n1} \ y^{n2} \ y^{n3}] \xrightarrow{\text{SOFTMAX}} [P1 \ P2 \ P3]$$

SOFTMAX FUCTION

Funzione che riconduce un vettore di valori su una distribuzione di probabilità in cui ciascun valore è compreso nell'intervallo $[0, 1]$ e la somma delle probabilità è pari a 1.

$$\text{softmax}(\mathbf{z}) = \left[\frac{e^{z_1}}{\sum_{i=1}^K e^{z_i}}, \frac{e^{z_2}}{\sum_{i=1}^K e^{z_i}}, \dots, \frac{e^{z_K}}{\sum_{i=1}^K e^{z_i}} \right]$$

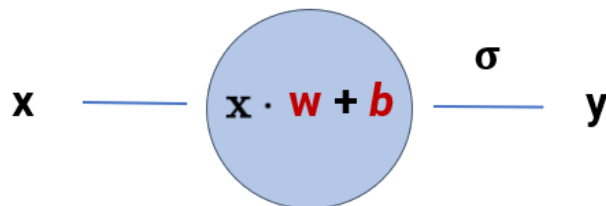
Deep learning

Il Machine Learning delle reti neurali profonde

Machine learning 1

Le reti neurali sono tipicamente reti che apprendono, che non debbono essere programmate a dare una certa prestazione ma che imparano da sole ad auto-organizzarsi in modo da dare quella prestazione.

La **conoscenza** posseduta da un rete neurale artificiale consiste nei pesi che stanno sulle connessioni. L'apprendimento, cioè l'acquisizione di nuova conoscenza, non può che consistere in una modificazione di questi pesi.



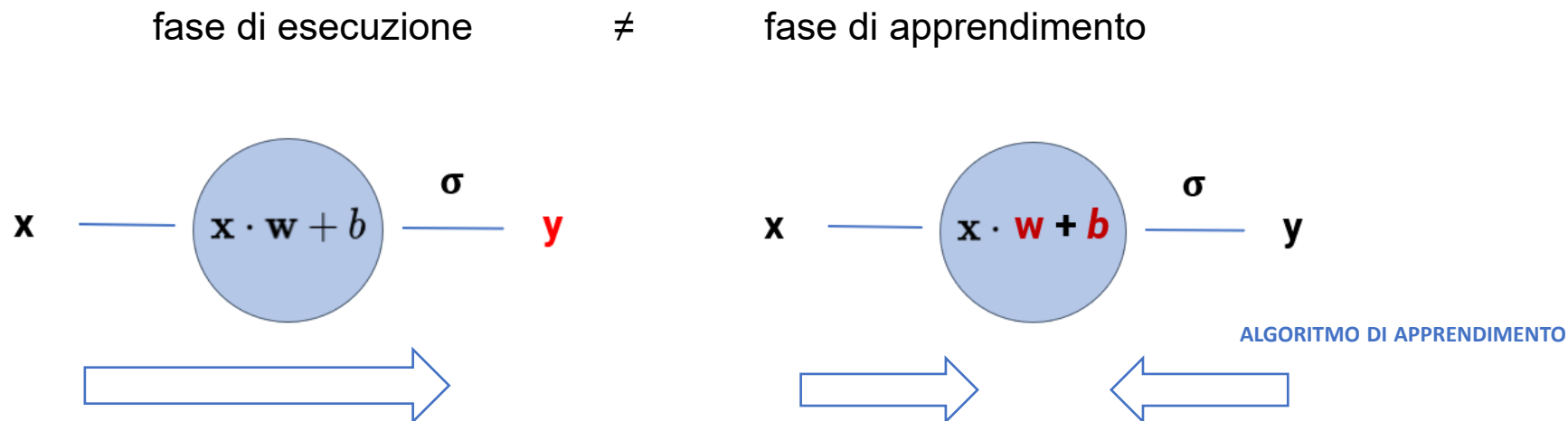
La reti neurali nascono senza conoscenza (diciamo con pesi assegnati in modo casuale) e apprendono spontaneamente con l'esperienza trovando progressivamente quei pesi che permettono alla rete di avere prestazioni desiderate.

"Instead of trying to reproduce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's?"

A. M. TURING, Computing Machinery and Intelligence, «Mind», New Series, vol. 59 n. 236 (ottobre 1950), p. 456.

Machine learning 2

Le reti neurali apprendono modificando gradualmente i propri valori sinaptici (weights e bias) attraverso la presentazione ripetuta di una serie di esempi.



I pesi delle sinapsi di una rete neurale sono i parametri che devono essere stimati durante l'apprendimento.

- Non è possibile individuare a priori i valori sinaptici di una rete neurale che debba esibire un certo comportamento, se non per casi rari ed estremamente semplici.
- Le reti neurali sono delle *black boxes*: presentano tanti parametri numerici (rappresentazioni quantitative dei pesi sinaptici) che risultano incomprensibili agli schemi di pensiero simbolici dell'uomo. Non danno spiegazioni sul loro funzionamento interno, perché non è frutto di una programmazione esplicita ma di un'auto-organizzazione. Il loro comportamento risulta opaco.

Machine learning – Algoritmi di apprendimento supervisionato

REGOLA DI HEBB

(D.O. Hebb, *The organization of behavior*, 1948)

PERCETTRONI SEMPLICI

(F. Rosenblatt, *Principles of neurodynamics*, 1962)

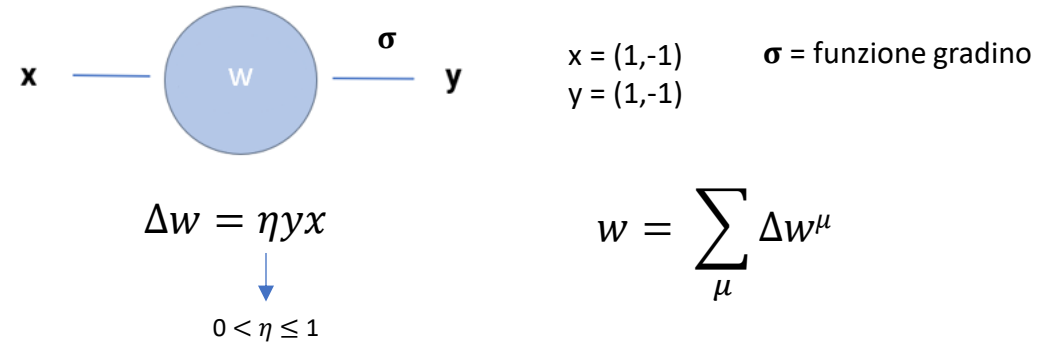
REGOLA DELTA O ADALINE

(B. Widrow, M. E. Hoff, *Adaptive switching circuits*, 1960)

BACK-PROPAGATION

(D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Learning representations by back-propagating errors*, 1986)

Teoria neuropsicologica per spiegare l'apprendimento umano



$y = output$ $t = output desiderato$ $error = (t - y)$

$$\Delta w_i = \eta(t - y)x_i$$

$$w_i^t = w_i^{t-1} + \Delta w_i^t \quad \text{weight} = \text{weight} + \eta(\text{error} * \text{input})$$

Per dati linearmente separabili, se esiste una soluzione (cioè un vettore di valori sinaptici che soddisfa la trasformazione richiesta tra input e output), la regola di apprendimento è in grado di trovarla in un numero finito di cicli di apprendimento.

Alternativa alla regola standard del perceptrone.

Estensione della regola delta per reti neurali multistrato.

Regola apprendimento percettrone semplice – esempio

PERCETTRONE

PREZZO
ALIM.

x_1
 x_2

$y = \begin{cases} 1 & \text{se } xw > 0 \\ -1 & \text{altrimenti} \end{cases}$

 $t = (1, -1)$

(0 elettr. - 1 ibrida -
2 benzina - 3 diesel)

TRAINING SET

$\begin{pmatrix} 17 \\ 1 \end{pmatrix}$

$= 1$

$\begin{pmatrix} 18 \\ 1 \end{pmatrix}$

$= 1$

$\begin{pmatrix} 18 \\ 0 \end{pmatrix}$

$= 1$

$\begin{pmatrix} 10 \\ 3 \end{pmatrix}$

$= 1$

$\begin{pmatrix} 13 \\ 2 \end{pmatrix}$

$= 1$

$\begin{pmatrix} 21 \\ 2 \end{pmatrix}$

$= -1$

$\begin{pmatrix} 23 \\ 1 \end{pmatrix}$

$= -1$

$\begin{pmatrix} 15 \\ 3 \end{pmatrix}$

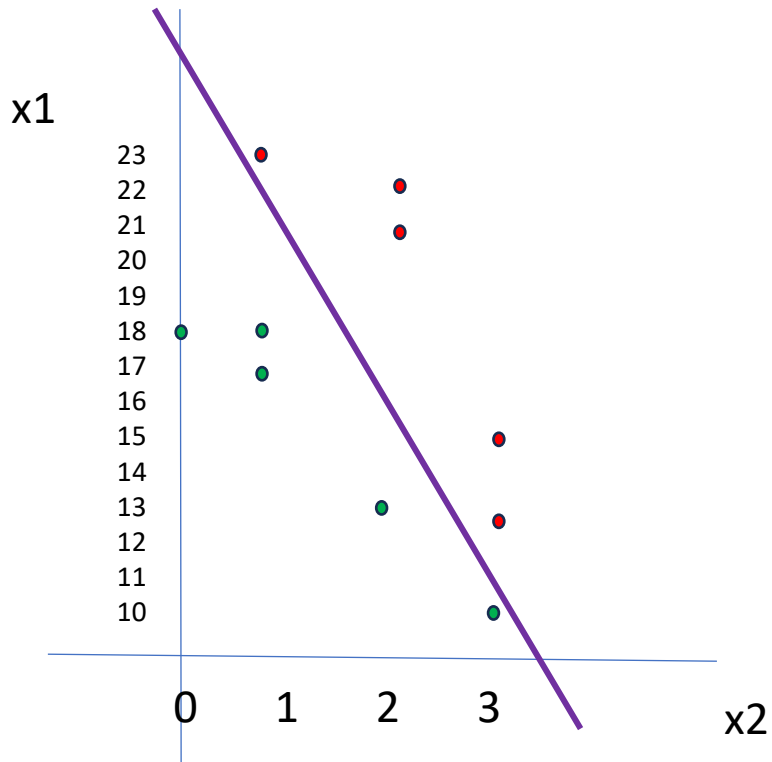
$= -1$

$\begin{pmatrix} 13 \\ 3 \end{pmatrix}$

$= -1$

$\begin{pmatrix} 22 \\ 2 \end{pmatrix}$

$= -1$



$\begin{pmatrix} 17 \\ 1 \end{pmatrix}$

$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$y = -1$

$t = 1$

$\Delta w_1 = \eta(1 - (-1)) x = + 34$

$\Delta w_2 = \eta(1 - (-1)) x = + 2$

$\begin{pmatrix} 21 \\ 2 \end{pmatrix}$

$\begin{pmatrix} 34 \\ 2 \end{pmatrix}$

$y = 1$

$t = -1$

$\Delta w_1 = \eta(-1 - 1) x = - 42$

$\Delta w_2 = \eta(-1 - 1) x = - 4$

$\begin{pmatrix} 18 \\ 1 \end{pmatrix}$

$\begin{pmatrix} -8 \\ -2 \end{pmatrix}$

$y = -1$

$t = 1$

$\Delta w_1 = \eta(1 - (-1)) x = + 36$

$\Delta w_2 = \eta(-1 - 1) x = + 2$

(...)

TRAINING
PER CICLI

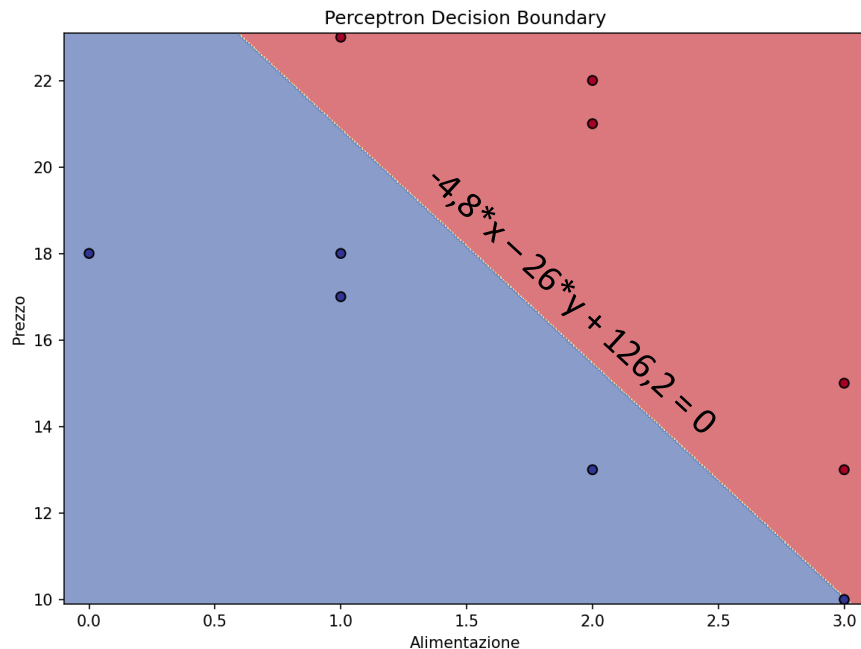
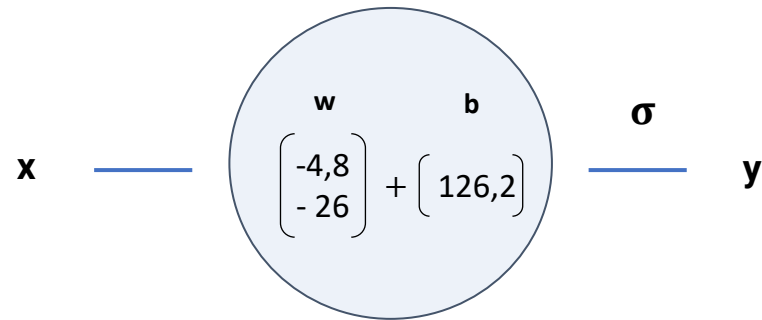
TRAINING
PER EPOCHE

per cicli = i pesi della rete sono calcolati dopo presentazione di ogni dato di addestramento

per epoche = i pesi della rete sono calcolati alla fine della presentazione di tutti i dati di addestramento

Regola apprendimento percettrone semplice - esempio

Con $\eta=0,1$, dopo 1.106 epoche (11.060 cicli):



Python code

```
import numpy as np

class Perceptron:
    def __init__(self, num_features, learning_rate=0.01):
        self.weights = np.zeros(num_features)
        self.bias = 0
        self.learning_rate = learning_rate

    def predict(self, inputs):
        activation = np.dot(inputs, self.weights) + self.bias
        return 1 if activation > 0 else -1

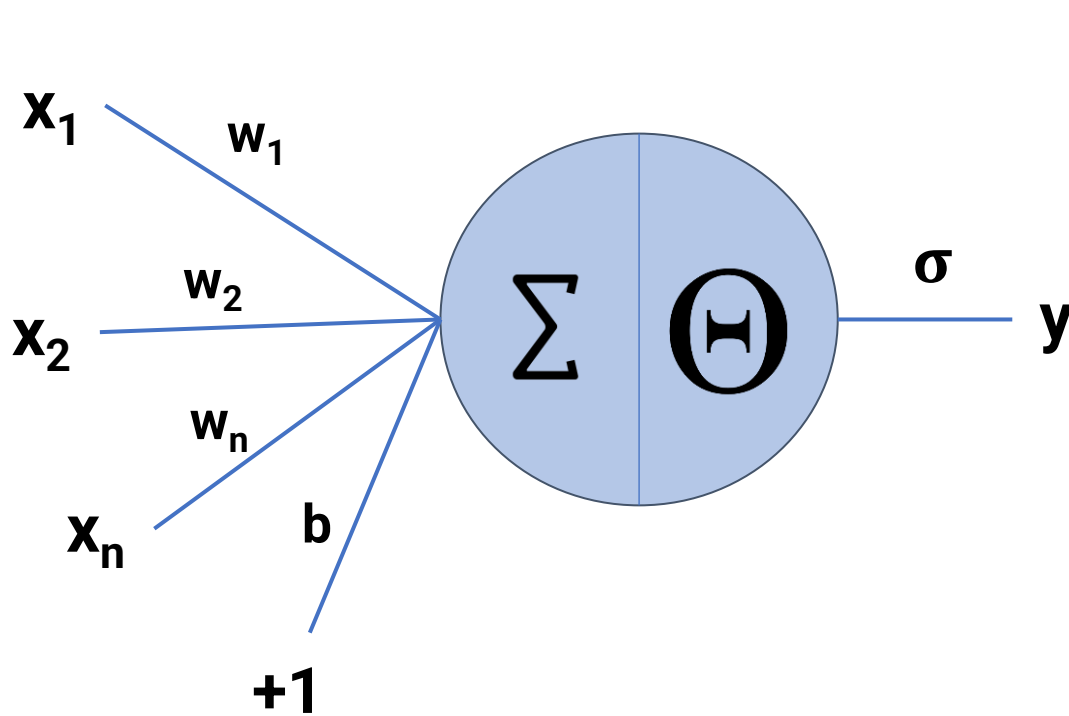
    def train(self, inputs, labels, epochs):
        for _ in range(epochs):
            for x, y in zip(inputs, labels):
                prediction = self.predict(x)
                error = y - prediction
                self.weights += self.learning_rate * error * x
                self.bias += self.learning_rate * error
                print(self.weights)
                print("bias =" + str(self.bias))

inputs = np.array([[17, 1], [21, 2], [18, 1], [23, 1], [15, 3], [18, 0], [10, 3], [13, 2], [13, 3], [22, 2]])
labels = np.array([1, -1, 1, -1, -1, 1, 1, 1, -1, -1])

perceptron = Perceptron(num_features=2, learning_rate=0.1)
perceptron.train(inputs, labels, epochs=1500)
```

Precisazione: apprendimento del *bias term*

Nella regola di apprendimento, consideriamo il bias come un elemento aggiuntivo del vettore di input (x), pari a 1, a cui dovrà corrispondere un altro valore del vettore dei pesi, da apprendere. Secondo i principi dell'algebra lineare, questo valore del vettore dei pesi corrisponde al bias.



$$\begin{bmatrix} x_1 & x_2 & x_n & +1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_n \\ b \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_n \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_n \end{bmatrix} + b$$

$$w = \eta(\text{error}) * x$$

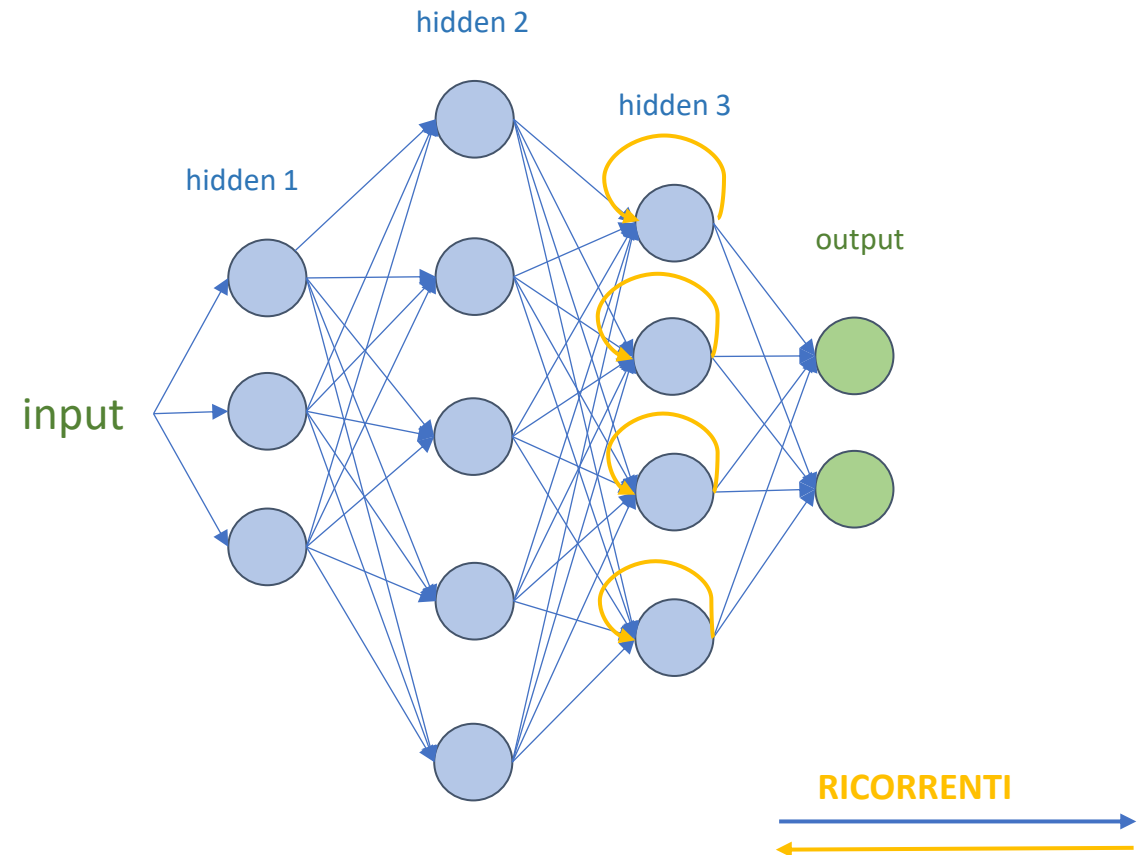
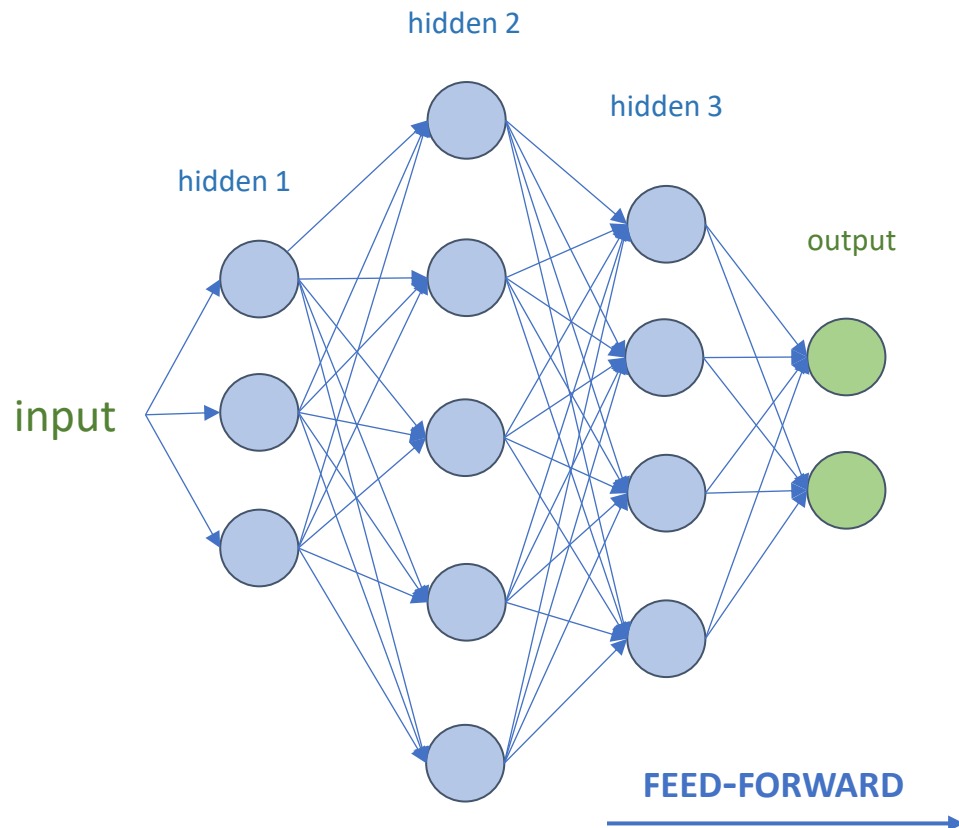
$$b = \eta(\text{error}) * 1$$

Reti neurali multistrato

Le reti neurali multistrato o profonde producono, nei loro *hidden layer*, delle **rappresentazioni interne dei concetti**.

Per il perceptrone semplice i concetti sono rappresentati esternamente, imposti dalle unità di input, a cui si insegnano determinate associazioni.

Nelle reti neurali a più livelli, **i concetti non sono imposti dall'esterno della rete, ma costruiti dalla rete stessa**.



Problemi di apprendimento delle reti neurali multistrato

- più la rete neurale è complessa (numero di neuroni e di pesi sinaptici), più deve aumentare il numero di dati di addestramento
- reti neurali complesse e dati di addestramento insufficienti o troppo numerosi possono compromettere la capacità di generalizzazione

DIMENSIONE E ARCHITETTURA

è fondamentale scegliere una rete neurale col numero adeguato di neuroni e di pesi



Metodi e algoritmi per la definizione di una rete neurale:

- Progressiva riduzione della rete:

RASOIO DI OCKAM di Thodberg

- Progressiva crescita della rete:

ALGORITMO DI CRESCITA DI CRESCITA DI FREAN

CASCADE CORRELATION

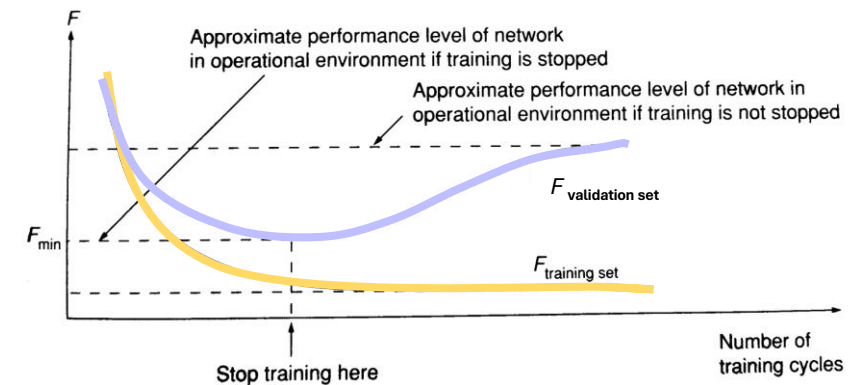
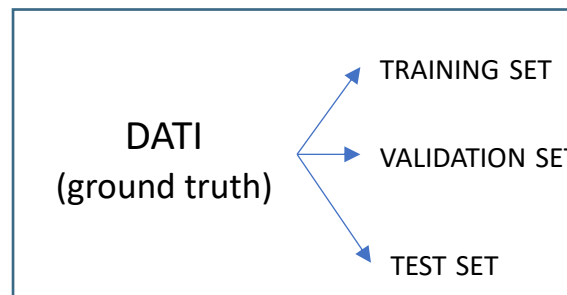
ecc...

QUANTITÀ DI ADDESTRAMENTO

è fondamentale scegliere il momento esatto in cui arrestare l'addestramento (prima che la rete cominci a imparare «troppo bene», *overfitting*)



Metodo empirico:



Tipi di apprendimento

SL

apprendimento supervisionato

- avviene tramite calcolo tra le predizioni del modello (output) per un determinato input e la risposta corretta (*gold standard* o *ground truth*)
- si serve di set di dati etichettati da esseri umani (dati strutturati)
- è utilizzato per modelli che svolgono attività che richiedono precisione

RL

apprendimento per rinforzo

- avviene tramite una serie di rinforzi: assegnazione di ricompense o punizioni (quantificabili in valori numerici) agli output forniti dal modello
- richiede un vasto numero di interazioni umane
- è utilizzato per migliorare prestazioni del modello non definibili in termini matematici

UL

apprendimento non supervisionato

- avviene tramite assegnazione di input che vengono classificati dal modello sulla base di caratteristiche comuni, individuate autonomamente, senza feedback espliciti
- È utilizzato per compiti 'esplorativi' come il clustering

SSL – app. autosupervisionato

addestramento che richiederebbe supervisione condotto in modo non supervisionato, usando come segnale di supervisione etichette implicite di dati non strutturati. La ground truth viene dedotta da dati non etichettati.

Rappresentazioni profonde

L'esempio del word embedding

Word2Vec con architettura Skip-Gram

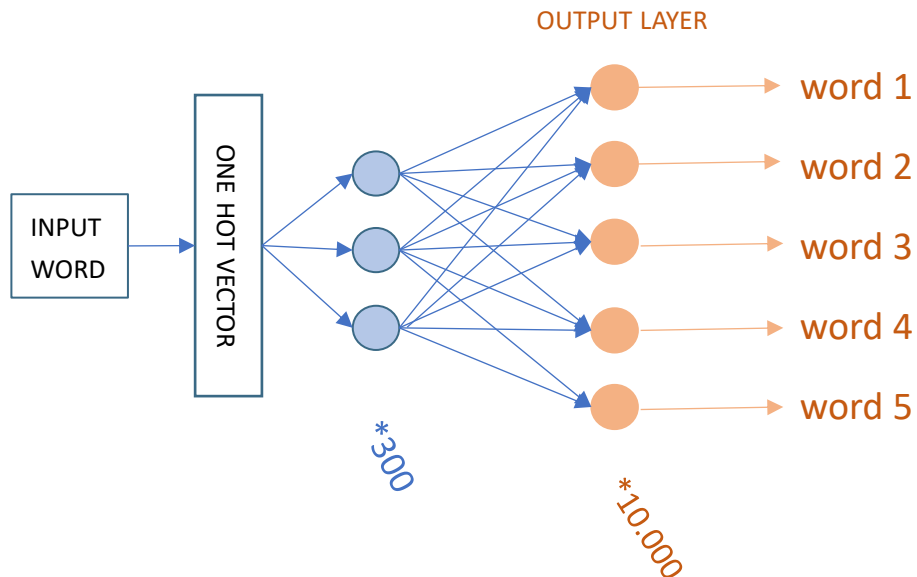
(T. Mikolov 2013)

Rete neurale artificiale a due strati con lo scopo di ottenere una rappresentazione del significato delle parole (*word embedding*).

“Rappresentazione distribuita” perché basata sull’ipotesi fondamentale della semantica distribuzionale: parole che hanno un contesto simile hanno anche un significato simile (*‘una parola si riconosce dalla compagnia che frequenta’* J. R. Firth)

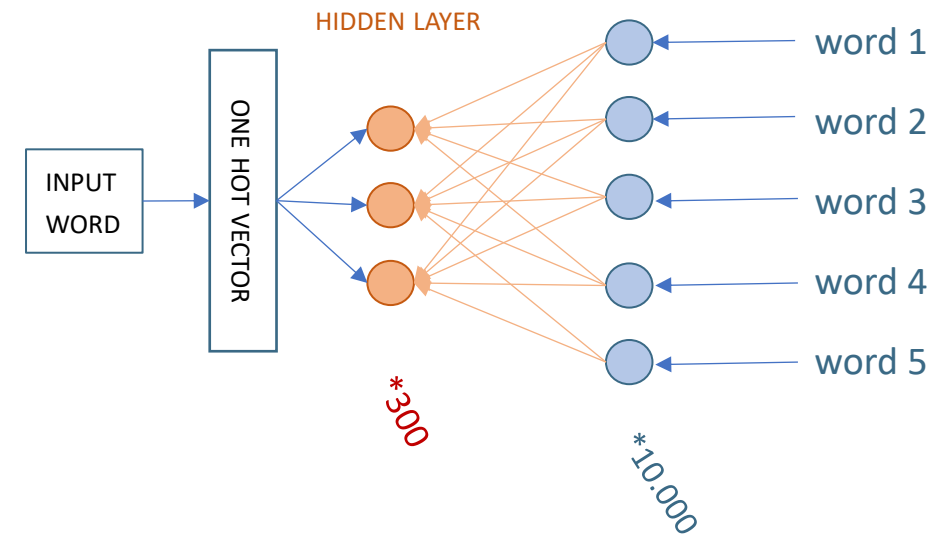
OBIETTIVO FASE DI ESECUZIONE

Le rappresentazioni vettoriali delle parole devono essere efficaci nel predire il contesto delle altre parole in una stessa frase.



OBIETTIVO FASE APPRENDIMENTO

L’obiettivo dell’addestramento della rete neurale è quello di apprendere le **rappresentazione vettoriali** delle parole (*embedding vectors*).



Word2Vec – esempio con Gensim

WORD VECTOR (WORD EMBEDDING)

Rappresentazione di una parola come vettore numerico in uno spazio multidimensionale, creata in un processo di addestramento.

Parole con significato simile hanno rappresentazioni simili.

```
from gensim.models import KeyedVectors
```

```
gensim_model =  
KeyedVectors.load_word2vec_format("C:\\Users\\LORENZO.BABINI\\Downloads\\GoogleNews-vectors-negative300.bin.gz",binary=True)
```

```
#print(gensim_model.most_similar(positive=["cat"]))
```

```
print(gensim_model["cat"])  
print(len(gensim_model["cat"]))
```

```
[ 0.0123291 0.20410156 -0.28515625 0.21679688 0.11816406 0.08300781 0.04980469 -0.00952148 0.22070312 -0.12597656 0.08056641 -0.5859375 -  
0.00445557 -0.296875 -0.01312256 -0.08349609 0.05053711 0.15136719 -0.44921875 -0.0135498 0.21484375 -0.14746094 0.22460938 -0.125 -0.09716797  
0.24902344 -0.2890625 0.36523438 0.41210938 -0.0859375 -0.07861328 -0.19726562 -0.09082031 -0.14160156 -0.10253906 0.13085938 -0.00346375  
0.07226562 0.04418945 0.34570312 0.07470703 -0.11230469 0.06738281 0.11230469 0.01977539 -0.12353516 0.20996094 -0.07226562 -0.02783203  
0.05541992 -0.33398438 0.08544922 0.34375 0.13964844 0.04931641 -0.13476562 0.16308594 -0.37304688 0.39648438 0.10693359 0.22167969  
0.21289062 -0.08984375 0.20703125 0.08935547 -0.08251953 0.05957031 0.10205078 -0.19238281 -0.09082031 0.4921875 0.03955078 -0.07080078 -  
0.0019989 -0.23046875 0.25585938 0.08984375 -0.10644531 0.00105286 -0.05883789 0.05102539 -0.0291748 0.19335938 -0.14160156 -0.33398438  
0.08154297 -0.7539062 0.10058594 -0.10449219 -0.12353516 -0.140625 0.03491211 -0.11767578 -0.1796875 -0.21484375 -0.23828125 0.08447266 -  
0.07519531 -0.25976562 -0.21289062 -0.22363281 -0.09716797 0.11572266 0.15429688 0.07373047 -0.27539062 0.14257812 -0.0201416 0.10009766 -  
0.19042969 -0.09375 0.14160156 0.17089844 0.3125 -0.16699219 -0.08691406 -0.05004883 -0.24902344 -0.20800781 -0.09423828 -0.12255859 -  
0.09472656 -0.390625 -0.06640625 -0.31640625 0.10986328 -0.00156403 0.04345703 0.15625 -0.18945312 -0.03491211 0.03393555 -0.14453125  
0.01611328 -0.14160156 -0.02392578 0.01501465 0.07568359 0.10742188 0.12695312 0.10693359 -0.01184082 -0.24023438 0.0291748 0.16210938  
0.19921875 -0.28125 0.16699219 -0.11621094 -0.25585938 0.38671875 -0.06640625 -0.4609375 -0.06176758 -0.14453125 -0.11621094 0.05688477  
0.03588867 -0.10693359 0.18847656 -0.16699219 -0.01794434 0.10986328 -0.12353516 -0.16308594 -0.14453125 0.12890625 0.11523438 0.13671875  
0.05688477 -0.08105469 -0.06152344 -0.06689453 0.27929688 -0.19628906 0.07226562 0.12304688 -0.20996094 -0.22070312 0.21386719 -0.1484375 -  
0.05932617 0.05224609 0.06445312 -0.02636719 0.13183594 0.19433594 0.27148438 0.18652344 0.140625 0.06542969 -0.14453125 0.05029297  
0.08837891 0.12255859 0.26757812 0.0534668 -0.32226562 -0.20703125 0.18164062 0.04418945 -0.22167969 -0.13769531 -0.04174805 -0.00286865  
0.04077148 0.07275391 -0.08300781 0.08398438 -0.3359375 -0.40039062 0.01757812 -0.18652344 -0.0480957 -0.19140625 0.10107422 0.09277344 -  
0.30664062 -0.19921875 -0.0168457 0.12207031 0.14648438 -0.12890625 -0.23535156 -0.05371094 -0.06640625 0.06884766 -0.03637695 0.2109375 -  
0.06005859 0.19335938 0.05151367 -0.05322266 0.02893066 -0.27539062 0.08447266 0.328125 0.01818848 0.01495361 0.04711914 0.37695312 -0.21875  
-0.03393555 0.01116943 0.36914062 0.02160645 0.03466797 0.07275391 0.16015625 -0.16503906 -0.296875 0.15039062 -0.29101562 0.13964844  
0.00448608 0.171875 -0.21972656 0.09326172 -0.19042969 0.01599121 -0.09228516 0.15722656 -0.14160156 -0.0534668 0.03613281 0.23632812 -  
0.15136719 -0.00689697 -0.27148438 -0.07128906 -0.16503906 0.18457031 -0.08398438 0.18554688 0.11669922 0.02758789 -0.04760742 0.17871094  
0.06542969 -0.03540039 0.22949219 0.02697754 -0.09765625 0.26953125 0.08349609 -0.13085938 -0.10107422 -0.00738525 0.07128906 0.14941406 -  
0.20605469 0.18066406 -0.15820312 0.05932617 0.28710938 -0.04663086 0.15136719 0.4921875 -0.27539062 0.05615234 ]
```

MOST SIMILAR WORD

```
gensim_model =  
KeyedVectors.load_word2vec_format("C:\\Users\\LORENZO.BABINI\\Downloads\\GoogleNews-  
vectors-negative300.bin.gz",binary=True)  
  
print(gensim_model.most_similar(positive=["cat"]))
```

```
[('cats', 0.8099379539489746), ('dog', 0.760945737361908), ('kitten', 0.7464985251426697), ('feline',  
0.7326234579086304), ('beagle', 0.7150582671165466), ('puppy', 0.7075453400611877), ('pup',  
0.6934291124343872), ('pet', 0.6891531348228455), ('felines', 0.6755931973457336) ...
```

CORRECT WORD

```
gensim_model =  
KeyedVectors.load_word2vec_format("C:\\Users\\LORENZO.BABINI\\Downloads\\GoogleNews-  
vectors-negative300.bin.gz",binary=True)  
  
print(gensim_model.most_similar(positive=["pet", "woof"], topn=1))
```

```
[('dog', 0.7027985453605652)]
```

SAME SIMILARITY

```
gensim_model =  
KeyedVectors.load_word2vec_format("C:\\Users\\LORENZO.BABINI\\Downloads\\GoogleNews-  
vectors-negative300.bin.gz",binary=True)  
  
print(gensim_model.similarity('italy', 'rome'))  
print(gensim_model.similarity('france', 'paris'))  
print(gensim_model.similarity('germany', 'berlin'))
```

```
0.56735367
```

```
0.5550796
```

```
0.5539934
```

Fonti: [CambridgeSpark](#), [Radimrehurek](#)

Word embeddings

PRINCIPALI MODELLI

WORD2VEC (2013; Google)

GLOVE (2014; Stanford)

FASTTEXT (2016; Meta)

COSA RAPPRESENTANO?

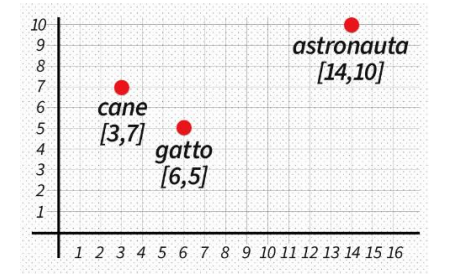
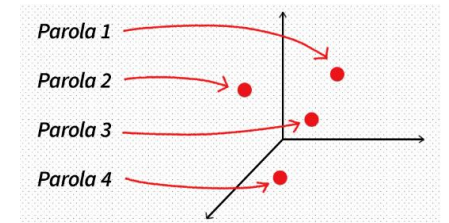
- RELAZIONI SEMANTICHE (es. re - regina)
- RELAZIONI GRAMMATICALI (es. mangiare - camminare)
- ANALOGIE COMPLESSE (es. Parigi - Francia; Roma - Italia)

COME SI ADDESTRANO?

- INIZIALIZZAZIONE: viene assegnato un vettore con valori casuali a ciascuna parola
- ANALISI del contesto: per ogni parola del corpus, vengono estratte le parole di contesto
- PREDIZIONE: il modello prova a predire
 - la parola da un contesto (CBOW)
 - il contesto da una parola (Skip-gram)
- CALCOLO DELL'ERRORE: differenza tra la predizione effettuata e le parole del corpus
- CORREZIONE: aggiornamento delle rappresentazioni vettoriali per minimizzare l'errore
- ITERAZIONE: molteplice ripetizione dei passaggi

Language models

- insieme di correlazioni statistico-probabilistiche tra le parole, ciascuna delle quali è rappresentata da un'ampia matrice di valori numerici (*word embedding*). Ogni matrice, raffigurabile in uno spazio astratto e multidimensionale, esprime le modalità d'uso della parola nella lingua. (N.B. Le caratteristiche e le dimensioni di ciascuna matrice/parola non corrispondono in alcun modo a categorie grammaticali o semantiche umane).
- sono creati utilizzando reti neurali profonde (in grado di gestire dati sequenziali), addestrate con dataset in linguaggio naturale.
- svolgono un'ampia varietà di compiti linguistici e sono utilizzati in diversi campi (riconoscimento vocale, traduzioni automatiche, generazione automatica di testo, estrazione dell'informazione ecc...)



Source: DATAPIZZA

Uno dei compiti più importanti è quello di *predire* una parola futura in una sequenza:

$w1 \ w2 \ w3 \ w4 \rightarrow w5$

Presupposti:

- ogni parola di una sequenza contribuisce a determinare le parole successive
- in una sequenza, ogni parola è influenzata dalle parole precedenti, mai da quelle successive.

Le reti neurali fondamentali

CNN, RNN, LSTM e Transformer

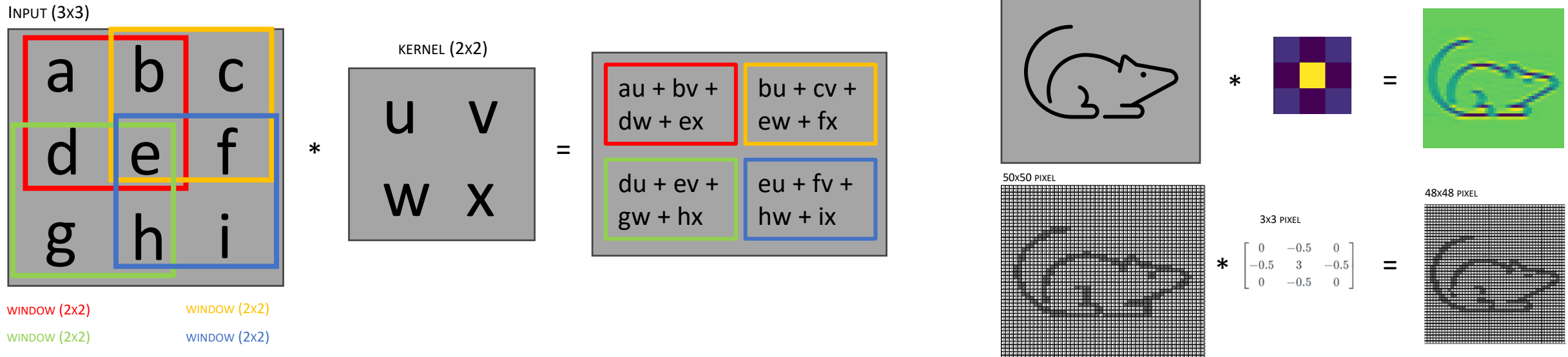
CNN - Reti neurali convoluzionali

(Fukushima 1980; LeCun 1989)

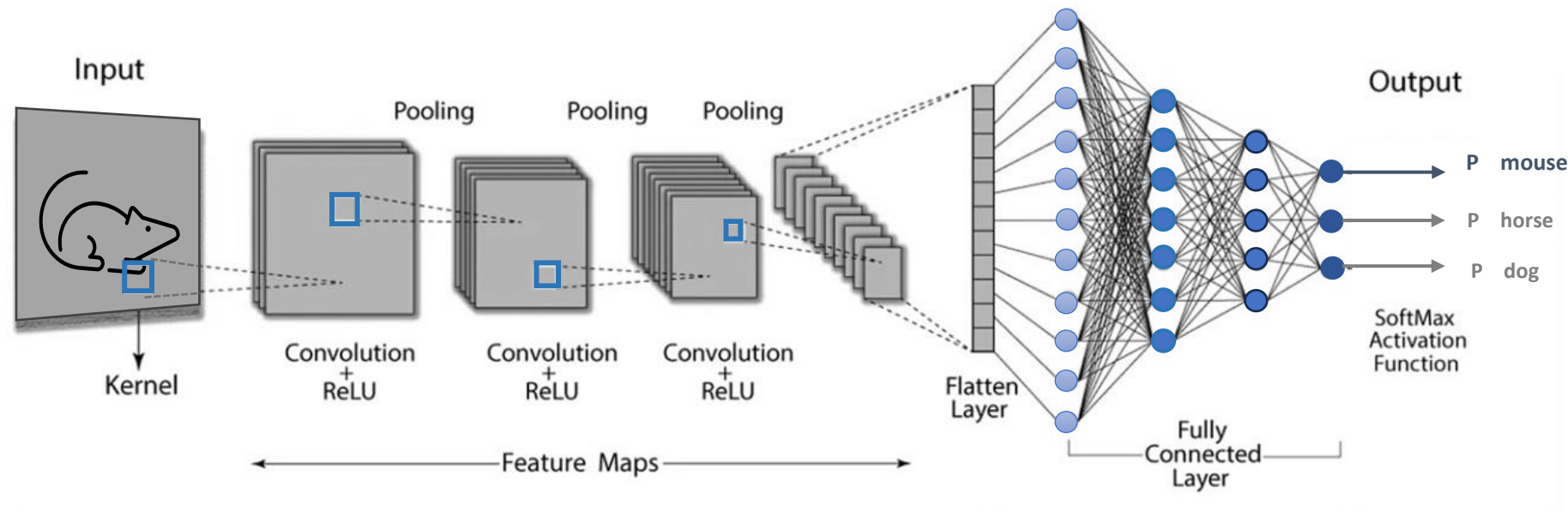
- ispirate agli studi sulla **corteccia visiva degli animali** dei premi Nobel D.H. Hubel e T. Wiesel
- progettate per elaborare dati strutturabili in griglie, cioè definiti dalla combinazione di elementi contigui, **in una sequenza spaziale** → immagini (2D), suoni (1D), video (3D)
- reti neurali che si avvalgono dell'operazione nota come **convoluzione**, al posto della semplice moltiplicazione tra matrici.

CONVOLUZIONE

Nella CNN i dati di input sono matrici, che vengono combinate con i pesi, rappresentati da matrici di dimensione minore (kernel). I kernel vengono fatti scorrere lungo la superficie della matrice di input, per ottenere una matrice di output.



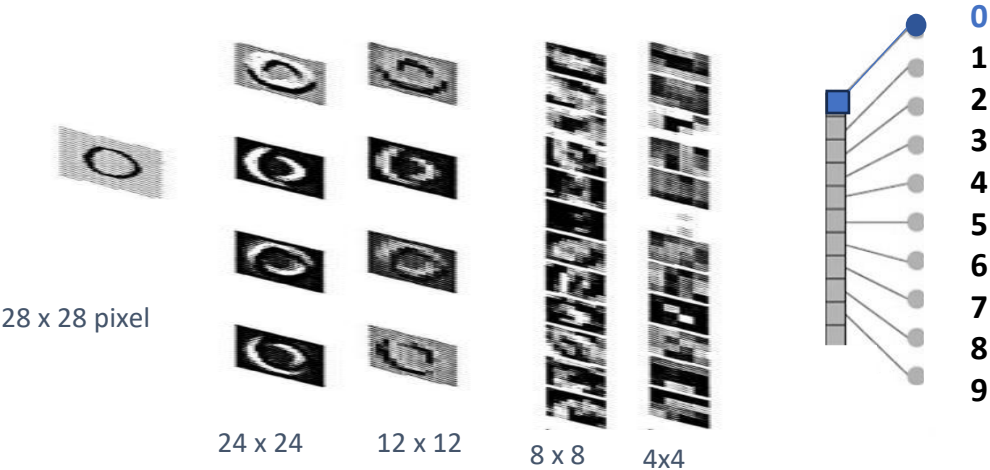
CNN - applicazioni



Handwritten Digit Recognition

LeCun 1989

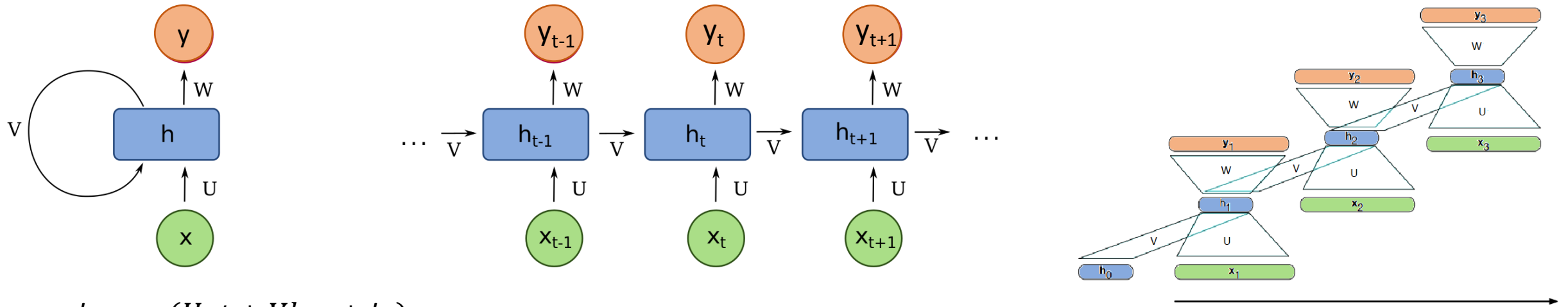
1 4 1 0 1 1 9 1 5 4 8 5 7 2 6 8 0 3 2 2 6 4 1 4 1
8 6 6 3 5 9 7 2 0 2 9 9 2 9 7 2 2 5 1 0 0 4 6 7
0 1 3 0 8 4 1 1 1 5 9 1 0 1 0 6 1 5 4 0 6 1 0 3 6
3 1 1 0 6 4 1 1 1 0 3 0 4 7 3 2 6 2 0 9 7 7 9 9
6 6 8 4 1 2 0 8 6 7 2 8 5 5 7 1 3 1 4 2 7 9 5 5 4
6 0 1 0 1 8 7 5 0 1 8 7 1 1 2 9 9 1 0 8 9 9 7 0 9
8 4 0 1 0 9 7 0 7 5 9 7 3 3 1 9 7 2 0 1 5 5 1 9 0
5 5 1 0 7 3 5 1 8 2 5 5 1 8 2 8 1 4 3 5 8 0 1 0 9
4 3 1 7 8 7 5 5 1 6 5 5 4 6 0 3 5 4 6 0 3 5 4 6 0
5 5 1 8 2 5 5 1 0 8 5 0 3 0 6 7 5 2 0 4 3 9 4 0 1



RNN - Reti neurali ricorrenti

(Jordan 1989, Elman 1990)

- ideate per elaborare **dati sequenziali** di diversa lunghezza, rappresentabili in una **serie temporale** (linguaggio, variazioni nel tempo, DNA, scrittura a mano)
- caratterizzate da uno strato nascosto di neuroni (*hidden state*) che mantiene **memoria dell'informazione degli input precedenti**
- applicano un **loop interno** (connessione ciclica) per combinare un nuovo input alla rappresentazione ottenuta dall'input precedente

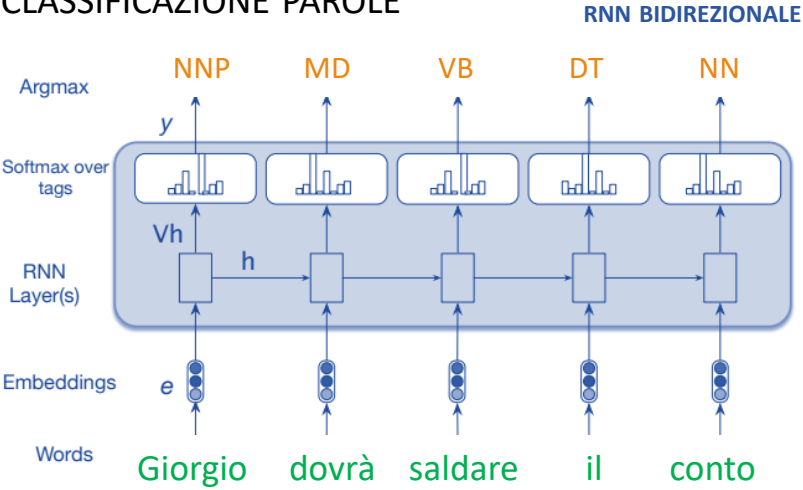


$$h_t = \sigma_h(Ux_t + Vh_{t-1} + b_h)$$

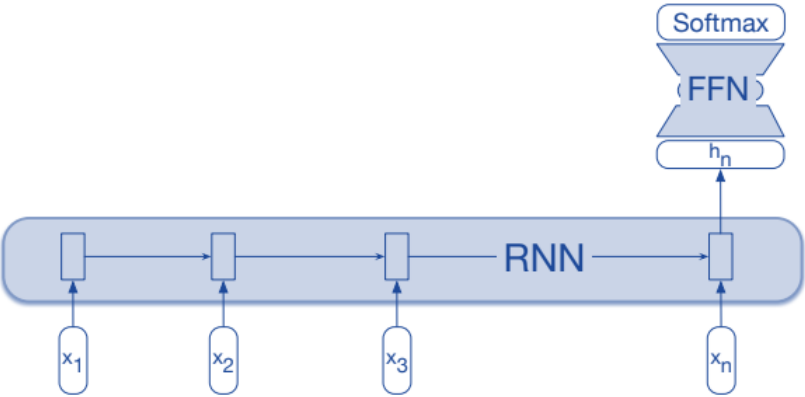
$$y_t = \sigma_y(W h_t + b_y)$$

RNN – applicazioni di NLP

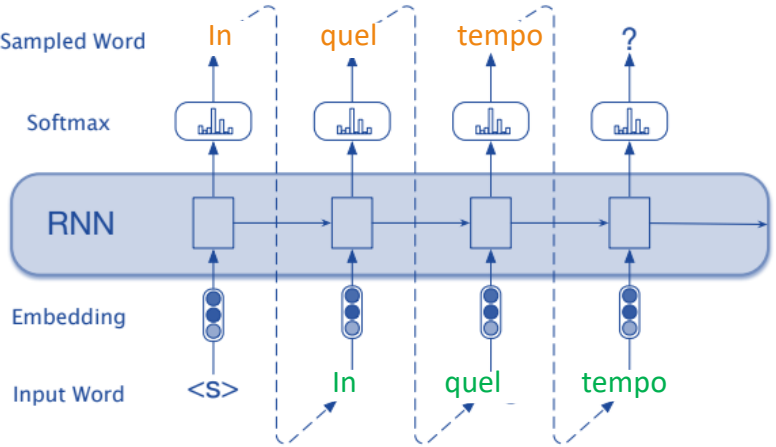
CLASSIFICAZIONE PAROLE



CLASSIFICAZIONE FRASI



GENERAZIONE DI TESTO



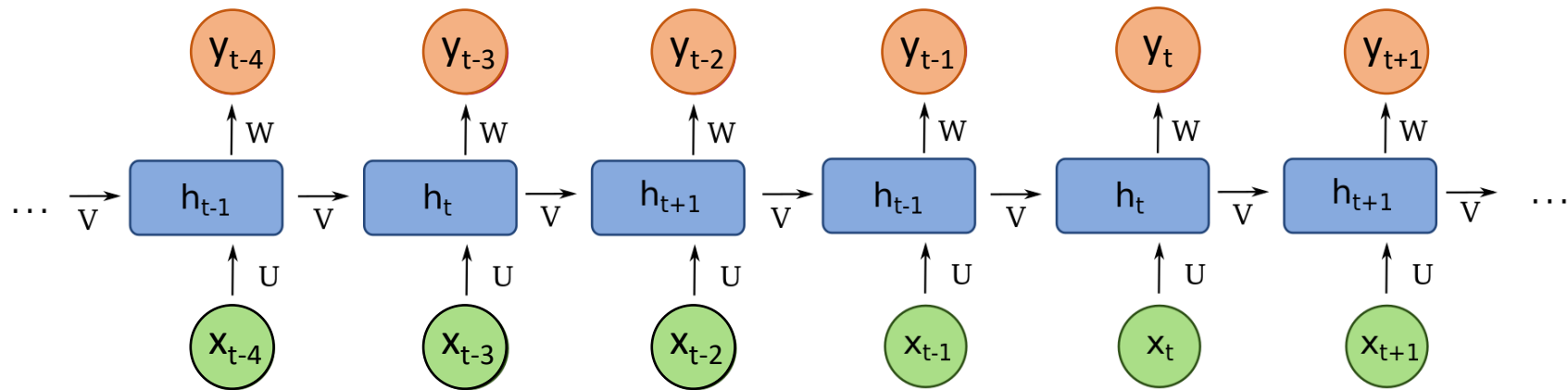
IA generativa ≠ IA discriminativa

autoregressive generation

FONTE DELLE ILLUSTRAZIONI: D. Jurafsky, J.H. Martin, *Speech and Language Processing* (2023), pp. 192-195.
POS TAG LIST: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

RNN – problemi

- esplosione/scomparsa del gradiente
- distorsioni di memoria (effetto «gioco del telefono»)



Gli atleti, che hanno vinto le qualificazioni locali e sono arrivati alle finali di Tokyo, ora ...

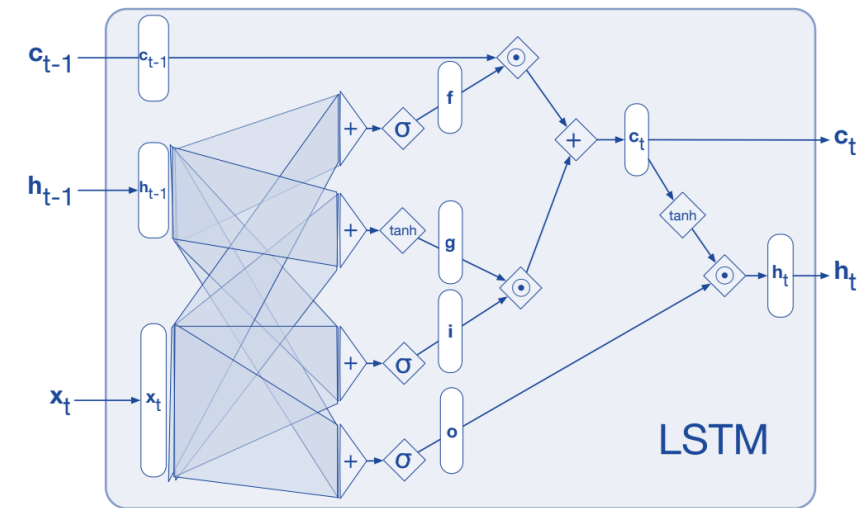
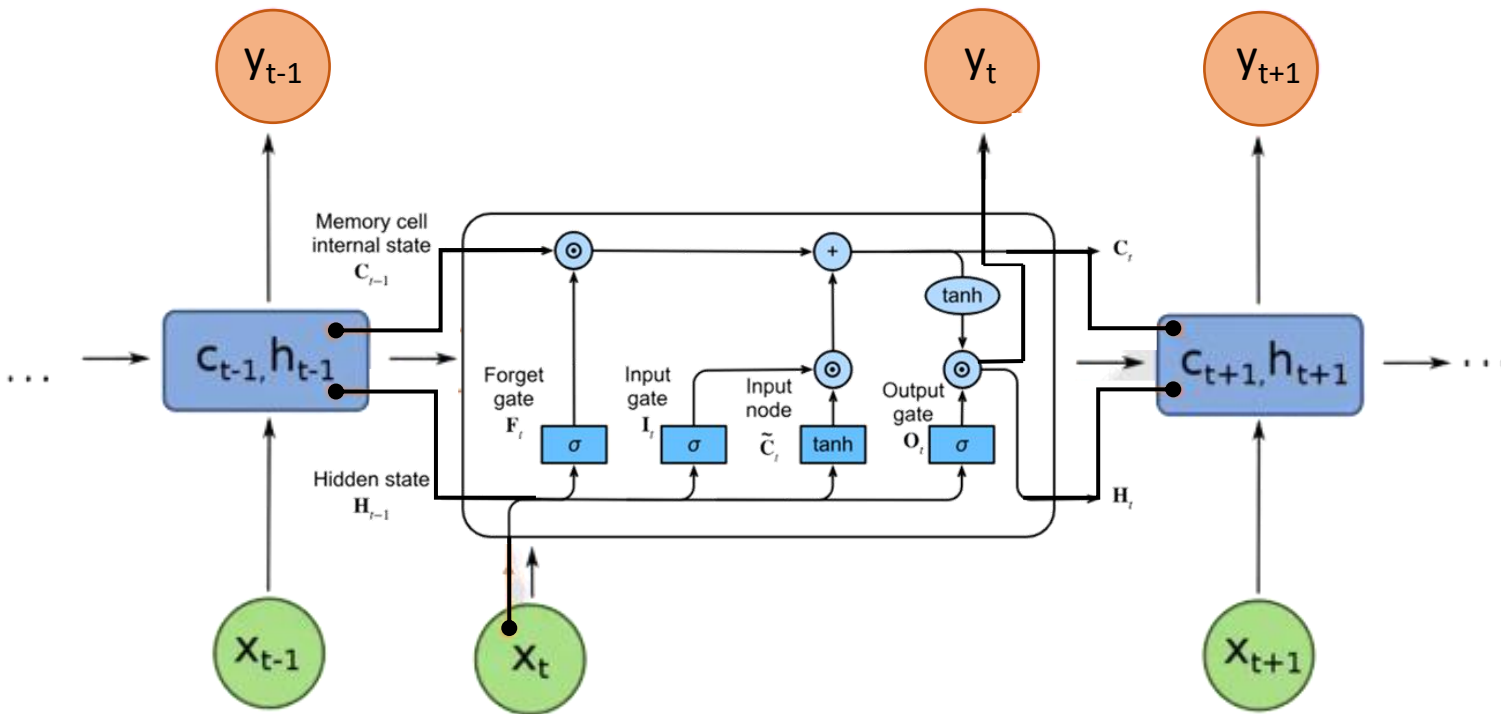
☐ gareggia

☐ gareggiano

LSTM - Long Short Term Memory

(Hochreiter 1997)

- è una RNN ottimizzata per sequenze lunghe e complesse di input
- è caratterizzata da una componente di memoria a lungo termine (c , o *memory cell*) che scorre da un passo temporale all'altro, arricchendosi di nuove informazioni, ma senza espandersi in modo moltiplicativo
- prevedono dei varchi (*internal gates*) per il controllo dei flussi di informazioni



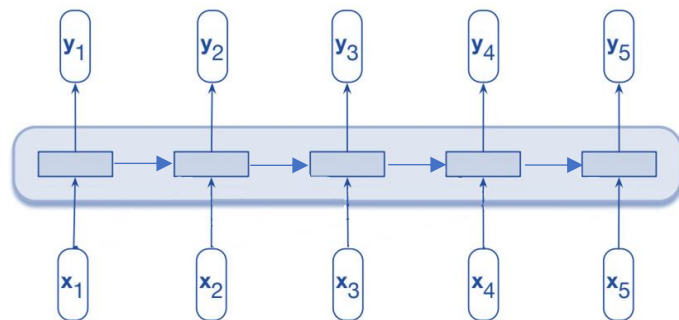
FONTE: D. Jurafsky, J.H. Martin, [Speech and Language Processing](#) (2024), p. 199.

Transformer

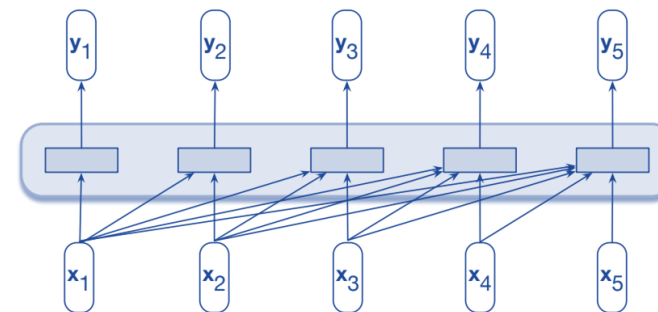
(Vaswani et al. 2017)

- rete neurale in grado di processare lunghe catene di dati sequenziali senza ricorrere al meccanismo della ricorrenza, ma svolgendo calcoli che possono anche essere svolti **in parallelo**, perché indipendenti l'uno dall'altro.

RNN



Transformer



- interpreta e rappresenta i legami tra gli elementi di una lunga sequenza attraverso un **meccanismo di auto attenzione** (*self attention layer*)

“Self-attention [...] is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence”

A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A.N. GOMEZ, L. KAISER, I. POLOSUKHIN, [Attention Is All You Need](#), Advances in Neural Information Processing Systems 30, New York 2017

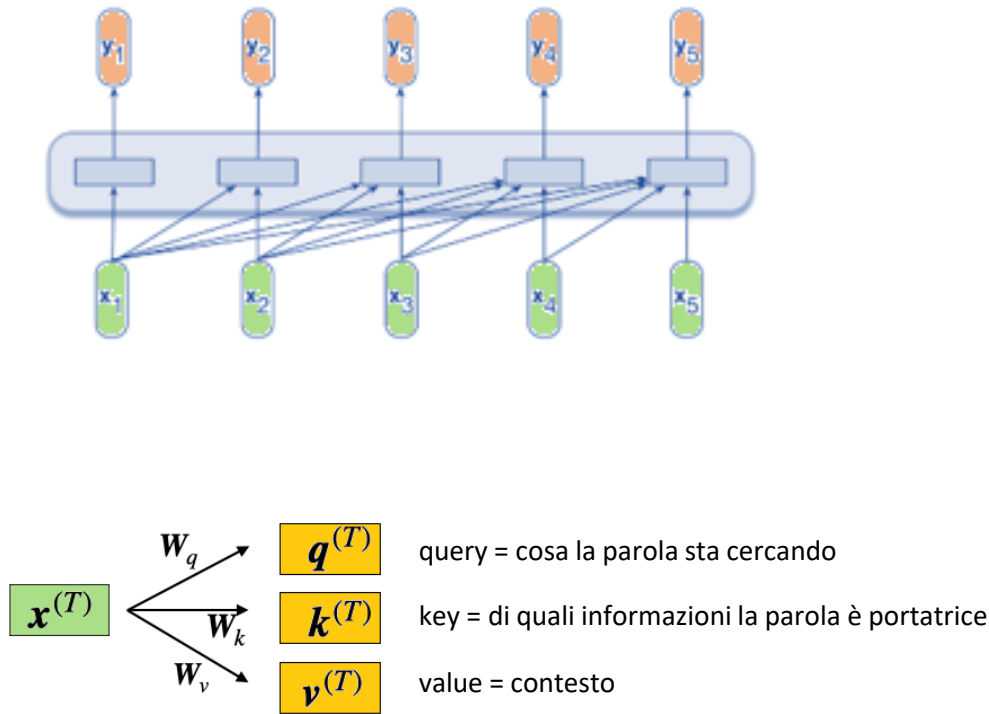
NONOSTANTE LUIGI NON AVESSSE MAI VISTO **GIANNI** FINO A QUEL MOMENTO, SI TROVÒ SUBITO D'ACCORDO CON ...

NONOSTANTE LUIGI NON AVESSSE MAI VISTO **ANNA** FINO A QUEL MOMENTO, SI TROVÒ SUBITO D'ACCORDO CON ...

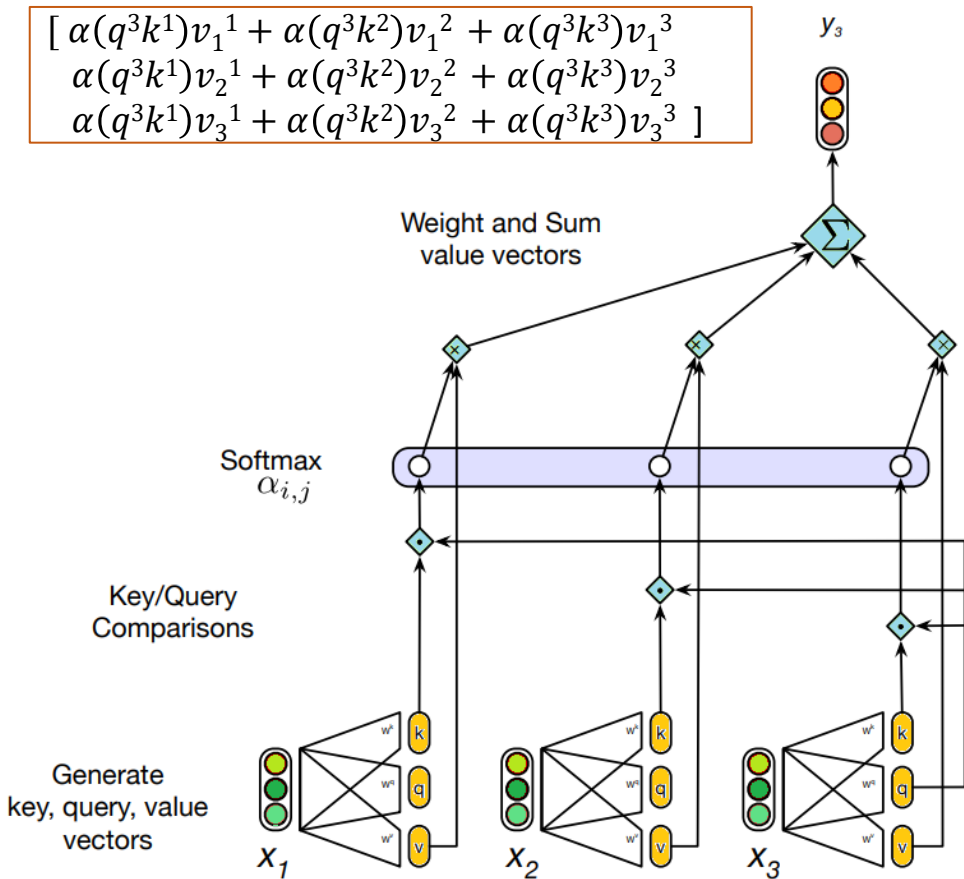


Self-attention layer

Un modo per costruire una rappresentazione contestualizzata del significato di ogni parola (*contextual embedding*), cioè che integri informazioni sulle parole precedenti, aiutando il modello a capire come le parole si relazionino tra loro in ampi tratti di testo.



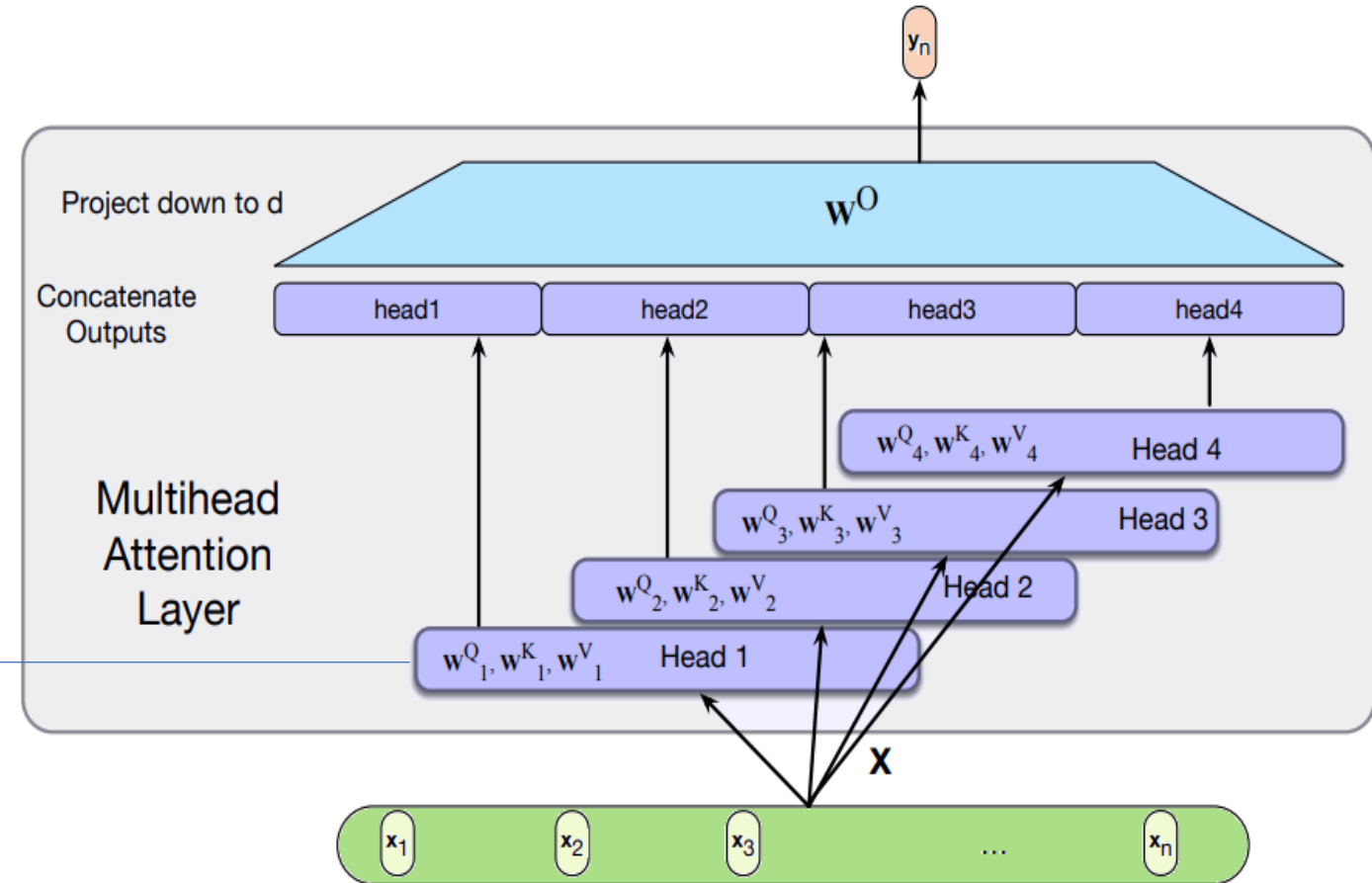
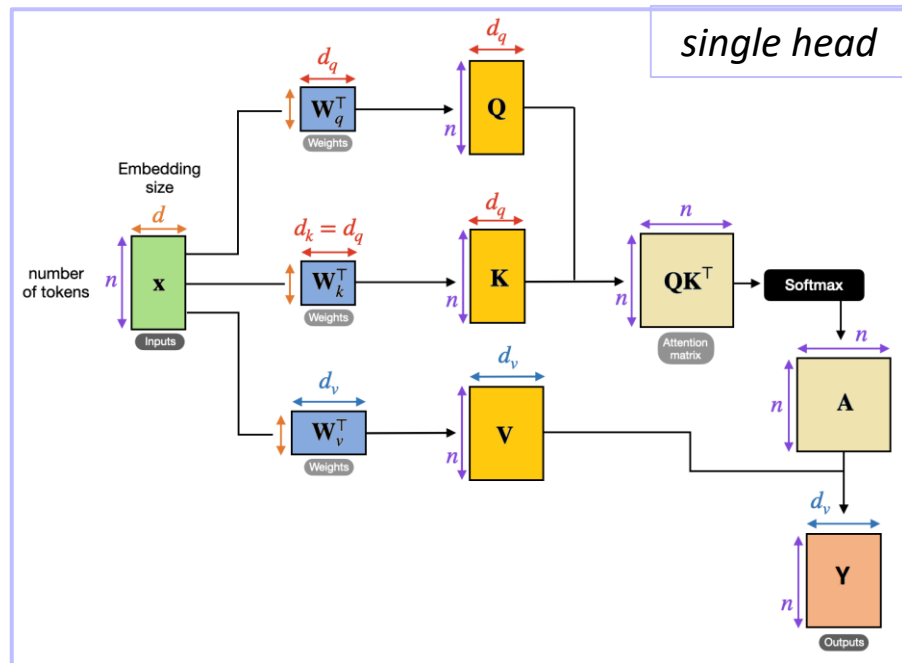
$$y^T = \sum_{N \leq T} (\text{softmax}(q^T k^N)) v^N$$



FONTE: D. Jurafsky, J.H. Martin, [Speech and Language Processing](#) (2024), p. 219.

Multihead attention

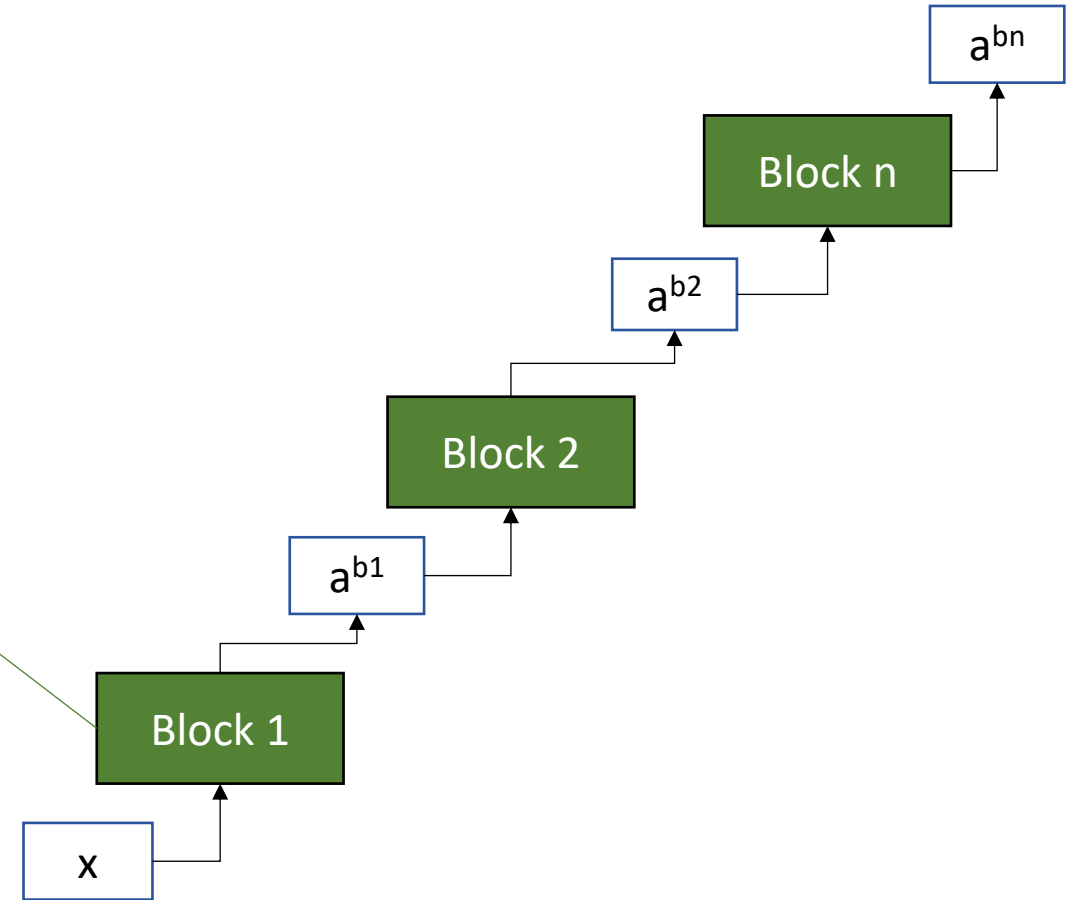
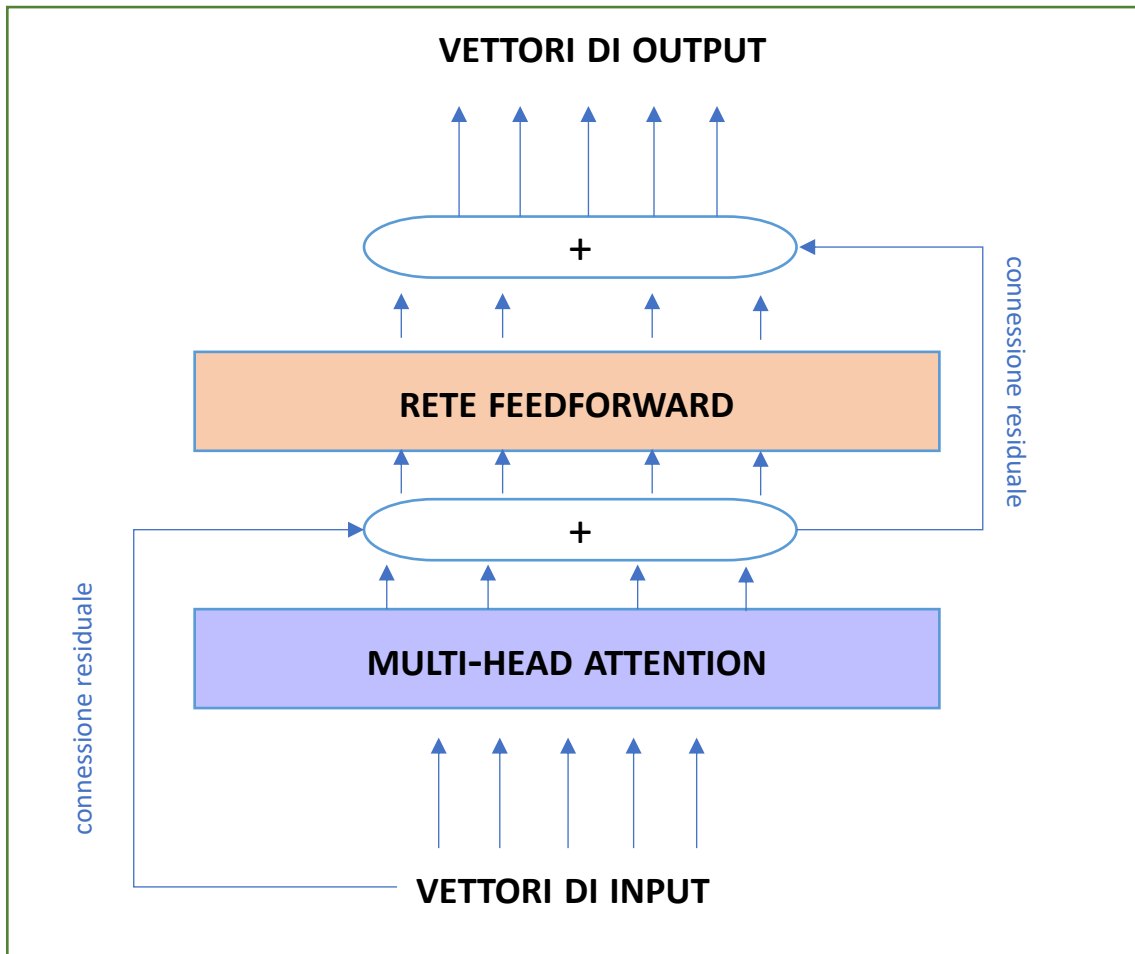
Le parole possono relazionarsi in modi diversi simultaneamente (relazioni semantiche, sintattiche, polirematiche, di analisi logica ecc...)
Una singola componente di self attention non riesce a esprimere tutti i tipi di relazione, per questo ne vengono usati molteplici.



FONTE: D. Jurafsky, J.H. Martin, [Speech and Language Processing](#) (2023).

Transformer Block

Una rete neurale profonda basata sul meccanismo parallelizzato delle *multihead attention*, unito ad altre reti neurali e sistemi di calcolo suddivisibili in blocchi.



Transformer Blocks + Language Modeling Head

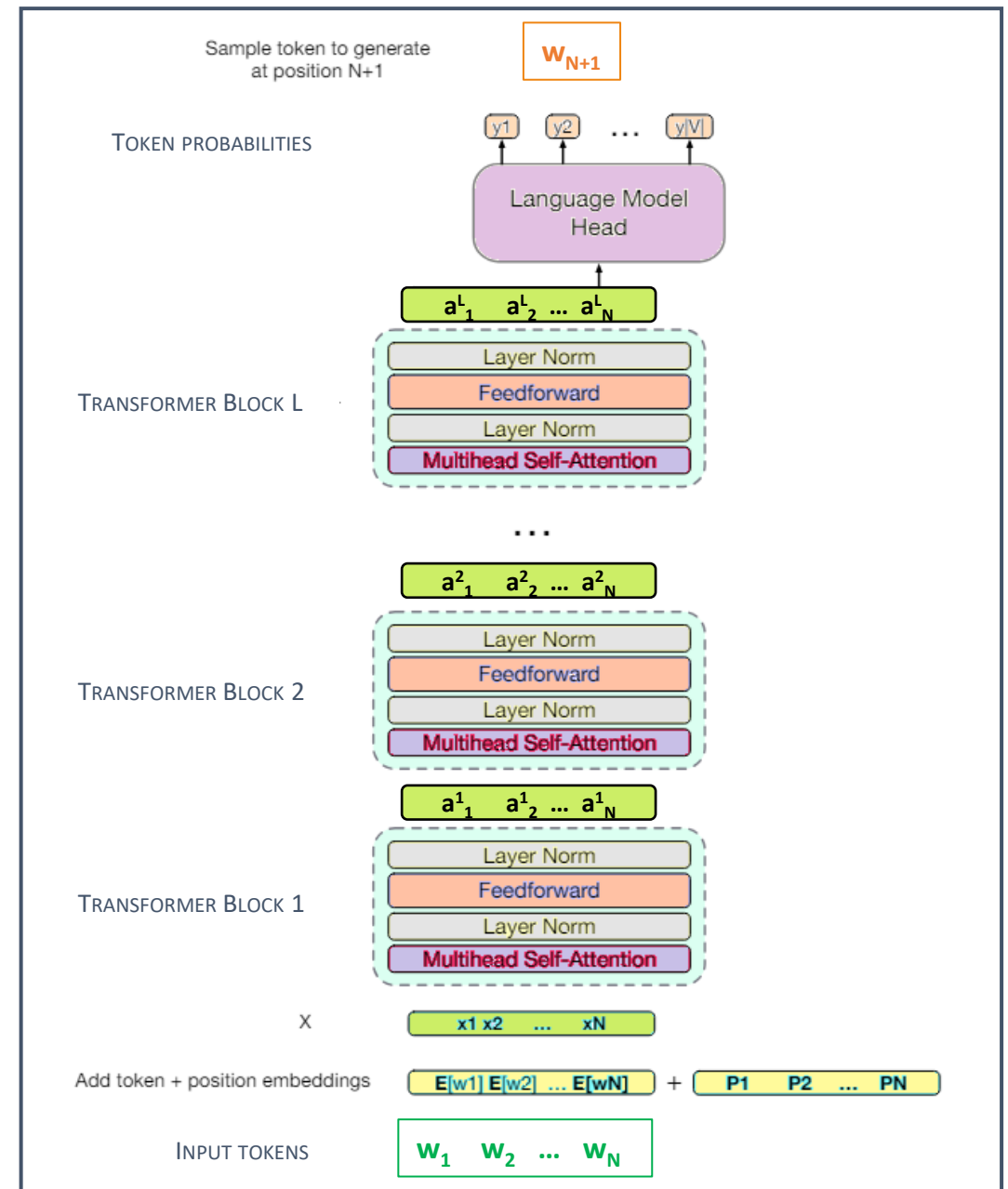
Per **Language Modeling Head** s'intende la rete neurale aggiuntiva, posta in cima all'architettura di base di un Trasformer, per svolgere un determinato compito linguistico.

Può usare l'output dell'ultima parola (a_N) dell'ultimo Transformer Block (L) per predire la parola successiva:

$$a_N^L \longrightarrow w_{N+1}$$

TRANSFORMER
DECODER-ONLY MODEL

FONTE: D. Jurafsky, J.H. Martin, [Speech and Language Processing](#) (2024).



Large Language Models

Sono modelli creati per svolgere compiti linguistici, addestrati con vastissime quantità di dati, utilizzando reti neurali con architettura Transformer.

2018



Bidirectional Encoder Representations from Transformers

Open Source

2018 - oggi



GPT

Generative Pre-trained Transformer
(ChatGPT, 2022-oggi)

GPT2 in versione pubblica

2023 - oggi



2023 - oggi

LLaMA
by  Meta

Large Language Model Meta AI

Open Source

2023 - oggi



Language Model for Dialogue Applications - Pathways Language Model

2024 - oggi



Open Source



Hugging Face

...

OPEN SOURCE LLMs

Bibliografia

INTELLIGENZA ARTIFICIALE – APPROCCIO GENERALE E STORIA

S. RUSSEL, P. NORVING, *Artificial intelligence: a modern approach* (4° edition), tradotto come *Intelligenza artificiale: un approccio moderno*, Pearson, Milano 2021 (quarta edizione).

D. ANDLER, *Intelligence artificielle, intelligence humaine: la double énigme* (2023), tradotto come *Il duplice enigma*, Einaudi, Torino 2024.

A. M. TURING, [*Computing Machinery and Intelligence*](#), «Mind», New Series, vol. 59 n. 236 (ottobre 1950), p. 456; anche in [traduzione italiana](#).

DEEP LEARNING

Y. LECUN, Y. BENGIO, G. HINTON, [*Deep learning*](#). Nature 521, pp. 436-444, 2015.

S. CANU, [*Machine Learning, deep learning and optimization in Computer Vision*](#), slides per Tokio QCAV 2017, 15 Maggio 2017.

B. LIQUET, S. MOKA, Y. NAZARATHY, [*Mathematical Engineering of Deep Learning*](#), in deeplearningmath.org, 2017.

Y. LECUN, Y. BENGIO, G. HINTON, [*Deep learning for AI*](#), Communications of the ACM, Volume 64, Issue 7, pp. 58-65, 21 giugno 2018.

AA.VV., [*Intelligenza artificiale e mondo radiotelevisivo*](#), numero monografico di “Elettronica e telecomunicazioni”, anno LXIX, n. 1/2020.

S. WEIDMAN, *Deep learning from scratch*, tradotto come *Deep learning. Dalle basi alle architetture avanzate con Python*, Tecniche nuove, Milano 2020.

S. THEODORIDIS, *Machine Learning: A Bayesian and Optimization Perspective*, Academic Press, Amsterdam 2020 (seconda edizione), cap. XVIII.

A. ZHANG, Z. C. LIPTON, M. LI, [*Dive into Deep Learning*](#), Cambridge University Press, Cambridge 2023.

DEEP LEARNING PER NLP - MANUALI

G. PAAß, S. GIESSELBACH, [*Foundation Models for Natural Language Processing*](#), Springer, 2023.

D. JURAFSKY, J. H. MARTIN, [*Speech and Language Processing*](#) (Third edition), in web.stanford.edu/~jurafsky/slp3/, 2024.

APPROCCIO SIMBOLICO - GOFAI

- J. MCCARTHY, M. MINSKY, N. ROCHESTER, C.E. SHANNON, [A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence](#), 1955, tradotto anche [in italiano](#)
- A.L. SAMUEL, [Some Studies in Machine Learning Using the Game of Checkers](#), IBM Journal, Vol. 3, n.3 1959.
- A. NEWELL, H.A. SIMON, [Computer Science as Empirical Inquiry: Symbols and Search](#), "Communications of the ACM", vol. 19 n. 3 (marzo 1976), p. 113-126.
- M. SOMALVICO, *Intelligenza artificiale*, Scienza & Vita nuova - Hewlett-Packard, Milano 1987.
- M. SOMALVICO, [Intelligenza artificiale](#), in: Enciclopedia italiana di scienze lettere ed arti, Appendice V, Roma, Istituto della Enciclopedia Italiana, 1991, pp. 735-738.
- F. AMIGONI, V. SCHIAFFONATI, M. SOMALVICO, [Intelligenza artificiale](#), in Enciclopedia della scienza e della tecnica, Roma, Istituto della Enciclopedia Italiana, 2008.

APPROCCIO CONNESSIONISTA STORICO – RETI NEURALI ARTIFICIALI

- D. PARISI, *Intervista sulle reti neurali. Cervello e macchine intelligenti*, Il mulino, Bologna 1989.
- R. HECHT-NIELSEN, *Neurocomputing*, Addison-Wesley, Reading 1990.
- D. FLOREANO, C. MATTIUSI, *Manuale sulle reti neurali*, Il mulino, Bologna 2002 (seconda edizione).

PERCETTRONI

- W. MCCULLOCH, W. PITTS, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, "Bulletin of Mathematical Biophysics" vol. 5, 1943.
- F. ROSENBLATT, *The perceptron: A probabilistic model for information storage and organization in the brain*, "Psychological Review" 65(6), 1958.
- F. ROSENBLATT, *Principles of Neurodynamics*, Spartan Books, New York 1962.
- M.L. MINSKY, S.A. PAPERT, *Perceptrons*, MIT Press, Cambridge 1969.
- [Percettrone](#), in Enciclopedia della Scienza e della Tecnica (2008), Treccani.it
- [Perceptron](#), in en.wikipedia.
- E. GOMEDE, [Bias in Perceptron: Unveiling the Impact of Preconceived Notions](#), Medium.com, 10 luglio 2023.
- C. DAVI, [Why Perceptron Neurons Need Bias Input?](#), Medium.com, 7 marzo 2020.

RETI CONVOLUZIONALI (CNN)

KUNIIKO FUKUSHIMA, [Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition](#), in 'Biological Cybernetics', vol. 36, n. 4, 1980, pp. 193-202.

Y. LECUN, B. BOSER, J.S. DENKER, D. HENDERSON, R.E. HOWARD, W. HUBBARD, L.D. JACKEL, [Backpropagation Applied to Handwritten Zip Code Recognition](#), Neural Computation vol. 1, n. 4, 1989.

Y. LECUN, B. BOSER, J.S. DENKER, D. HENDERSON, R.E. HOWARD, W. HUBBARD, L.D. JACKEL, [Handwritten Digit Recognition with a Back-Propagation Network](#), in TOURETZKY D. (curatore), *Advances in Neural Information Processing Systems* (NIPS 1989), Denver, CO. Vol. 2. Morgan Kaufmann 1990.

P. Y. SIMARD, D. STEINKRAUS, J.C. PLATT, [Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis](#), Proceedings of International Symposium on Circuits and Systems (ISCAS), 2003.

Y. LECUN, K. KAVUKCUOGLU, C. FARABET, [Convolutional Networks and Applications in Vision](#), Proceedings of International Symposium on Circuits and Systems (ISCAS), 2010.

U. KARN, [An Intuitive Explanation of Convolutional Neural Networks](#), in kudnuggets.com, 2016.

H. S. Chatterjee, [A Basic Introduction to Convolutional Neural Network](#), in medium.com, 2019.

A. UMER, [Understanding Convolutional Neural Networks: A Beginner's Journey into the Architecture](#), in medium.com, 2023.

WORD EMBEDDING

T. MIKOLOV, K. CHEN, G. CORRADO, J. DEAN, [Efficient Estimation of Word Representations in Vector Space](#), 2013, in arXiv:1301.3781

T. MIKOLOV, Q.V. LE, I. SUTSKEVER, [Exploiting Similarities among Languages for Machine Translation](#), 2013, in arXiv:1309.4168

J. PENNINGTON, R. SOCHER, C.D. MANNING, [GloVe: Global Vectors for Word Representation](#), in <https://nlp.stanford.edu/projects/glove/>
[Introduction to word embedding](#), in <https://www.alpha-quantum.com/blog>

RETI RICORRENTI (RNN) E TRANSFORMERS

S. HOCHREITER, J. SCHMIDHUBER, [Long short-term memory](#), Neural Computation 9(8), 1997.

A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A.N. GOMEZ, L. KAISER, I. POLOSUKHIN, [Attention Is All You Need](#), Advances in Neural Information Processing Systems 30, NY 2017.

J. USZKOREIT, [Transformer. A novel neural network architecture for language understanding](#), Google Research blog post, August 31, 2017.

S. RASCHKA, [Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch](#), blog post, February 9, 2023.

Z. CHEN, *How does ChatGPT work?* [Session 1](#), [session 2](#), [session 3](#), [session 4](#) [PowerPoint slides]. Constellate, ITHAKA, 2024.

LLMs

- DEVLIN, CHANG, LEE, TOUTANOVA, [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#), Google, 2018, in arXiv:1810.04805
- A. RADFORD, K. NARASIMHAN, T. SALIMANS, I. SUTSKEVER, [Improving Language Understanding by Generative Pre-Training](#), Open AI, 2018.
- E. M. BENDER E ALTRI, [On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?](#), in Proceedings of the 2021 ACM Conference on Fairness, Accountability and Transparency '21, Association for Computing Machinery, New York 2021, pp. 610-623.
- T.M. BROWN E ALTRI, [Language Models are Few-Shot Learners](#), Open AI, 2020, in arXiv:1810.04805
- B. LEMOINE, [Is LaMDA Sentient? - an Interview](#), in Medium.com, 2022.
- D. J. CHALMERS, [Could a Large Language Model be Conscious?](#), "Boston Review", 2023, in arXiv:2303.07103.
- M. BENASAYAG, A. PENNISI, *ChatGPT non pensa (e il cervello neppure)*, Jaca Book, Milano 2024.

IMPATTO SU LAVORO E CULTURA

- [IFLA Statement on Libraries and Artificial Intelligence](#), International Federation of Library Associations and Institutions, 2020.
- [Large, creative AI models will transform lives and labour markets](#), The Economist, 22 Aprile 2023.
- L. HYMAN, [It's Not the End of Work. It's the End of Boring Work](#), The New York Times, 22 Aprile 2023.
- F. CIOTTI, [Minerva e il pappagallo: IA generativa e modelli linguistici nel laboratorio dell'umanista digitale](#), «Testo e senso», n. 26, dicembre 2023.
- G. RONCAGLIA, *L'architetto e l'oracolo. Forme digitali del sapere da Wikipedia a ChatGPT*, Laterza, Bari 2023.
- G. RONCAGLIA, *Nuove forme di lettura fra rete e intelligenza artificiale*, in *Biblioteche e nuove forme di lettura*, Editrice bibliografica, Milano 2024.
- F. LAMBERTI, M. MELLIA, M. VIVARELLI (a cura di), *Biblioteche, lettura, intelligenza artificiale*, Editrice Bibliografica, Milano 2024.
- F. BUTERA, G. DE MICHELIS, *Intelligenza artificiale e Lavoro, una rivoluzione governabile*, Marsilio, Venezia 2024.
- Strategia Italiana per l'Intelligenza Artificiale 2024-2026 – Executive Summary*, Agenzia per l'Italia Digitale, 2024.
- C. CHESI, *Linguaggio e Intelligenza Artificiale*, in G. Graffi, S. Scalise, *Le lingue e il linguaggio*, Quarta edizione, Il Mulino, Bologna 2024.
- F. ROSSI, *Intelligenza artificiale. Come funziona e dove ci porta la tecnologia che sta trasformando il mondo*, Laterza, Bari 2024.
- M. Spitzer, *Intelligenza artificiale. Opportunità e rischi* [ecc..], Corbaccio, Milano 2024, traduzione di *Künstliche Intelligenz: Dem Menschen überlegen*, Droemer, Monaco 2023.

Fonti delle immagini

Slide 4 - Ritagli di sagome basate sulle immagini

- "Alan Turing (1912-1954) in 1936 at Princeton University", di pubblico dominio. Disponibile su: [https://en.wikipedia.org/wiki/Turing_Award#/media/File:Alan_Turing_\(1912-1954\)_in_1936_at_Princeton_University.jpg](https://en.wikipedia.org/wiki/Turing_Award#/media/File:Alan_Turing_(1912-1954)_in_1936_at_Princeton_University.jpg)
- 'John Searle speaking at Google 1'" di FranksValli, concesso in licenza sotto CC BY 4.0. Modificato da Lorenzo Babini. Disponibile su: https://en.wikipedia.org/wiki/John_Searle#/media/File.jpg

Slide 6 - Ritagli di sagome basate sulle immagini:

- "Yoshua Bengio 2019 cropped" di Maryse Boyce, concessa in licenza sotto CC BY-SA 4.0. Modificata da Lorenzo Babini. Disponibile su: https://en.wikipedia.org/wiki/Yoshua_Bengio#/media/File:Yoshua_Bengio_2019_cropped.jpg
- 'Yann LeCun - 2018 (cropped)'" di J  r  my Barande, concesso in licenza sotto CC BY-SA 2.0. Modificato da Lorenzo Babini. Disponibile su: [https://en.wikipedia.org/wiki/LeNet#/media/File:Yann_LeCun_-_2018_\(cropped\).jpg](https://en.wikipedia.org/wiki/LeNet#/media/File:Yann_LeCun_-_2018_(cropped).jpg)
- 'Geoffrey Hinton - Collision 2023 - Centre Stage RCZ 1307 (cropped)'" di Ramsey Cardy / Collision, concessa in licenza CC BY 2.0. Modificata da Lorenzo Babini. Disponibile su: [https://en.wikipedia.org/wiki/Geoffrey_Hinton#/media/File:Geoffrey_Hinton_-_Collision_2023_-_Centre_Stage_RCZ_1307_\(cropped\).jpg](https://en.wikipedia.org/wiki/Geoffrey_Hinton#/media/File:Geoffrey_Hinton_-_Collision_2023_-_Centre_Stage_RCZ_1307_(cropped).jpg)

Slide 8 - Immagine della rete neurale da:

- D. FLOREANO, C. MATTIUSI, *Manuale sulle reti neurali*, Il mulino, Bologna 2002 (seconda edizione), p. 17.

Slide 21 – Grafico della curva di apprendimento da:

- R. HECHT-NIELSEN, *Neurocomputing*, Addison-Wesley, Reading 1990.

Slides 31-39 – Rappresentazioni, modificate nei colori, da:

- D. JURAFSKY, J. H. MARTIN, *Speech and Language Processing* (Third edition), in web.stanford.edu/~jurafsky/slp3/, 2024.

with a little help from my friend: Alessandro Armato

Libraries can educate users about AI, and help them thrive in a society which uses AI more extensively.

IFLA - Statement on Libraries and AI



lorenzo.babini@unicatt.it