



Dipartimento
di Matematica
e Informatica

Corso di Laurea
in Informatica

Cervello e computer: bellezza e segreti dei bit di tutti i giorni



cybersecurity • intelligenza artificiale • realtà virtuale • Android

Ciclo di seminari di divulgazione informatica
in collaborazione con NOVA a.p.s.



Ferrara, 8–12 Giugno 2020



Lunedì 8 Giugno: Carlo Giannelli

Cyber security: istruzioni per l'uso – Principi di sicurezza informatica



Martedì 9 Giugno: Guido Sciavicco

Come imparano le macchine – Principi di intelligenza artificiale



Mercoledì 10 Giugno: Marco Alberti

Neuroni di bit – Reti neurali e applicazioni



Giovedì 11 Giugno: Antonino Casile

Informatica e percezione sensoriale – L'ultima frontiera della realtà virtuale



Venerdì 12 Giugno: M. Roma, G. Turri, L. Travaglia – NOVA Ferrara

La nascita di un'app Android – Programmazione in Android



- La presentazione, il filmato, i materiali e i contenuti in essi inclusi sono di proprietà dell'Università di Ferrara
- Il diritto morale d'autore ("Proprietà Intellettuale") appartiene ai singoli docenti/relatori dell'evento
- L'utilizzo è concesso **per uso esclusivo e personale**
- Nessun altro utilizzo può essere legittimamente esercitato senza la previa autorizzazione scritta dell'Ateneo e dei proprietari del diritto morale d'autore
- Qualunque abuso verrà perseguito a norma di legge
- Per ulteriori informazioni visitare il sito **dmi.unife.it/stageInformatica**

Neuroni di bit

Reti neurali e applicazioni

Marco Alberti
marco.alberti@unife.it



Dipartimento
di Matematica
e Informatica

Sappiamo scrivere algoritmi espliciti che rendono i computer bravi a fare le cose complesse (per noi), come:

- Calcoli numerici
- Calcolo di integrali
- Ricerca fra grandi moli di dati

ma non a fare quelle naturali (per noi):

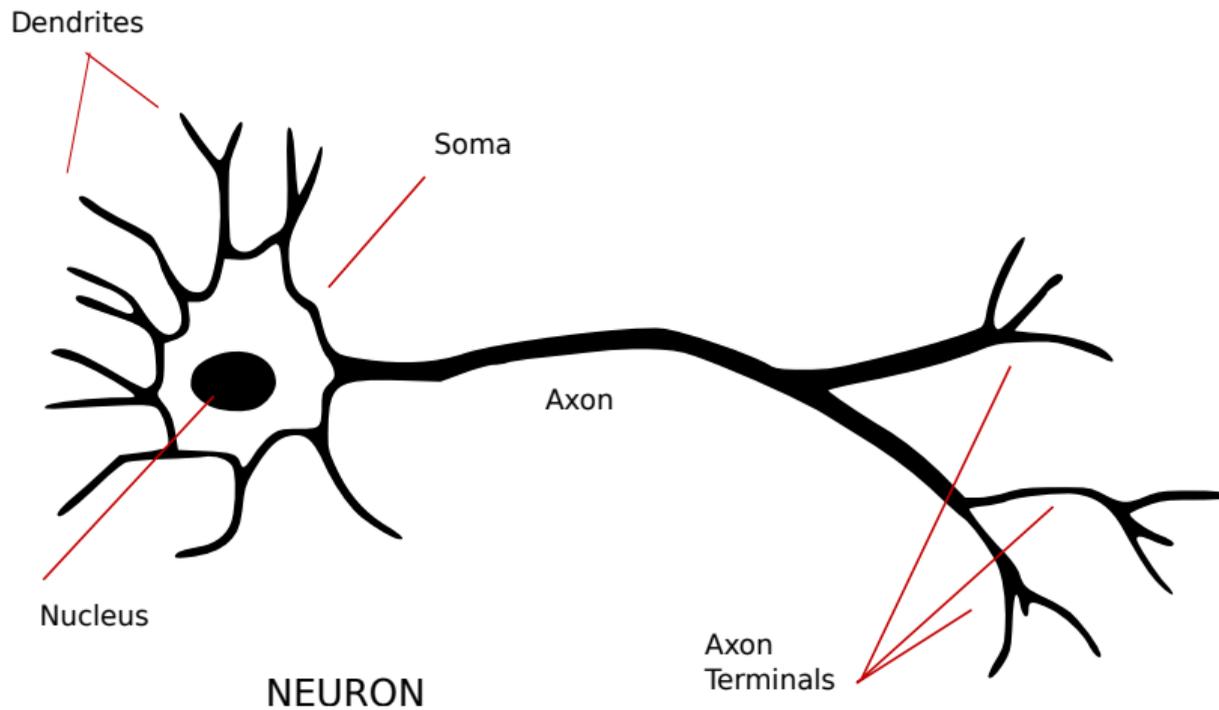
- riconoscimento volti
- coordinazione sensoriale-motoria
- usare il linguaggio naturale

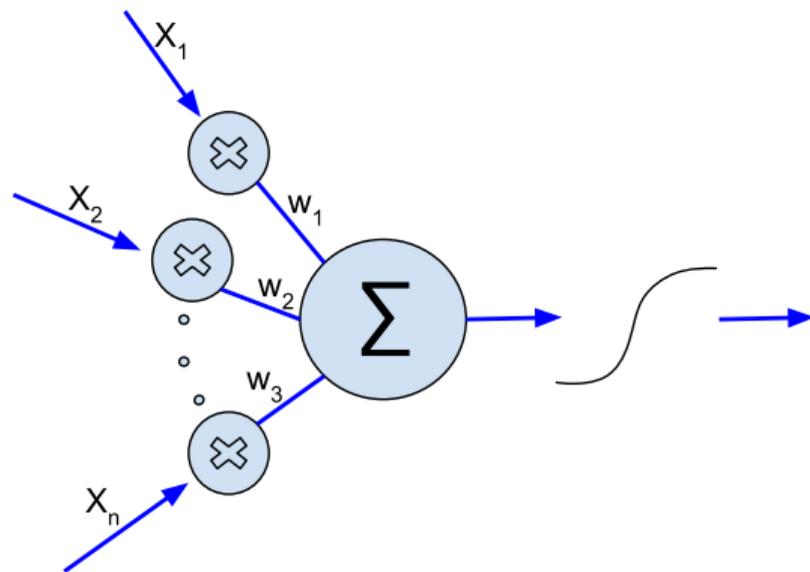
In realtà queste ultime attività sono così complesse che spesso non si riesce a codificare esplicitamente un algoritmo per eseguirle.

Meglio mettere i computer in grado di impararle da soli, osservando (molti) esempi:
apprendimento automatico.

- Modelli matematici ispirati a sistemi biologici (il nostro sistema nervoso)
- Possono essere addestrate automaticamente a effettuare (alcune) operazioni
- Nate negli anni Cinquanta (*perceptron*)
- Applicate con successo a molti problemi grazie ai progressi (hardware e algoritmi) degli ultimi 15 anni.
- Applicazioni: riconoscimento e classificazione in immagini, previsioni (es. andamento quotazioni, default), guida autonoma, ...
- Sommario:
 - Perceptron
 - Reti multistrato
 - Applicazioni

Neurone



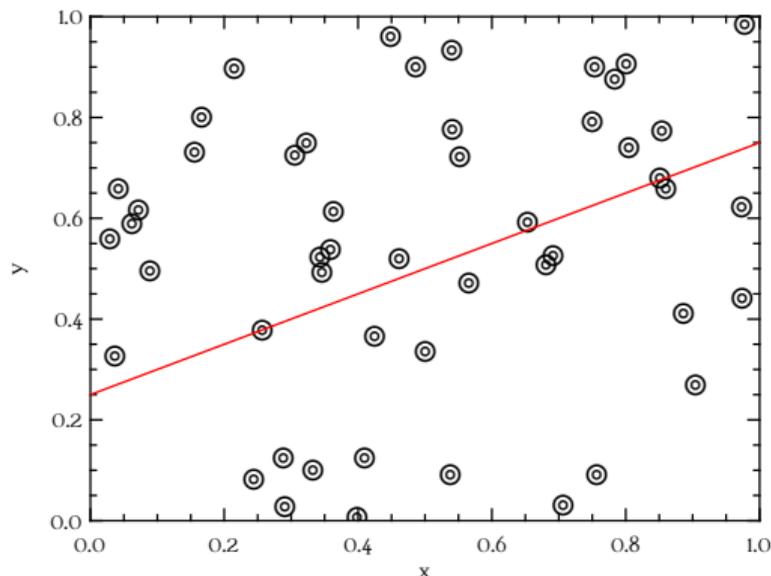


By Raquel Garrido Alhama, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=63514278>

- Ispirato a un neurone biologico
- x_1, \dots, x_n sono gli input
- w_1, \dots, w_n sono i pesi
- gli input vengono moltiplicati per il loro peso e sommati
- alla somma si applica una funzione per ottenere l'output

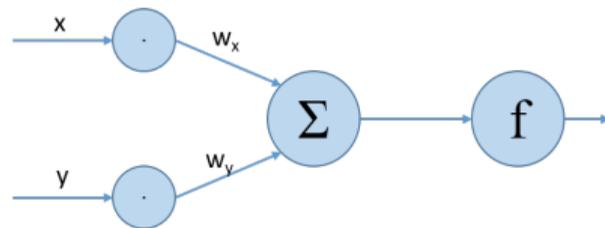
Classificazione con il perceptron

Vogliamo classificare degli oggetti
(rappresentati da x e y) in positivi e negativi:
consideriamo positivi gli esempi al di sopra
della retta $-x + 2y = 0.5$

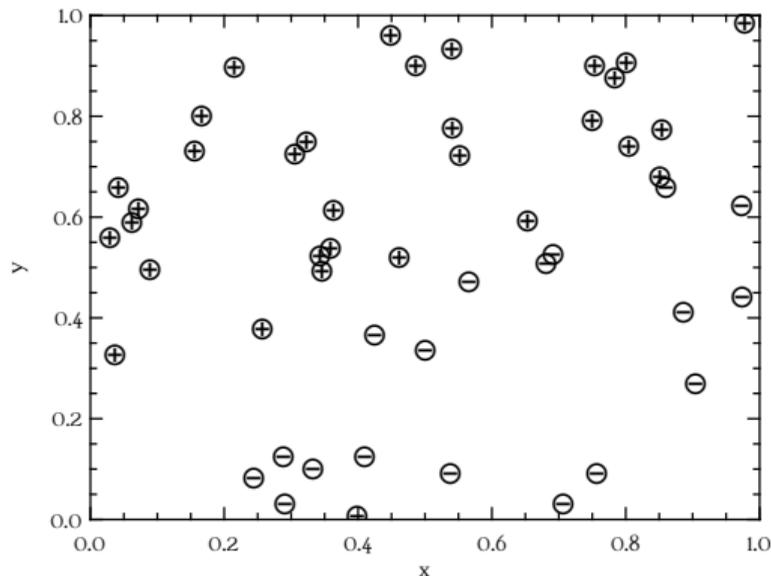


Questi oggetti vengono classificati
correttamente da un perceptron con

- input x e y
- $w_x = -1$ e $w_y = 2$
- funzione di attivazione
$$f(x) = \begin{cases} 1 & \text{se } x > 0.5 \\ 0 & \text{altrimenti} \end{cases}$$

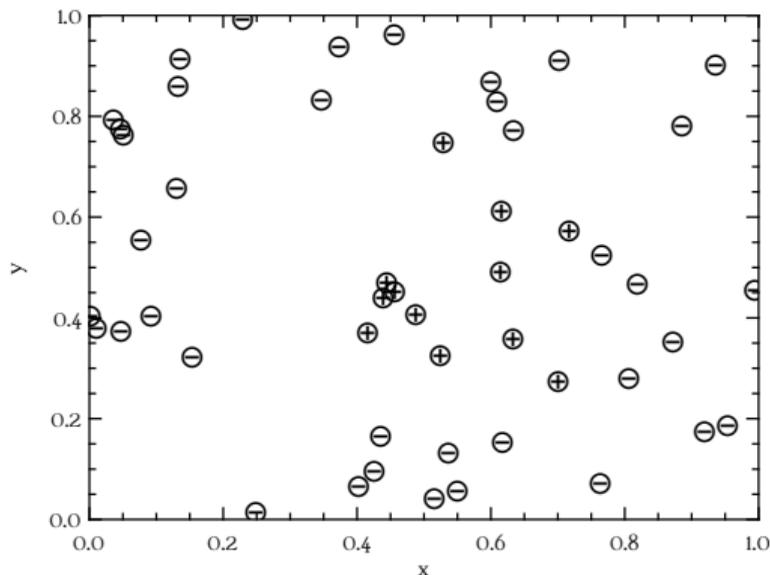


E se la retta non c'è? Mostriamo al perceptron degli esempi per addestrarlo.



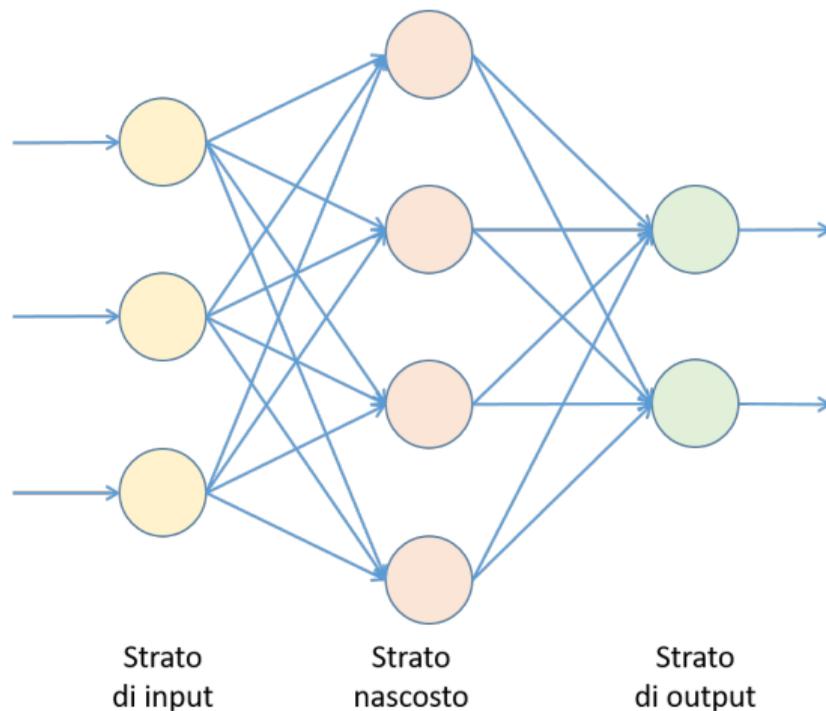
Algoritmo di apprendimento standard

- Inizialmente assegna a w_x e w_y valori casuali
- Se il perceptron classifica scorrettamente un esempio (x, y) , aggiorna i pesi:
 $w'_x = w_x + \alpha gx$, $w'_y = w_y + \alpha gy$
 dove
 - α è un parametro dell'algoritmo, detto velocità di apprendimento
 - g vale 1 se l'esempio è negativo e -1 se è positivo
- Termina quando tutti gli esempi sono classificati correttamente: il perceptron è addestrato e può classificare oggetti non ancora incontrati.



- Teorema (Novikoff 1962): l'algoritmo standard *converge* (cioè arriva a classificare correttamente tutti gli esempi)...
- ... se gli esempi sono *linearmente separabili* (cioè se esiste una retta che separa i positivi dai negativi)
- altrimenti, un perceptron non basta: reti multistrato.

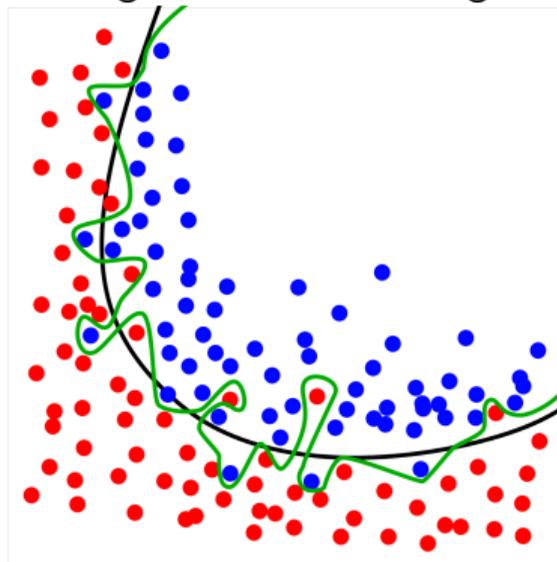
Reti multistrato (feedforward)



Reti a due strati:

- Possono classificare fra più classi (si prende la classe con l'attivazione di output più alta)
- Separano qualsiasi regione convessa (basta un neurone nascosto per ogni lato)
- In generale, approssimano una qualsiasi funzione (con abbastanza neuroni nascosti)
- Algoritmo di addestramento: backpropagation (corregge i pesi per bilanciare gli eventuali errori, andando all'indietro)

Riga verde: overfitting



autore:

https://commons.wikimedia.org/wiki/User:Ignacio_Icke

Quindi aggiungiamo neuroni allo strato nascosto fino ad annullare l'errore?

- Mentre in teoria è possibile, in pratica non è una buona idea:
- gli esempi possono contenere rumore (dati errati)
- si rischia di avere una rete che classifica perfettamente gli esempi (compresi quelli sbagliati) ma è tratta in inganno dai nuovi oggetti (che è il suo lavoro principale)

Rimedi:

- Complessità della rete commisurata ai dati disponibili
- Accettare una certa quantità di errore di addestramento

Mentre in teoria è sufficiente una rete feedforward a due strati, in molte applicazioni si ottengono risultati migliori

- con un numero maggiore di strati nascosti (*deep learning*):
- con connessioni non solo fra uno strato e il successivo

Più complessa l'architettura,

- più complessi i dati che riesce a discriminare, ma
- più dati richiesti per l'addestramento,
- più complesso l'addestramento,
- maggior errore di generalizzazione (*overfitting*).

Fra le tante architetture, reti neurali convoluzionali: ispirate all'apparato visivo degli animali, riducono connessioni non necessarie, ottime prestazioni nel riconoscimento di immagini.

- Composte da due sottoreti in conflitto: *generativa* e *discriminativa*. Esempio: fotografie di persone
- La rete discriminativa è addestrata con dati validi a riconoscere foto reali da foto sintetiche
- La rete generativa genera dati sintetici (false fotografie di persone inesistenti) ed è addestrata cercando di massimizzare l'errore della rete discriminativa
- La rete discriminativa affina il suo addestramento sulle immagini della rete generativa
- Così entrambe migliorano la loro abilità nel tempo

Generazione di

- testo, immagini artistiche (<https://thisartworkdoesnotexist.com/>), foto (<https://thispersondoesnotexist.com/>) ...
- <https://thisxdoesnotexist.com/>

Pregi (rispetto ad altri modelli addestrabili):

- Prestazioni: in molti casi raggiungono livelli di accuratezza superiori ad altri sistemi, o anche ad esperti umani

Difetti:

- non sempre fanno l'uso più appropriato delle informazioni simboliche
- parametri (spesso numerosissimi) difficili da interpretare
- difficile motivare una risposta (limitante per applicazioni che comportano responsabilità): ricerca attiva su "explainable AI"

L'AI generale (HAL 9000) è dietro la porta? Probabilmente non ancora.

bup

Rete neurale: modello matematico ispirato al cervello umano, addestrato automaticamente a partire da dati.

Usate per riconoscimento, classificazione, previsioni, creazione di contenuti.

Una rete neurale è caratterizzata da

- Architettura (neuroni, strati, connessioni)
- Parametri (pesi)

Il *data scientist*, considerati i dati a disposizione,

- seleziona un'architettura,
- addestra la rete con i dati, determinandone i pesi
- verifica le prestazioni su dati non usati per l'addestramento

Esistono librerie che facilitano la creazione di reti neurali per molti linguaggi di programmazione.

- Tensorflow Playground:
<https://playground.tensorflow.org/>
- Libro su Deep Learning:
<https://www.deeplearningbook.org/>
- Campionario di architetture:
<https://www.asimovinstitute.org/neural-network-zoo/>
- Università di Ferrara!
<http://www.unife.it>