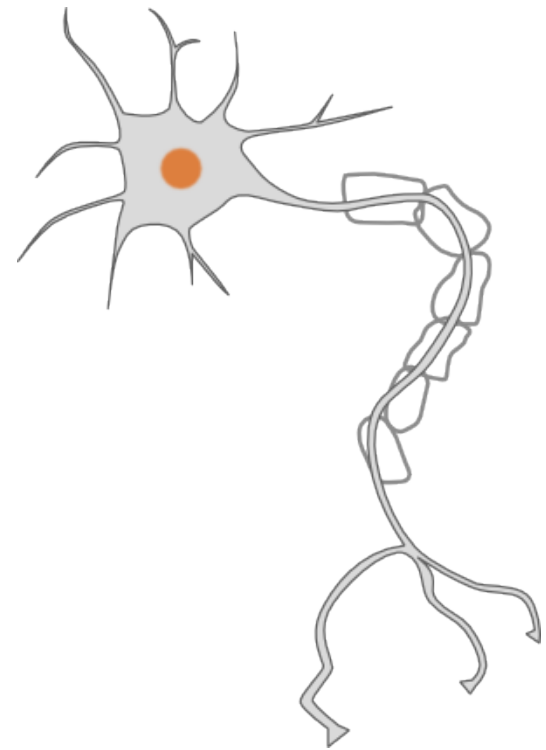


# Reti neurali

una breve introduzione

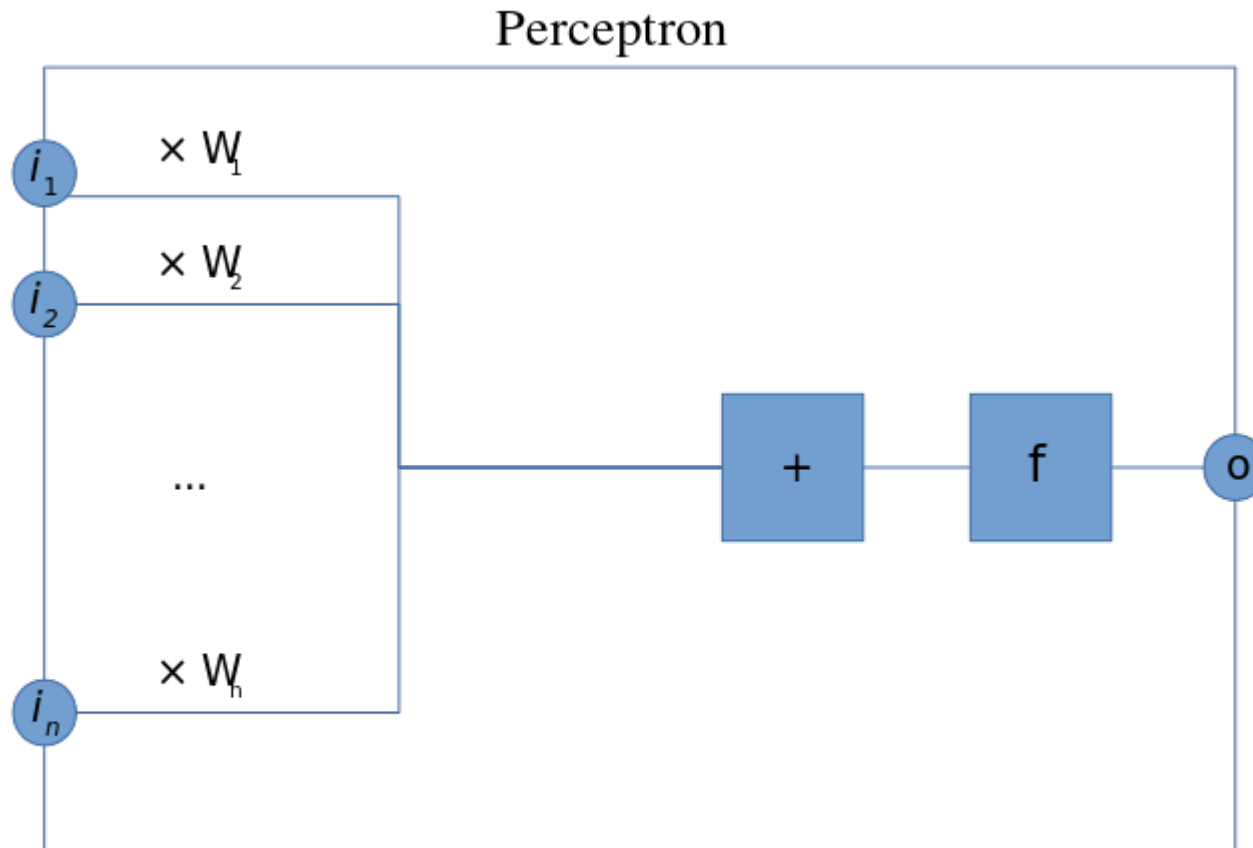
# Idea

- Con un solo neurone non posso fare molto...
- Con molti neuroni connessi posso ottenere un **sistema** di elaborazione
- nasce una nuova **proprietà emergente** del sistema



# Perceptron

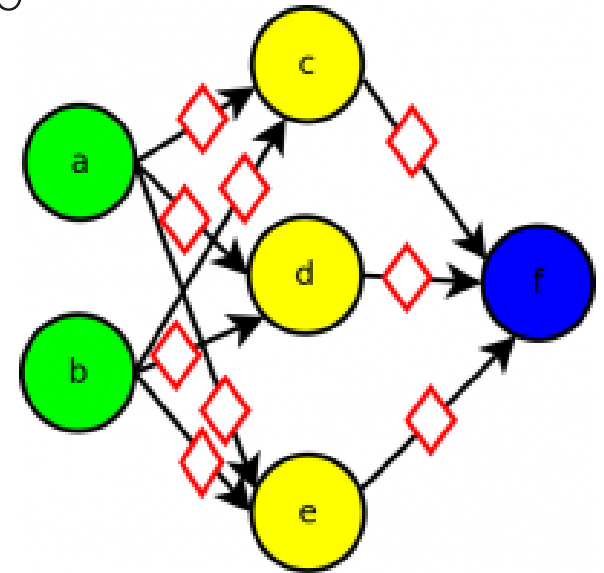
- input
- weight
- sum
- function
- output



$$o = f\left(\sum_{k=1}^n i_k \cdot W_k\right)$$

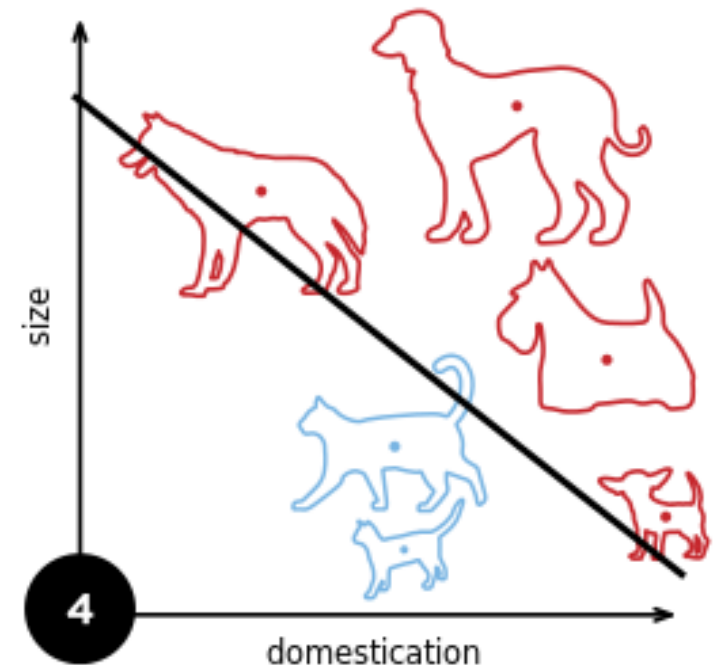
# Esempio

- sinapsi = collegamento con “peso”
- output = somma “pesata” di input
  - fase di addestramento = algoritmo iterativo che stabilisce i pesi “giusti”
  - fase di funzionamento = in output una funzione degli input
- La rete neurale **non** è un **cervello!**
- **L'addestramento** non è un vero **apprendimento!**



# Utilizzi pratici

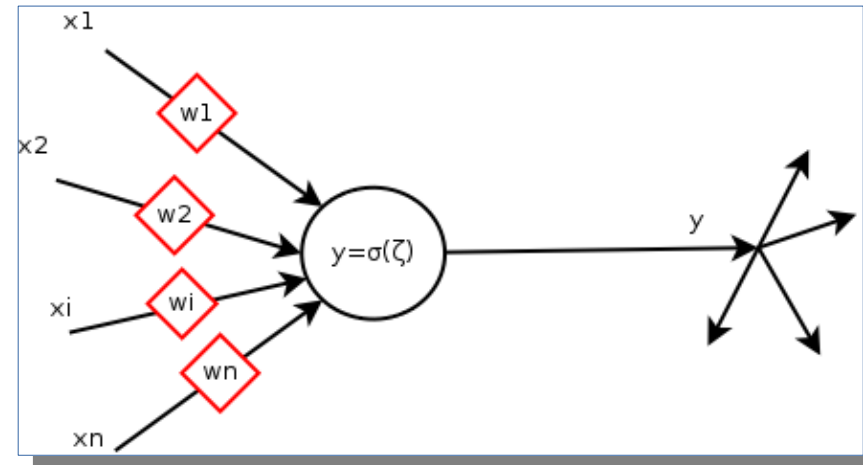
- Utili quando NON conoscono la funzione **analiticamente**.  
Conosco solo input e output desiderati
- Nel 1969 si dimostra che queste reti possono generare solo alcune funzioni (es: OR, AND, non XOR)
- Alcuni esempi:
  - Classificatori lineari: classificano i dati di input in **categorie linearmente separabili**



# Tre generazioni

- reti di Prima generazione
  - threshold perceptron (soglia attivaz.)
  - **multilayer**
- reti di Seconda generazione
  - **funzione di attivazione  $\sigma$** 
    - non lineare (es: sigmoid function)
  - conness. feedforward o recurrent
  - addestramento tramite algoritmo di **back-propagazione** dell'errore

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$



# Spiking Neural Network

- reti di Terza generazione

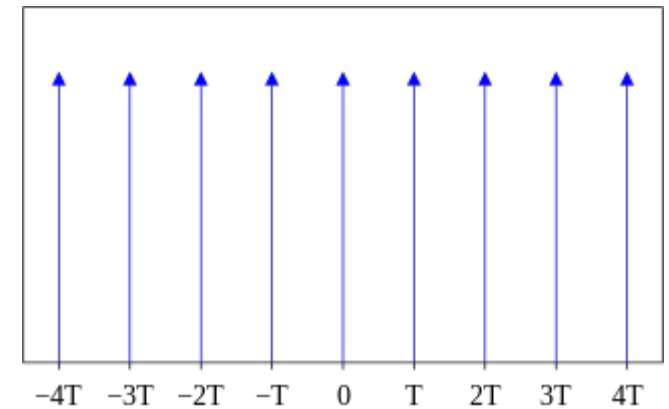
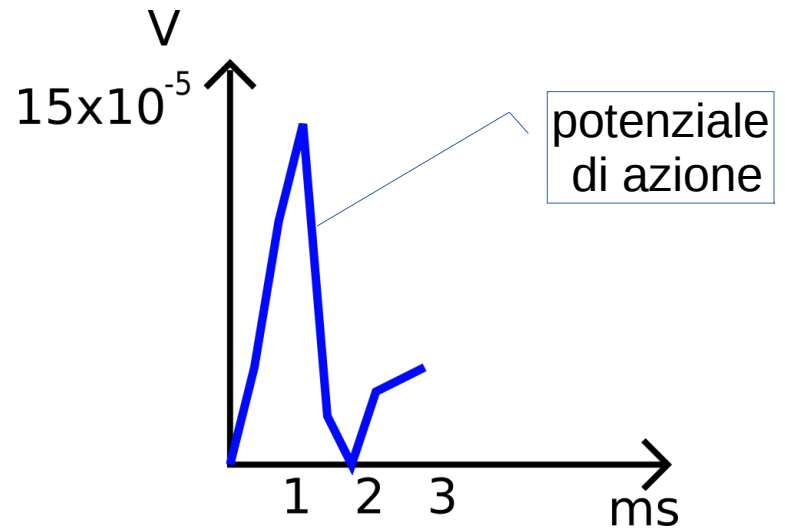
- Spiking Neural Network (**SNN**)

- Imitare il vero neurone

- impulsi di corrente a 100 Hz
- **tempo** di elaborazione non nullo
- plasticità delle sinapsi

- Modelli di neuroni

- Modelli di sinapsi



# Spiking Neural Network

- Come tradurre gli input esterni in impulsi di corrente?
- Come tradurre gli impulsi di corrente in un output comprensibile?
  - temporal coding: valore tradotto come intervallo tra impulsi
  - rate coding: valore tradotto come frequenza di impulsi
  - population coding...



# Modelli di neuroni

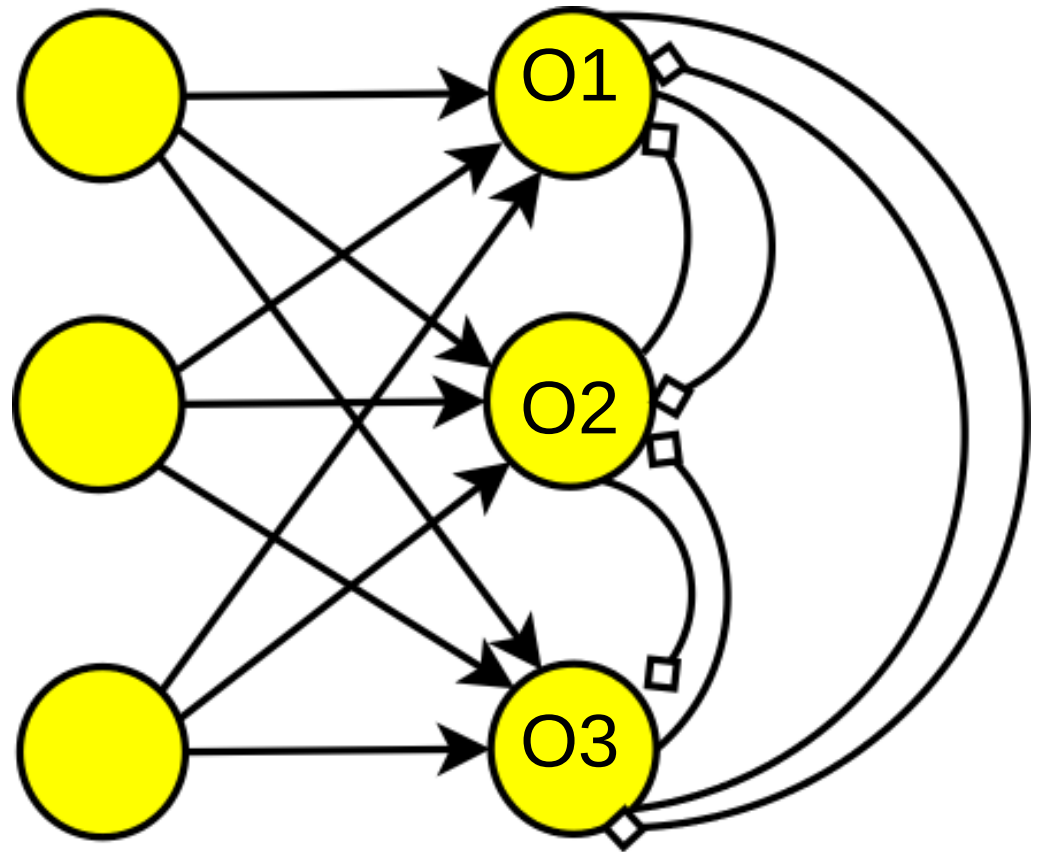
- Per le reti **SNN** sono stati sviluppati dei modelli matematici di neurone artificiale.
- Contengono **equazioni differenziali** (da risolvere durante la simulazione)
- Ogni **modello** è un compromesso tra realismo e semplificazione dei calcoli
  - integrate-and-fire : modello molto semplice
  - Hodgkin-Huxley: modello molto accurato
  - **Izhikevich**: una via di mezzo
  - altri: FitzHugh–Nagumo, Hindmarsh–Rose

# Modello di Izhikevich

- Modello flessibile grazie a quattro parametri ( $a, b, c, d$ )
- Descrive tre variabili ( $I, V, U$ )
  - corrente, tensione, periodo di recupero
- Si possono simulare diversi tipi di “veri” neuroni
  - Con una CPU @ 1GHz si può simulare una rete di  **$10^4$  neuroni**, con  $10^6$  sinapsi ad una risoluzione di 1 ms.
  - Purtroppo il cervello umano possiede ben  **$10^{11}$  neuroni** con  $10^{15}$  sinapsi (7 ordini di grandezza in più!)

# Connessioni sinaptiche (esempio)

- sinapsi eccitatoria
- ◊ sinapsi inibitoria
- One winner take all
  - se si attiva per primo il neurone O2 esso inibisce l'attivazione di O1 e O3
- entrambi i tipi (eccitat./inibit.) possono essere nel tempo "statiche" oppure "dinamiche"



# Modelli di sinapsi

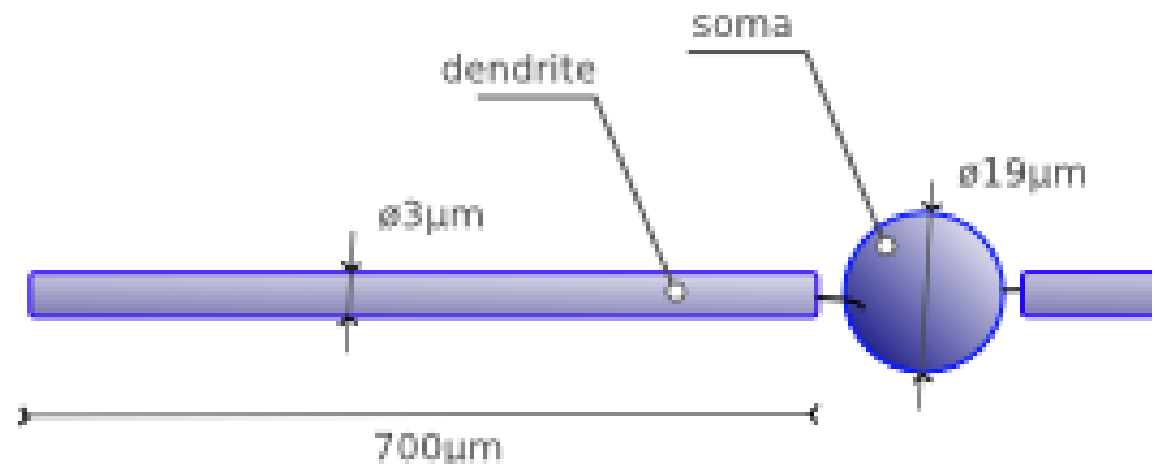
- Spike-Time Dependent Plasticity (STDP)
  - Sinapsi statiche vs dinamiche
  - Come nella realtà: La forza della sinapsi può dipendere da:
    - **l'ordine temporale** di attivazione di due neuroni
    - il tipo di sinapsi (**eccitatoria** o **inibitoria**)
  - La **consequenzialità** causale (e temporale) dello stato eccitato di due neuroni determina **potenziamento/depressione** di una sinapsi se è la sinapsi è di tipo eccitatoria/inibitoria.
- Può essere personalizzata tramite sei parametri ( $\tau^+$ ,  $\tau^-$ ,  $\alpha$ ,  $\mu^+$ ,  $\mu^-$ ,  $W_{\max}$ )

# Modelli di sinapsi

- LTP: Long Term Potentiation
- Ipotizzando che le nuove informazioni si memorizzano nei pesi sinaptici (non nel n. di neuroni)
  - Long Term Potentiation (LTP)
  - Long Term Depression (LTD) evita accrescimento infinito sinapsi
- **Meccanismi**
  - rafforzamento peso sinaptico dei neuroni vicini che si eccitano allo stesso momento (si imposta una finestra temporale)
  - somma **spaziale**: eccitazione causata da **due** stimoli contemporanei provenienti da **due** sinapsi adiacenti al neurone considerato
  - somma **temporale**: eccitazione causata da **due** stimoli consecutivi provenienti da **una** sola sinapsi

# Software

- **Software per reti neurali**
  - NeMo usa C++ (reti di terza generazione)
  - NEST usa Python (reti di terza generazione)
  - pyBrain usa Python (reti di seconda generazione)
- **Software per modelli cellulari**
  - EDLUT
  - Neural
  - Python Brain Simulator



# Copyright

- Author Fabio Proietti
- License Creative Commons 3.0
  - <http://creativecommons.org/licenses/by-sa/3.0/>