



Il volto nascosto della IA

***Modello connessionista
Pietre miliari ed esempi***

Ing. Claudio Sanft

Ordine degli Ingegneri della Provincia di Roma

Roma, 02 Dicembre 2025



Introduzione

IA: Un grande successo planetario

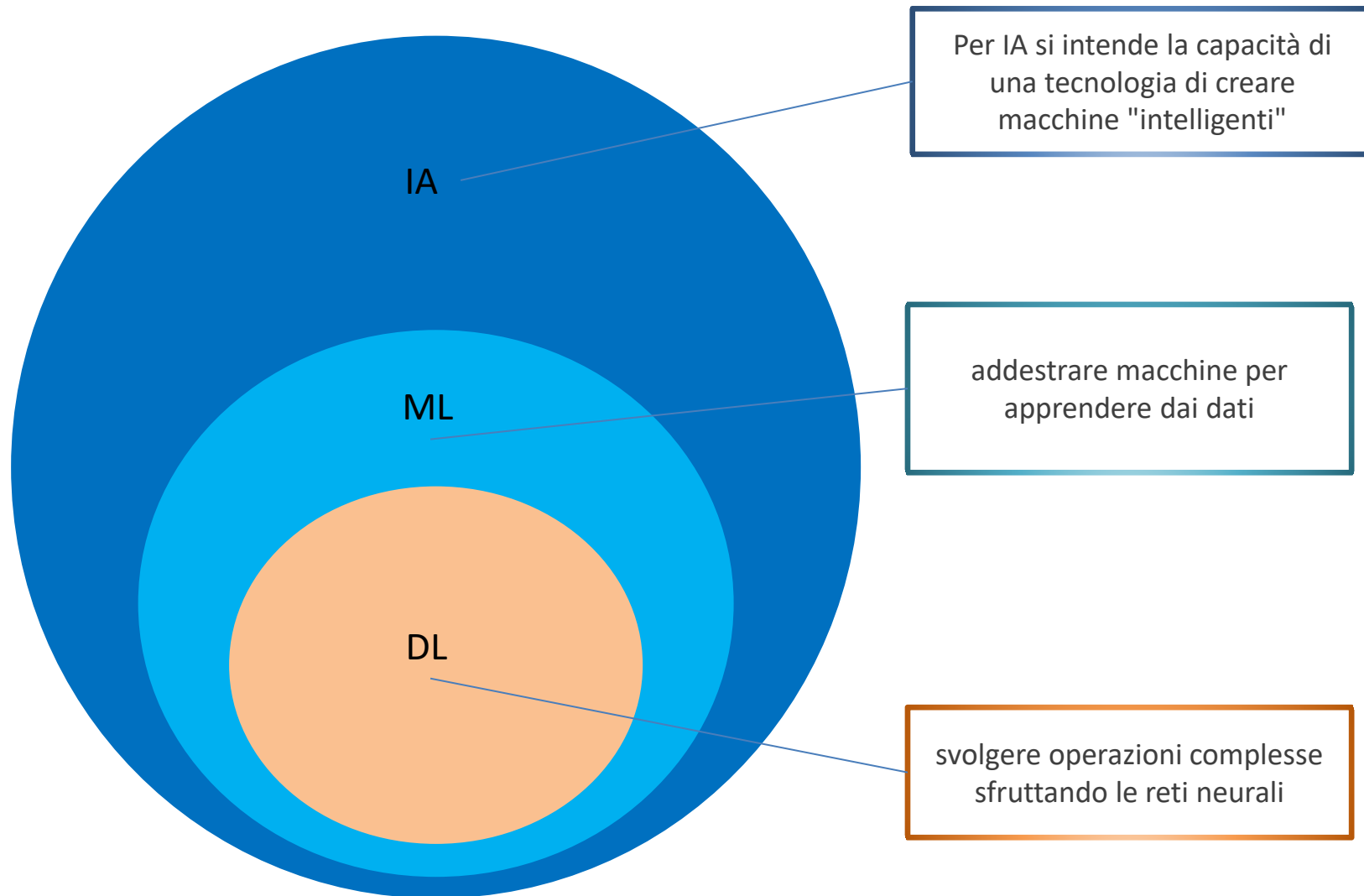
Asset oramai fondamentale in molti
processi decisionali

Direttrici specializzate di sviluppo per
sistemi IA verticalizzati

Riattraversamento, in intervallo ristretto di tempo, del cammino dello
scibile umano: dall'individuo onniscente allo specialista di settore



IA in sintesi





IA in sintesi

IA

Per IA si intende la capacità di una tecnologia di imitare l'intelligenza umana: apprendere, ragionare, risolvere problemi e prendere decisioni. Questa tecnologia permette di semplificare, e quasi mai sostituire, le funzioni di un cervello pensante appartenente ad un essere umano.

ML

Utilizza diversi metodi matematici e algoritmici, tra cui tecniche di apprendimento automatico. Questi algoritmi imparano da dati e sono in grado di migliorare le proprie prestazioni nel fare previsioni o prendere decisioni in modo autonomo, solo con un piccolo (talvolta anche senza nessun) intervento umano.

DL

Il Deep Learning è la tecnica più sofisticata di Machine Learning e si ispira alla struttura e al funzionamento del cervello umano. È costituito da **reti neurali artificiali profonde**, cioè migliaia di **nodi** interconnessi che imitano i **neuroni biologici**. Questi nodi consentono di creare **modelli di apprendimento** che producono dei risultati simili a quelli che creerebbe un essere umano: analizzare dati, comprendere testi e fotografie, fino a generare testi, audio, video ed immagini.



IA specializzate

Guru - Offre una piattaforma robusta che combina ricerca aziendale, wiki aziendale e funzionalità intranet.

IBM Watson - Una potente piattaforma AI che offre una suite di strumenti AI per vari settori, tra cui NLP, visione artificiale e machine learning.

Google Cloud AI - Fornisce una gamma di servizi AI, dalla speech-to-text all'analisi delle immagini, supportato dall'infrastruttura potente di Google.

Microsoft Azure AI - Una piattaforma completa che offre machine learning, servizi cognitivi e servizi per bot.

GPT-4 di OpenAI - Conosciuto per la sua avanzata comprensione e generazione del linguaggio naturale, utile per la creazione di contenuti e il servizio clienti.

Amazon Web Services (AWS) AI - Offre una varietà di servizi AI, tra cui modelli di machine learning e analisi dei dati guidata dall'AI.



IA specializzate

H2O.ai - Si concentra sulla democratizzazione dell'AI con una piattaforma open-source che offre strumenti di machine learning e analisi predittiva.

DataRobot - Una piattaforma AI aziendale che automatizza la creazione e l'implementazione di modelli di machine learning.

UiPath - Specializzato in RPA, permette alle aziende di automatizzare compiti ripetitivi con l'AI.

Salesforce Einstein - Integra capacità di AI direttamente in Salesforce CRM, fornendo informazioni e automazione per vendite e marketing.

Clarifai - Una piattaforma leader nella visione artificiale per attività di riconoscimento di immagini e video.

C3.ai - Un fornitore di software AI aziendale che offre strumenti per analisi predittiva e sviluppo di applicazioni AI.



Schema di alto livello di Chat GPT





Schema di alto livello di Chat GPT

Somiglianze tra AI e umani

•Esperienze passate come riferimento

- L'IA usa i dati su cui è stata addestrata.
- L'essere umano usa le proprie esperienze di vita, sociali e culturali.

•Pattern matching implicito

- L'IA cerca sequenze linguistiche, contesti e strutture che ha già visto.
- L'umano riconosce situazioni analoghe a quelle già vissute, anche inconsciamente.

•Contestualizzazione

- L'IA considera il contesto della conversazione.
- L'umano legge il contesto sociale, il tono della voce, la postura, ecc.

Differenze sostanziali

Flessibilità e creatività

- L'essere umano non si limita al pattern già visto: può immaginare scenari nuovi, inventare risposte mai sperimentate prima, fare salti intuitivi.
- L'IA lavora "statisticamente": generalizza bene, ma rimane vincolata a ciò che ha già appreso.

Componenti emotive

- L'umano integra empatia, emozioni e intenzioni proprie.
- L'IA riconosce segnali linguistici che "somigliano" a emozioni, ma non le vive.

Livelli di coscienza

- L'umano può riflettere sulla propria intenzione: "Perché ho detto questo? Qual era il mio vero intento?"
- L'IA non ha auto-riflessione autentica: applica regole probabilistiche e categorizzazioni senza consapevolezza.



Schema di alto livello di Chat GPT

L'utente afferma "Fa freddo qui"

Chat GPT

Parsing linguistico → riconosco la struttura: soggetto implicito + verbo + aggettivo + avverbiale di luogo.

Pattern matching → il modello ha visto migliaia di esempi simili nei dati di addestramento, molti dei quali si collegano al tema "temperatura ambiente".

Probabilità di intenti

70%: dichiarazione sul clima

20%: richiesta implicita di azione (chiudere finestra)

10%: significato figurato (emozionale/metaforico)

Risposta generata → tendenzialmente neutra ("Vuoi che accendiamo il riscaldamento?"), per coprire il caso più frequente senza fraintendere troppo



Schema di alto livello di Chat GPT

L'utente afferma "Fa freddo qui"

Essere umano

Costatazione neutra → semplice osservazione sul clima

Richiesta implicita → "Chiudi la finestra" oppure "Accendi il riscaldamento"

Segnale sociale → "Non sto a mio agio, potremmo cambiare stanza?"

Metafora/emozione → "Mi sento distante, freddo dentro"

Un umano, a seconda del contesto (luogo, tono di voce, relazione con l'interlocutore), decide quale interpretazione è più probabile.



Un pizzico di umanità! Le allucinazioni

Un sistema di AI può avere le "Allucinazioni"

Affermazioni plausibili ma ... false



Derive dell'apprendimento → il processo di apprendimento seleziona le risposte mediante lo schema premio/punizione. Il sistema di valutazione delle prestazioni potrebbe favorire la supposizione in luogo della dichiarazione di non conoscenza

Non lo so, e quindi ci provo → invece di: "Non lo so, Professore. Ieri anziché studiare sono andato in sala giochi 😊". Cioè: se ci provo, forse indovino. Se sbaglio prendo uno "0", che è il medesimo punteggio che prenderei dichiarando la mia ignoranza

Domanda con unica risposta corretta. Il sistema può:

- Rispondere correttamente
- Sbagliare
- Astenersi

L'astensione è qualificata come "Umiltà". Modulandone il peso, si contrastano le "Allucinazioni"



Ma

Da dove viene tutto ciò?



Un po' di storia del modello connessionista

1943 - McCulloch & Pitts (Chicago University): proposta di un modello artificiale di neurone (combinatore lineare a soglia)

1949 – D.O. Hebb (McGill University, Montreal): pesi sinattici adattivi come meccanismo di apprendimento

1962 – F. Rosenblatt (Cornell University): Perceptron per riconoscimento e classificazione di forme

1969 – M. Minsky e S. Papert (MIT): individuazione dei limiti del Perceptron. Conseguenza: stasi di circa 20 anni

1985 – D.E. Rumelhart, G. Hinton, R.J. Williams: algoritmo EBP, che apre le porte all'apprendimento per reti multistrato. Rinascita dell'interesse per il connessionismo



Il Perceptron

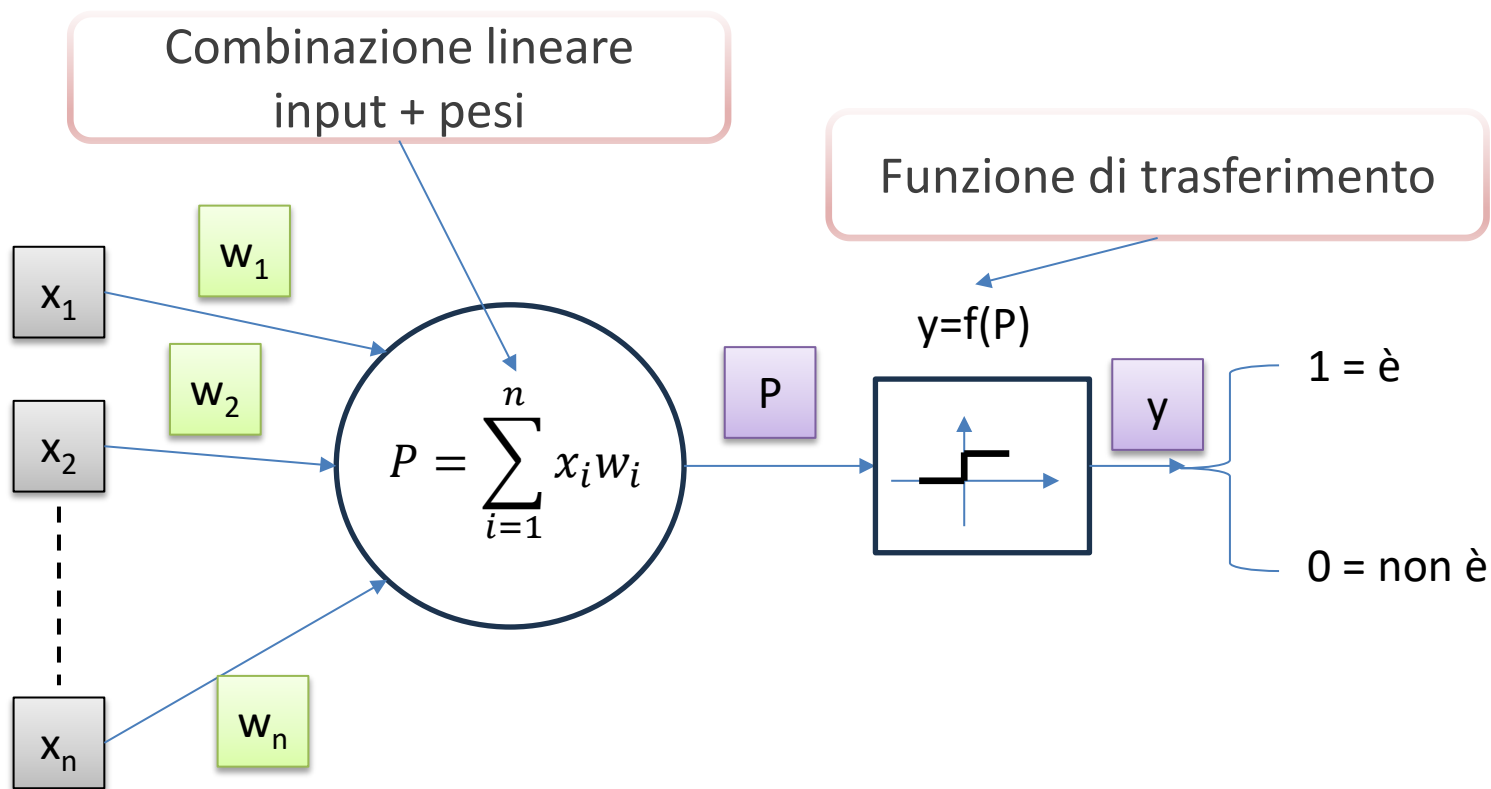
1958: F. Rosenblatt (Cornell University) Modello di neurone artificiale a pesi adattivi

Capacità di riconoscere se una generica forma "X" appartiene o meno ad una data classe "C"

Di fatto, la plasticità delle connessioni consente l'apprendimento dell'azione di riconoscimento delle forme "C" rispetto alle forme "non C"



Il Perceptron





Il Perceptron

Il Perceptron riceve una serie di esempi la cui risposta è nota.

Se la risposta elaborata e quella nota coincidono, si passa al successivo

Altrimenti, in accordo con "**1949** – D.O. Hebb": i pesi w_i subiscono una modifica controllata, in modo da apprendere l'esempio



Il Perceptron

L'apprendimento termina quando il Perceptron è in grado di classificare correttamente gli esempi.

A questo punto si passa alla parte di validazione, in cui si verifica la bontà dell'apprendimento mediante la somministrazione di altri esempi a risposta nota, per i quali ci si attende assenza di errore

Quando anche la validazione è completata positivamente, allora si può esercire il sistema



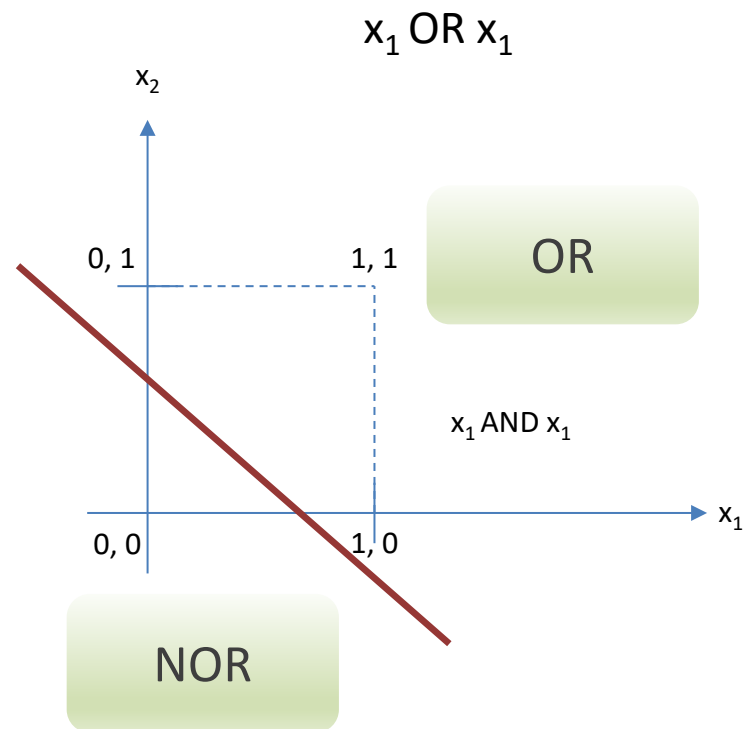
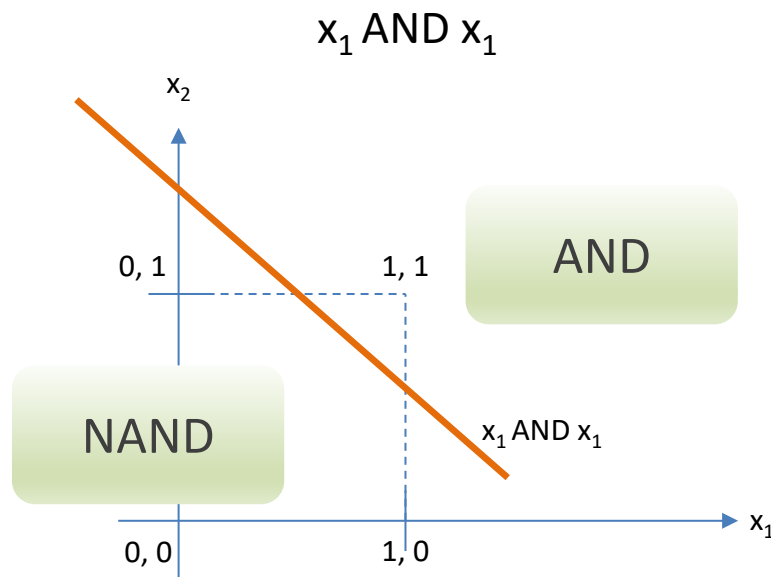
Limiti del Perceptron

Il Perceptron può solo operare suddivisioni lineari. Un determinato input può solo **appartenere** o **meno** ad un dato insieme **F**, con il vincolo per quest'ultimo che esso sia linearmente separabile da tutti gli elementi **non F**



Limiti del Perceptron - esempi

F = Operatori logici AND; OR

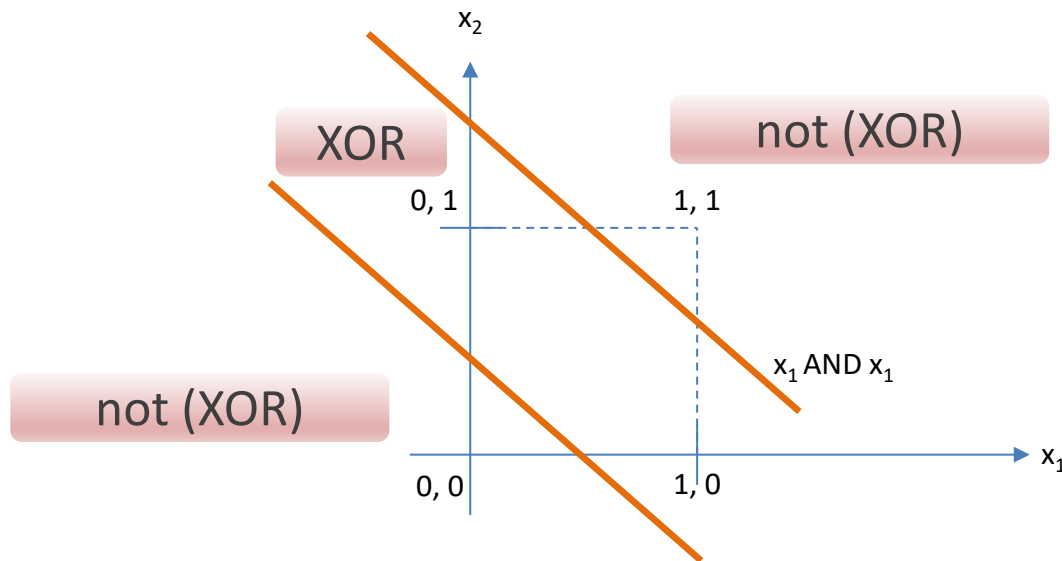




Limiti del Perceptron - esempi

$F = \text{Operatore logico XOR}$

$x_1 \text{ XOR } x_1$



Impossibile
separare
linearmente
elementi della
classe **F** da
elementi della
classe **non F**



Il Perceptron

Esempio pratico



Perceptron a più uscite

La generalizzazione consiste nell'adoperare il vettore di input su più uscite

Ciascuna uscita sarà dunque in grado di riconoscere forme di una determinata classe

In ciascuna coppia di classi, le medesime dovranno essere linearmente separate tra loro, data la natura stessa del perceptron



Perceptron a più uscite

In termini più sintetici, si attua il riconoscimento di una forma X tra le classi C_j , con $j \in [1, n]$

Ciò si verifica se:

$$\sum_{i=1}^n w_{ji} x_i > \sum_{i=1}^n w_{ki} x_i \quad \forall k \neq j$$

In fase di addestramento, per ogni esempio " k " e per ogni uscita " h ", si avrà una risposta che, confrontata con la risposta attesa, consente di misurare il delta tra le due

$$\Delta_{kh} = (y_{kh} - \bar{y}_{kh})$$



Perceptron a più uscite

Algoritmo di apprendimento di Widrow Hoff

Presentazione di pattern ingresso/uscita, e rilevazione degli errori

$$E_p = \frac{1}{2} \sum_{j=1}^n (y_{pj} - \bar{y}_{pj})^2$$
$$E = \sum E_p$$



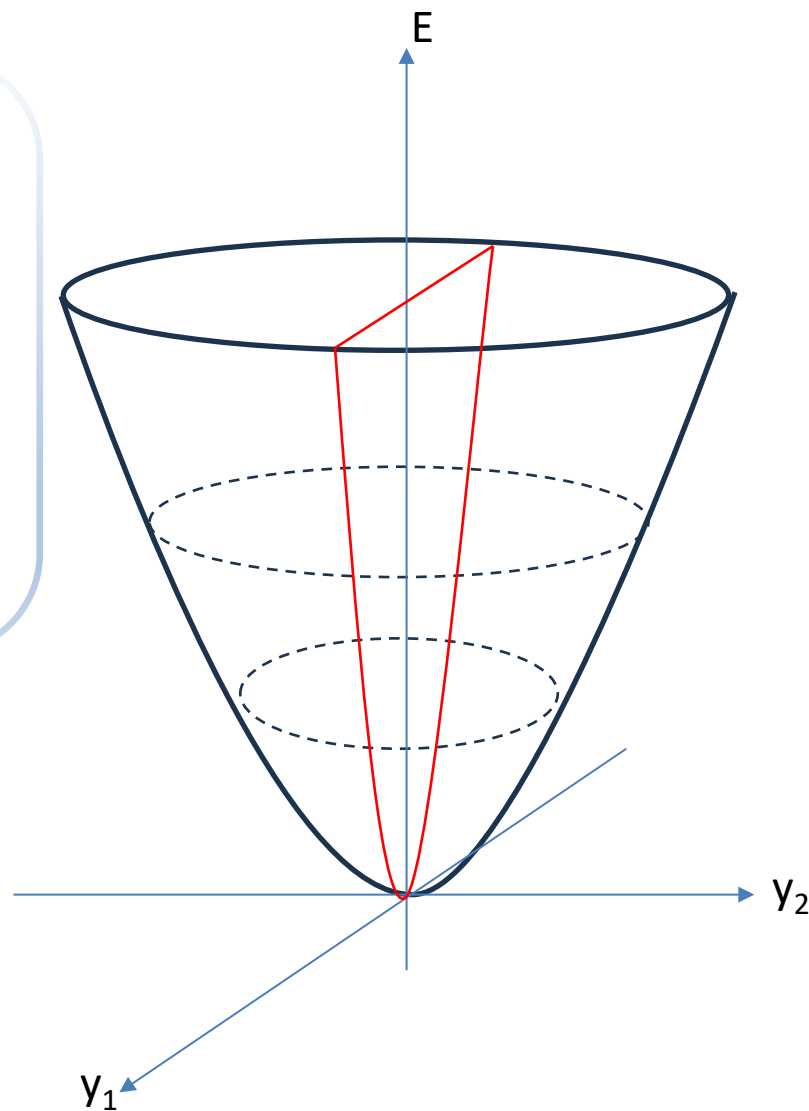
Perceptron a più uscite

Nel caso in cui:

$$f(\mathbf{P}) = \mathbf{P}$$

la minimizzazione dell'errore consiste nella ricerca del minimo dell'iper paraboloide.

La variazione dei pesi si traduce in un cammino lungo la superficie





Perceptron a più uscite

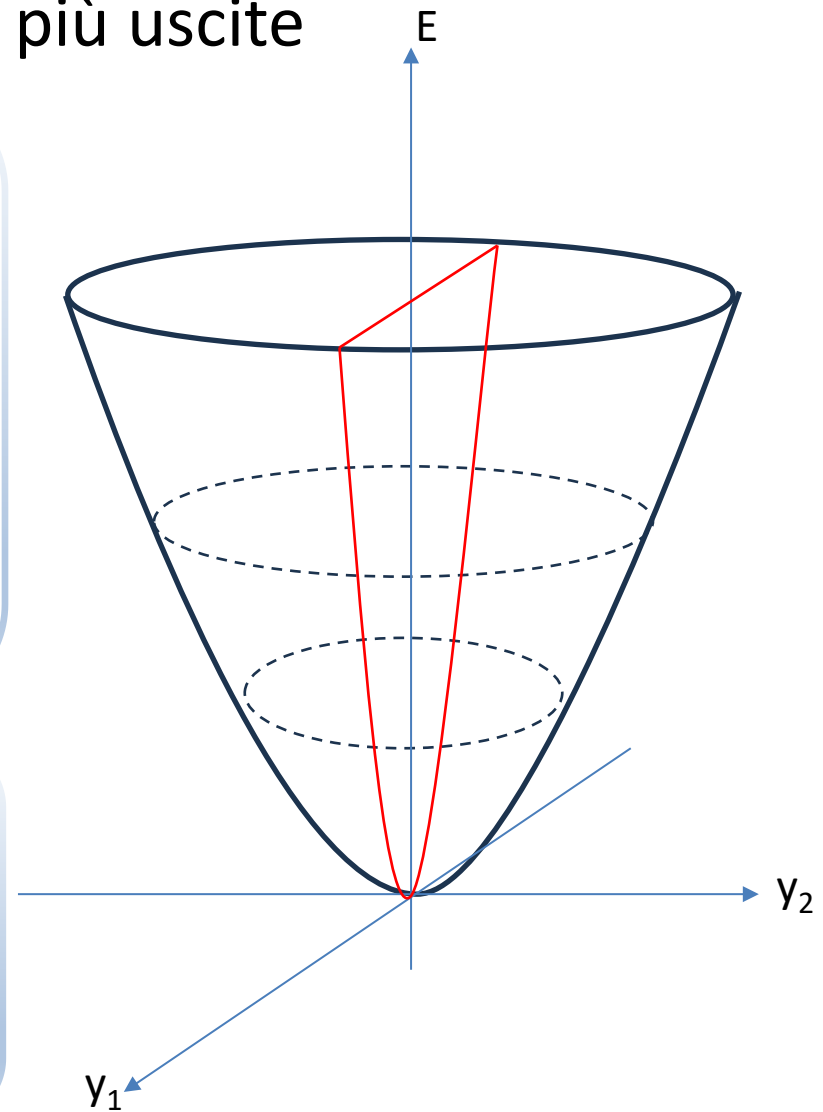
Le ascisse sono costituite dalle uscite y_i , mentre l'ordinata è l'errore medesimo.

Dunque, ogni intersezione del paraboloide con il generico piano E/y_j dà luogo alla parabola:

$$E_p = \frac{1}{2} \sum_{p=1}^q (y_{pj} - \bar{y}_p)^2$$

La cui derivata è:

$$\frac{\partial E_p}{\partial y_j} = \sum_{p=1}^q (y_{pj} - \bar{y}_p)$$





Perceptron a più uscite

Minimizzazione dell'errore

Formula di
aggiornamento dei
pesi:
regola **Delta estesa**



$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}$$

In cui $\eta \leq 1$ prende il nome di "learning rate", e ha l'effetto di modulare il rateo di apprendimento.

L'apprendimento si intende completato quando "E" è al di sotto di una determinata soglia "s"



Perceptron a più uscite

Calcolo della derivata parziale
(Per semplicità, rispetto al singolo E_p)

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial y_j} \frac{\partial y_j}{\partial P_j} \frac{\partial P_j}{\partial w_{ji}}$$

E tali derivate sono tutte note. Quindi si ottiene:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = -\eta \Delta_{kj} f'(P_j) x_i$$



Perceptron a più uscite

Esempio pratico



Reti neurali a più strati interni

!!!Grande problema!!!

Widrow Hoff presuppone la
disponibilità dell'errore
calcolato

$$\Delta w_{jr} = -\eta \frac{\partial \mathbf{E}}{\partial w_{jr}}$$

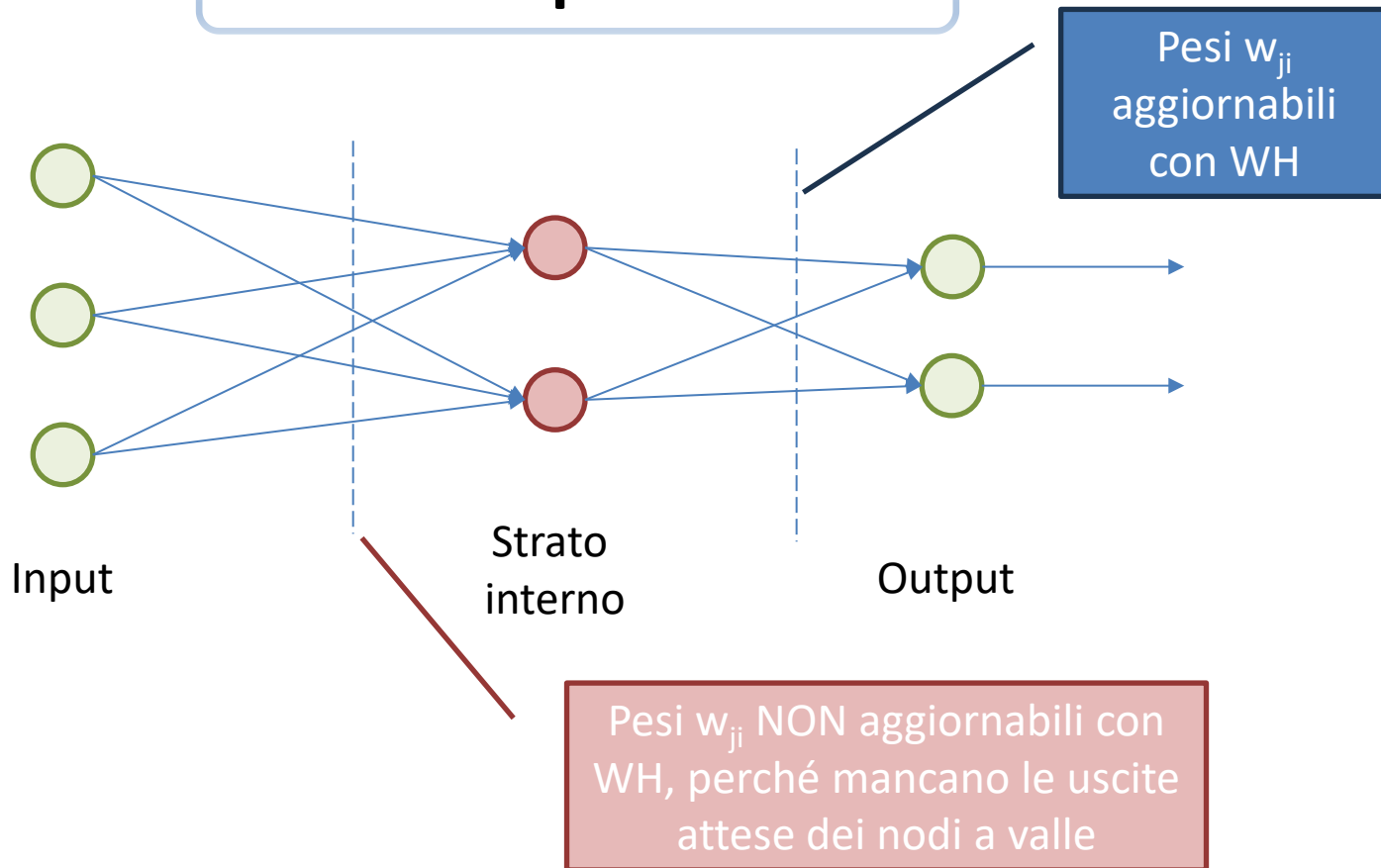
In una rete con almeno
uno strato interno,
occorrerebbe
conoscere i valori attesi
per ciascun nodo

Ma questo è impossibile, dato
che l'unico dato noto,
nell'addestramento, è il
risultato finale atteso!!!



Reti neurali a più strati interni

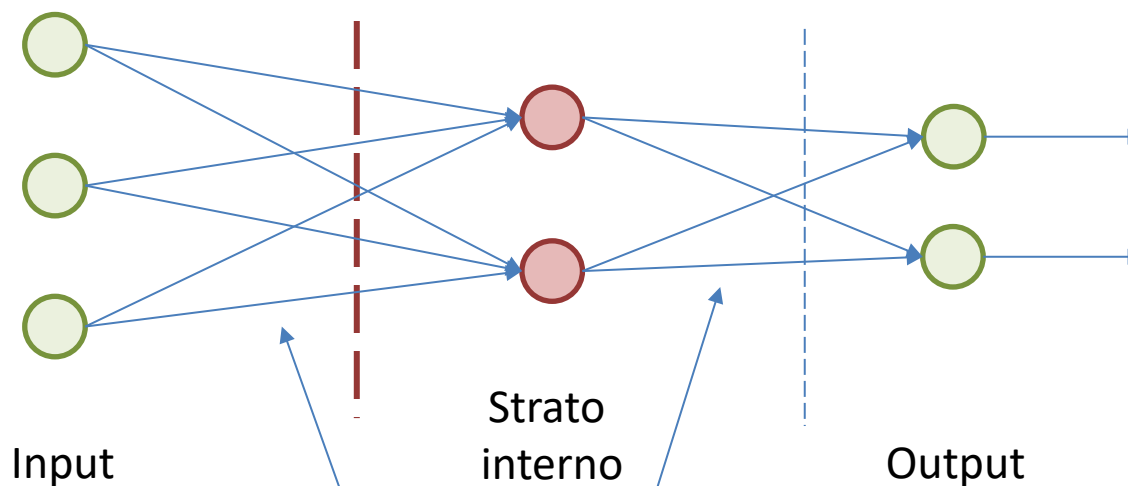
!!!Grande problema!!!





Reti neurali a più strati interni

!!!Grande problema!!!



$$\Delta w_{jr} = -\eta \Delta_j^{(k)} f'(P_j) x_r$$

$$\Delta_j^{(k)} = \left(\overline{x_i^{(k)}} - x_i \right)$$



Reti neurali a più strati interni

!!!Tuttavia si osserva che!!!

Ogni unità intermedia
è connessa all'uscita

Dunque posto E_k l'errore al
passo "k", si risale al valore
ignoto dello strato interno

$$\frac{\partial E_k}{\partial x_i} = \sum_{j=1}^n \frac{\partial E_k}{\partial y_j} \frac{\partial y_j}{\partial P_j} \frac{\partial P_j}{\partial x_i}$$

$$\frac{\partial E_k}{\partial x_i} = \sum_{j=1}^n \Delta_j^{(k)} f'(P_j) w_{ji}$$



Reti neurali a più strati interni

!!!Soluzione finale!!!

Riassemblando il tutto si ottiene la formula di aggiornamento dei pesi per lo strato interno

$$\Delta w_{jr} = -\eta \frac{\partial E_k}{\partial x_i} f'(P_j) x_r$$

Questo apparentemente semplice calcolo diede nuovo impulso allo sviluppo del modello connessionista



Reti neurali a più strati interni

Algoritmo EBP apre le porte
all'addestramento di reti con "n" strati

Il resto, lo ha reso
possibile l'immensa
potenza di calcolo
divenuta
progressivamente
disponibile

Ore di addestramento per
reti con due strati interni
all'inizio degli anni '90 su
IBM RISC 6000

Pochi secondi su un
moderno laptop

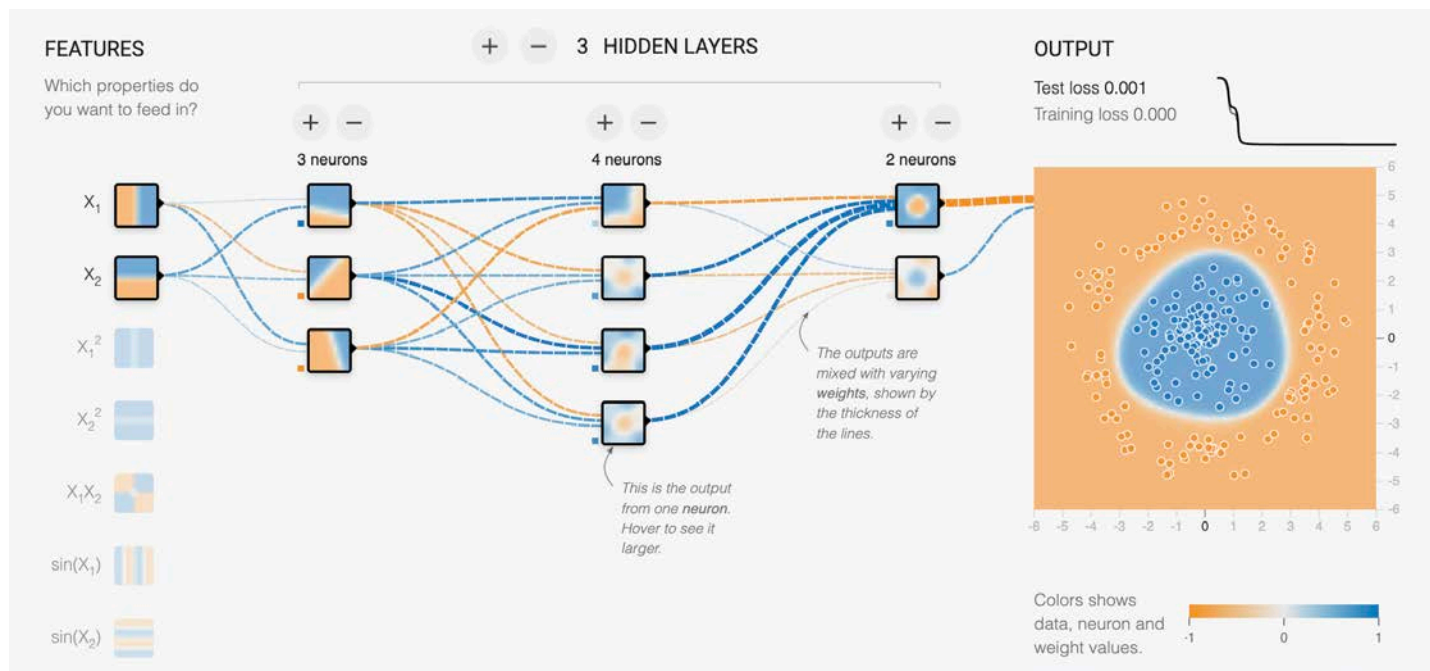
E si continua



Reti neurali a più strati interni

Un esempio immediatamente fruibile

<https://playground.tensorflow.org/>





Funzioni di trasferimento

Elemento di calcolo di primaria importanza

Diverse funzioni,
legate al tipo di rete
da addestrare

Devono avere
caratteristiche di
derivabilità

Il calcolo della derivata dovrebbe
essere comodo, al fine di
risparmiare tempo di calcolo



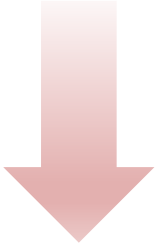
Funzioni di trasferimento

Vi sono moltissime funzioni di trasferimento

$$y = f(P) \text{ con } P_j = \sum_{i=1}^{n_j} w_{ji} x_i$$

Fondamentale la
semplificazione del
calcolo sia della
funzione che della
sua derivata prima

Ciascuna con vantaggi
relativi all'applicazione in
cui è impiegata



- Lineare
- Sigmoide (semplice/parametrica)
- Tangente iperbolica
- Rampa
- Gaussiana
- Gradino
- ReLU: Rectified Linear Unit



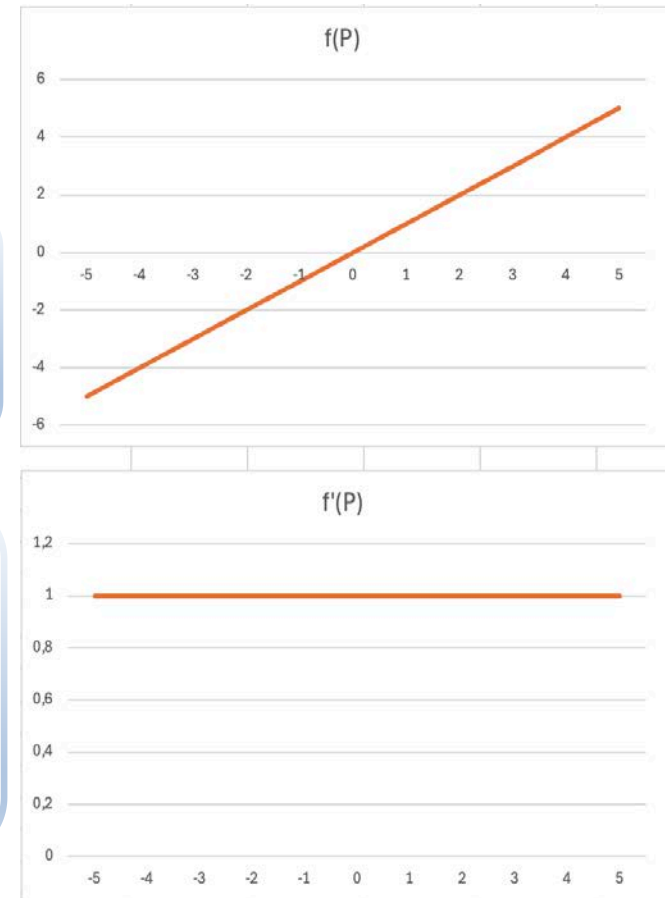
Funzioni di trasferimento

Funzione lineare

$$f(P) = P$$
$$f'(P) = 1$$

Quando si vuole che un nodo fornisca il suo output senza applicare alcuna soglia

Tipicamente si usa nei nodi di uscita.
Non si usa negli strati interni, per evitare di avere una semplice soluzione linearmente separabile



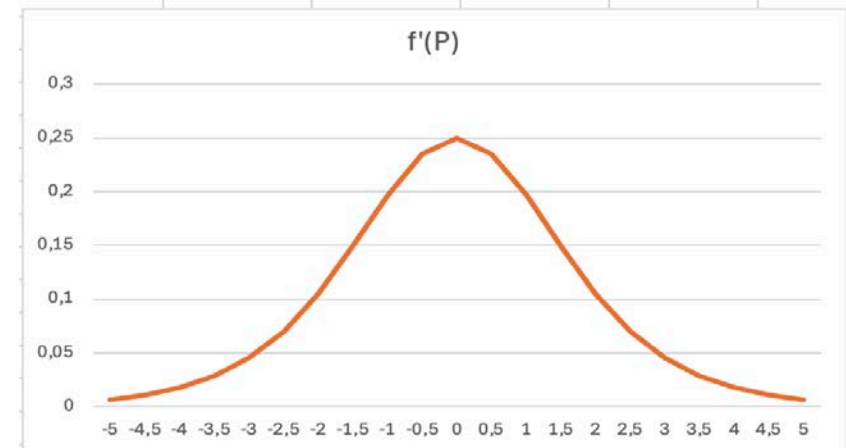
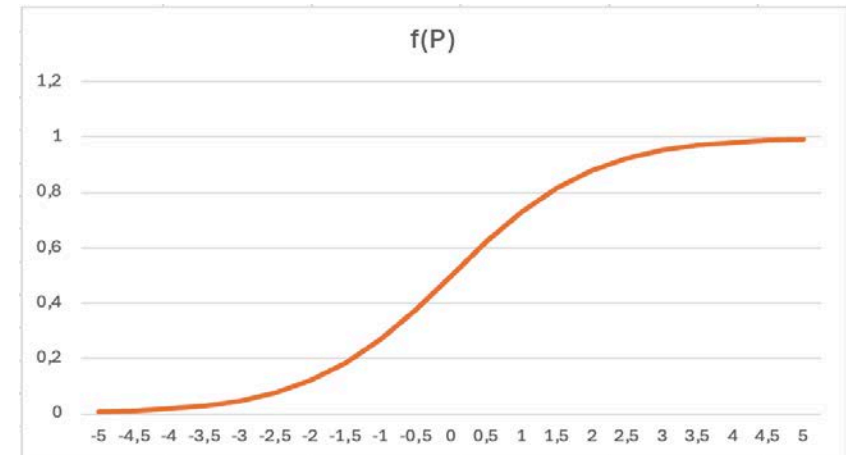


Funzioni di trasferimento

Sigmoide (semplice)

$$f(P) = \frac{1}{1 + e^{-P}}$$
$$f'(P) = f(P)[1 - f(P)]$$

$f(P)$	$f'(P)$
$\lim_{p \rightarrow -\infty} f(P) = 0$	$\lim_{p \rightarrow -\infty} f'(P) = 0$
$f(0) = 0,5$	$f'(0) = 0,25$
$\lim_{p \rightarrow +\infty} f(P) = 1$	$\lim_{p \rightarrow +\infty} f'(P) = 0$





Funzioni di trasferimento

Sigmoide (semplice)

$$f(P) = \frac{1}{1 + e^{-P}}$$
$$f'(P) = f(P)[1 - f(P)]$$

Viene spesso scelta per la semplicità nel calcolo della derivata prima.

Ciò semplifica i calcoli per adeguamento pesi con metodo EBP
Varia tra 0 e 1, con soglia a 0,5

Problema: per input fortemente negativi, può portare ad un aggiornamento quasi nullo dei pesi, rendendo il nodo "morto" (i.e.: statico rispetto al EBP)



Funzioni di trasferimento

Sigmoide (parametrica)

$$f(P) = \frac{a}{1 + e^{-kP}}$$
$$f'(P) = \frac{k}{a} f(P) [a - f(P)]$$

Aggiunge la possibilità di controllare soglia e inclinazione della curva

$f(P)$	$f'(P)$
$\lim_{p \rightarrow -\infty} f(P) = 0$	$\lim_{p \rightarrow -\infty} f'(P) = 0$
$f(0) = 0,5a$	$f'(0) = 0,25ka$
$\lim_{p \rightarrow +\infty} f(P) = a$	$\lim_{p \rightarrow +\infty} f'(P) = 0$

Problema: per input fortemente negativi, può portare ad un aggiornamento quasi nullo dei pesi, rendendo il nodo "morto" (i.e.: statico rispetto al EBP)

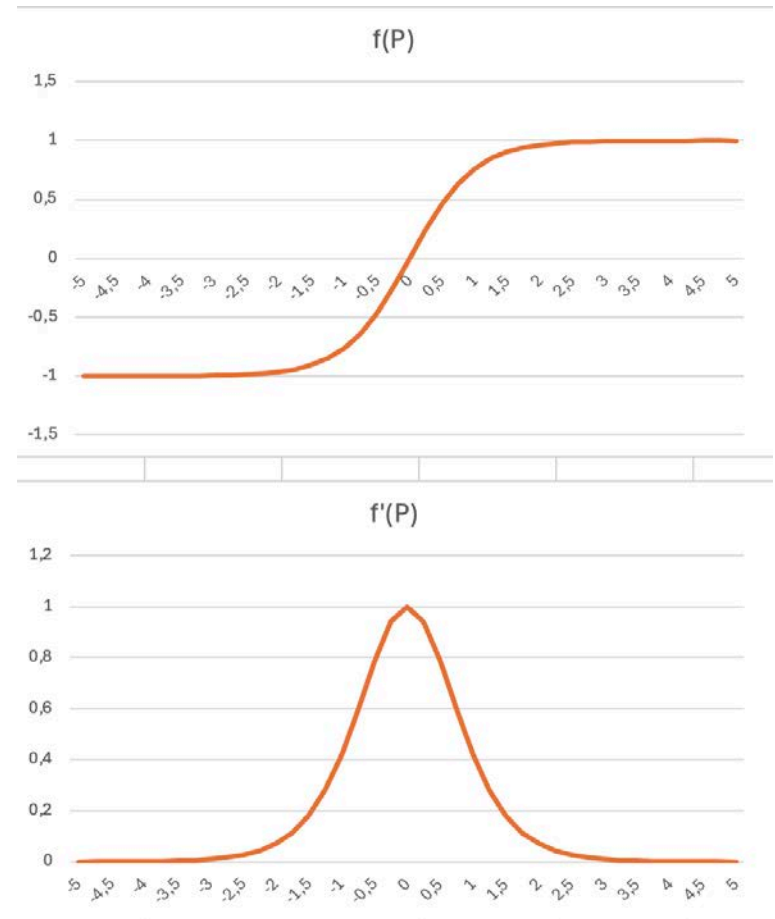


Funzioni di trasferimento

Tangente iperbolica (parametro "a")

$$f(P) = \tanh(aP)$$
$$f'(P) = a[1 - f^2(P)]$$

$f(P)$	$f'(P)$
$\lim_{p \rightarrow -\infty} f(P) = -a$	$\lim_{p \rightarrow -\infty} f'(P) = 0$
$f(0) = 0$	$f'(0) = a$
$\lim_{p \rightarrow +\infty} f(P) = a$	$\lim_{p \rightarrow +\infty} f'(P) = 0$





Funzioni di trasferimento

Tangente iperbolica (parametro "a")

$$f(P) = \tanh(aP)$$
$$f'(P) = a[1 - f^2(P)]$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Soglia e inclinazione sono controllate da "a"

Varia tra "-a" e "a", e ha soglia pari a 0. "Punisce" gli input negativi con un aggiornamento altrettanto negativo. Evita il problema dei "nodi morti" che potrebbero capitare con la sigmoide.

Robusta rispetto ai nodi morti



Funzioni di trasferimento

Funzione rampa

$$f(P) = aP + b \begin{cases} P < P_{min} \\ P \in [P_{min}; P_{max}] \\ P > P_{max} \end{cases}$$
$$f'(P) = \begin{cases} a \quad \forall P \in]P_{min}; P_{max}[\\ 0 \quad \forall P: \{P < P_{min}; P > P_{max}\} \end{cases}$$

Come la lineare, con l'obiettivo di escludere valori esterni ad un intervallo di controllo. Può essere vista anche come una versione "ruvida" della sigmoide, nel senso che ha dei punti certi di taglio in P_{min} e P_{max}



Funzioni di trasferimento

Funzione ReLU (Rectified Linear Unit)

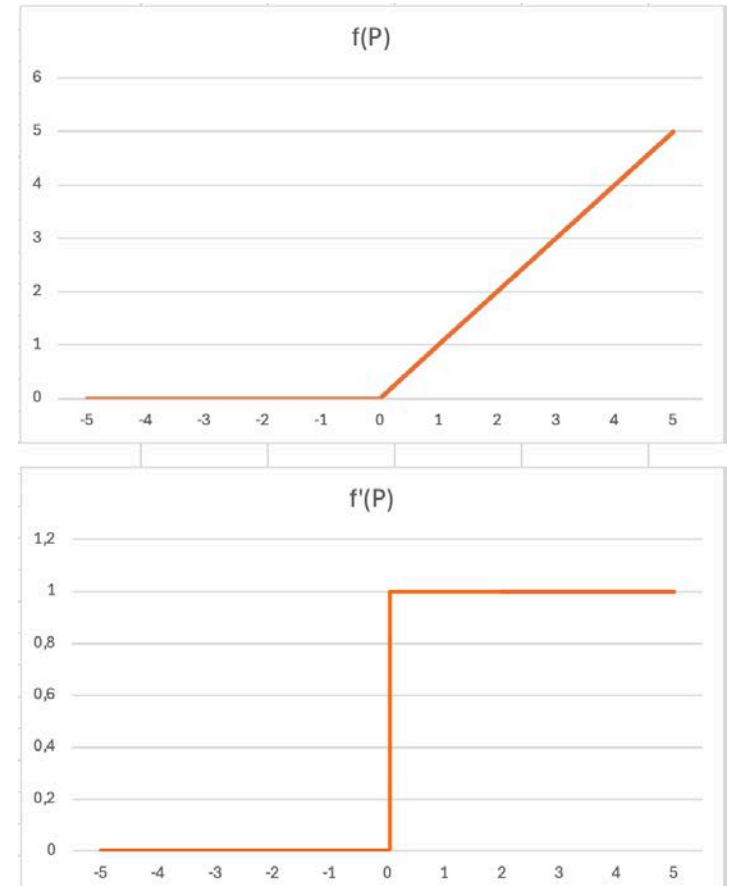
Caso particolare di rampa, con:

$$P_{min} = 0$$

$$P_{max} \rightarrow +\infty$$

$$a = 1; b = 0$$

Viene adoperata per reti a convoluzione, impiegate per riconoscimento immagini



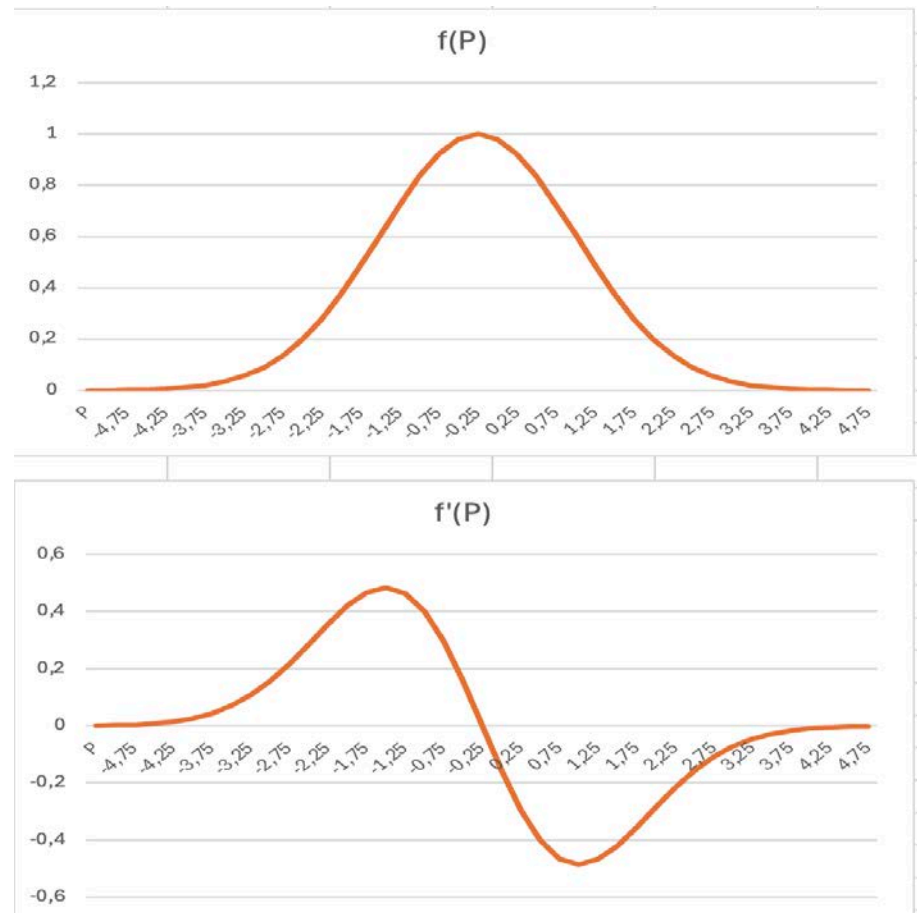


Funzioni di trasferimento

Gaussiana

$$f(P) = e^{-\frac{2P^2}{\sigma^2}}$$
$$f'(P) = -\frac{4P}{\sigma^2} f(P)$$

Ampiezza della campana
controllata da " σ "
punti di flesso in $P = \pm\sigma$





Funzioni di trasferimento

Gaussiana

$$f(P) = e^{-\frac{2P^2}{\sigma^2}}$$
$$f'(P) = -\frac{4P}{\sigma^2} f(P)$$

Caratteristica forma a campana: è una funzione pari, mentre la sua derivata è dispari.

Si annulla per input lontani dal centro

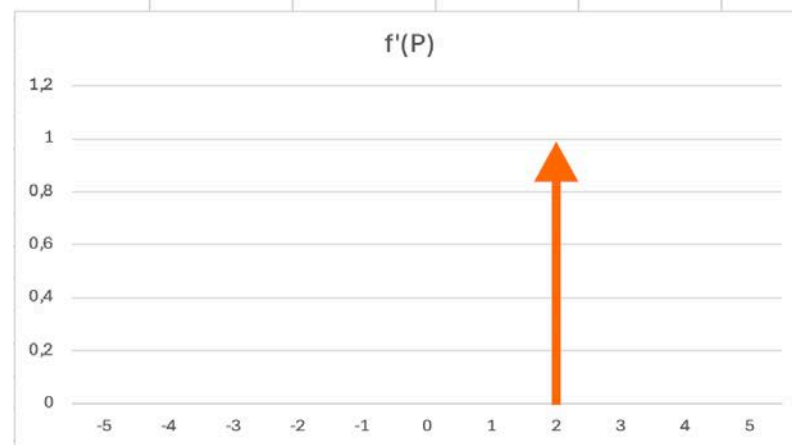
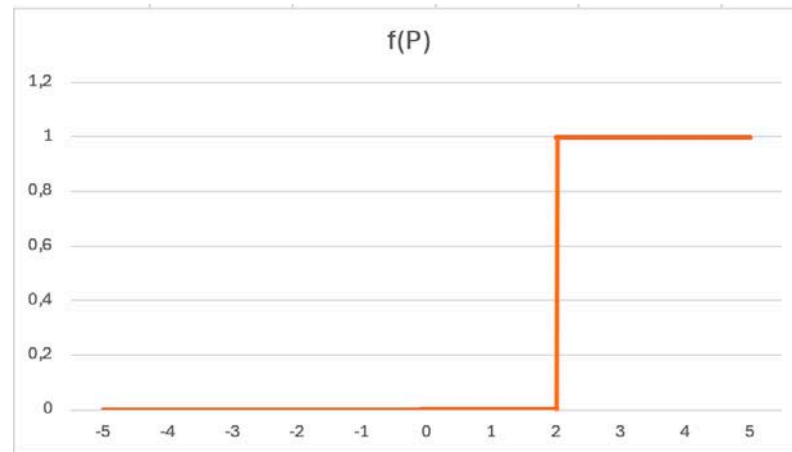
Favorisce gli input centrati intorno ad un punto.
Adoperata nelle reti RBF (Radial Basis Function),
utili ad esempio nell'individuazione dei valori futuri
di una traiettoria



Funzioni di trasferimento

Funzione gradino

$$f(P) = \begin{cases} 1: \forall P \geq T \\ 0: \forall P < T \end{cases}$$
$$f'(P) = \delta(P - T)$$





Funzioni di trasferimento

Funzione gradino

A volte occorre un'uscita che renda conto drasticamente del superamento di una determinata soglia "T".

Tuttavia, tale funzione non ha le caratteristiche di continuità necessarie per l'addestramento (la sua derivata è la funzione delta di Dirac nel punto $P=T$)

Può essere impiegata nei problemi di separazione lineare (i.e.: perceptron)



GRAZIE PER L'ATTENZIONE

ing. Claudio Sanft
claudio.sanft@delta-lambda.com
+39 348 320 2837