
Introduzione agli Algoritmi Genetici ed alle Reti Neurali

Maurizio Palesi

Sommario

■ Algoritmi Genetici

- Introduzione e applicazioni
- Terminologia
- Mutazione, ricombinazione, selezione
- Problemi e regole euristiche

■ Reti Neurali Artificiali

- Introduzione e applicazioni
- Differenze con computer seriale
- Perceptrone e perceptrone multistrato
- Valutazione

Parte I

Algoritmi Genetici

Introduzione (1/2)

- L'obiettivo di creare l'*intelligenza artificiale* risale alle origini dell'era informatica
- Già i primi informatici inseguivano il sogno di instillare nei programmi l'intelligenza
 - La capacità di *duplicarsi* e di *imparare*, nonché quella di *controllare* l'ambiente circostante

Introduzione (2/2)

- Il tentativo di costruire modelli per il cervello ha dato origine al settore delle *reti neurali* (NN)
- Quello di imitare l'apprendimento umano al settore dell'*apprendimento automatico*
- La simulazione dell'evoluzione biologica ha dato vita al campo della *computazione evolutiva*, di cui gli *algoritmi genetici* (GA) sono l'esempio più importante

Ispirarsi all'Evoluzione

■ Perché ispirarsi all'evoluzione per risolvere problemi computazionali?

→ E' necessario ricercare **la soluzione** tra un **numero enorme** di possibili **alternative**

✓ Es., Individuare una proteina con determinate proprietà tra un numero enorme di possibili sequenze di amminoacidi

→ E' necessario che un programma funzioni anche quando l'ambiente cambia

✓ Es., Problemi di controllo dei robot

Evoluzione

- L'evoluzione è un **processo continuo distribuito** su una scala temporale ampia che cambia una **popolazione di organismi generando** prole via via **migliore** attraverso la **riproduzione**

Computazione Evoluzionistica

- Termine generico che indica una gamma di sistemi di risoluzione dei problemi basati sull'utilizzo del calcolatore e affini ai processi evolutivi
 - Algoritmi Genetici
 - Programmazione Evolutiva
 - Strategie Evolutive
 - Sistemi Classificatori
 - Programmazione Genetica

Applicazioni

- *Ottimizzazione* (es., addestramento di reti neurali, minimizzazione di funzioni costo, layout di circuiti, scheduling, ...)
- *Predizione* (es., previsioni meteorologiche, disposizione spaziale di proteine, ...)
- *Classificazione* (es., verifica di qualità, ...)
- *Economia* (es., strategie d'offerta, valutazione del mercato, ...)
- *Ecologia* (es., competizione biologica, coevoluzione ospite-parassita, ...)

Indicati per...

- Ricerca in spazi di dimensionalità elevata
 - Multimodali
 - Non “smooth”
- Funzioni con rapporto S/N elevato
 - Di cui non si conosce la formulazione analitica
 - O tale formulazione è estremamente complessa
- Convergenza sub-ottima ma in un tempo ragionevole

Terminologia

■ Cromosoma

- La codifica di una possibile soluzione per un dato problema
- Solitamente rappresentata tramite un array di bit o caratteri

■ Gene

- Un singolo bit o insieme di bit che codifica una parte della soluzione

■ Allele

- Uno degli elementi utilizzati per codificare i geni

■ Fitness

- Valutazione della soluzione attuale

Simulare l'Evoluzione

■ Ricombinazione

→ Generare nuove soluzioni “mescolando” due o più soluzioni esistenti

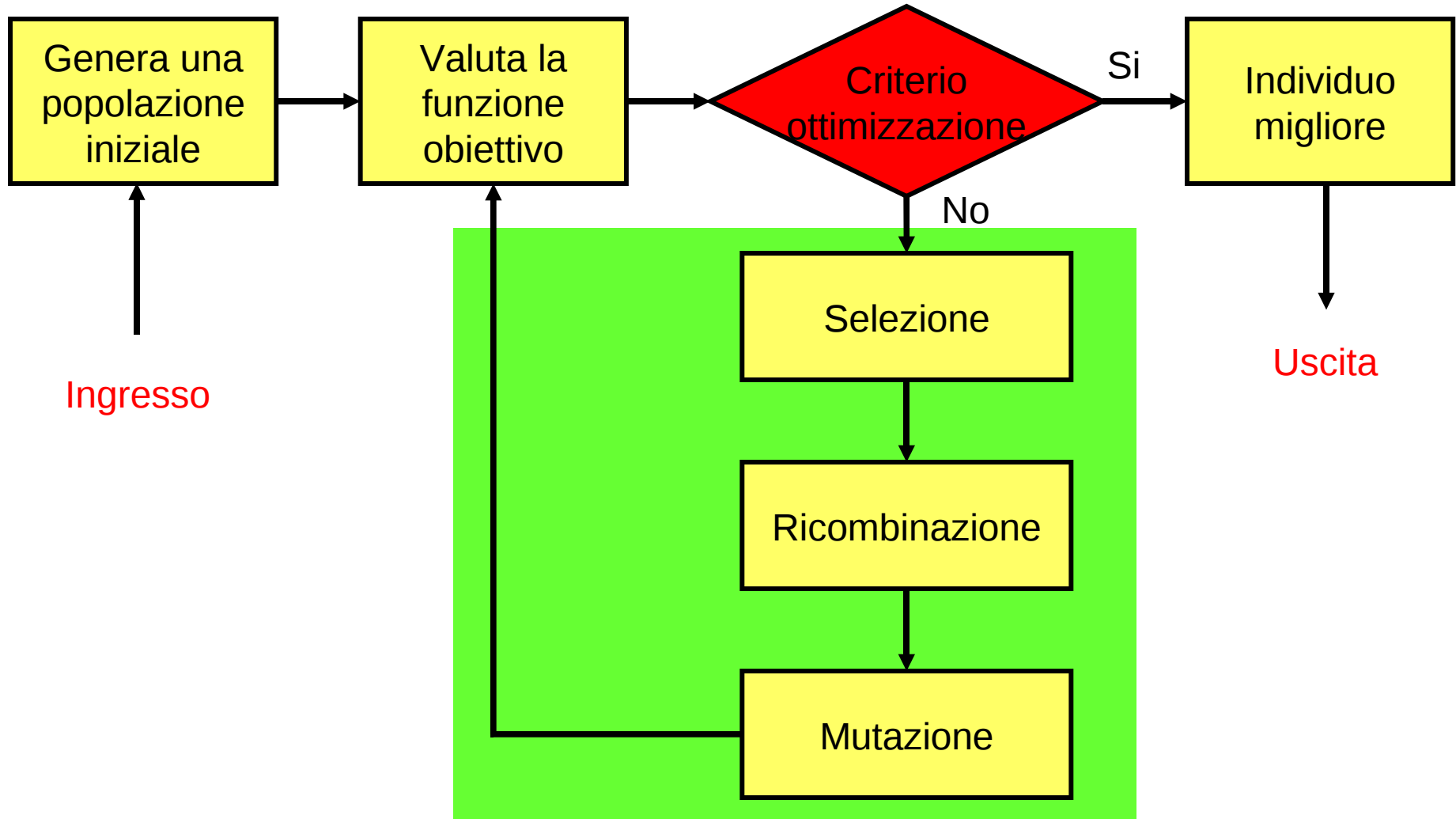
■ Mutazione

→ Cambiamenti casuali nella soluzione

Problemi di Ottimizzazione

- Lo spazio delle soluzioni possibili contiene una o più soluzioni accettabili
 - ➔ Eventualmente alcune ottime e altre sub-ottime
- Gli algoritmi di ricerca servono a localizzare tali soluzioni evitando di esplorare tutte le soluzioni possibili

Algoritmo Genetico



Algoritmo Genetico

- **1.** [**Inizia**] Genera una popolazione casuale di n cromosomi (soluzioni plausibili - genitori)
- **2.** [**Fitness**] Valuta la fitness $f(x)$ di ciascun cromosoma x nella popolazione corrente
- **3.** [**Nuova popolazione**] Crea una nuova popolazione (prole) attraverso
 - [**Selezione**] Seleziona i cromosomi genitori in base alla loro fitness
 - [**Ricombinazione**] Con una certa probabilità di ricombinazione incrocia due genitori per generare un nuovo figlio. Se la ricombinazione ha probabilità nulla, il figlio è una copia esatta dei genitori
 - [**Mutazione**] Con una certa probabilità di mutazione cambia gli elementi che costituiscono il figlio
 - [**Accettazione**] Introduce il nuovo figlio nella popolazione
- **4.** [**Verifica**] Se la condizione di uscita è soddisfatta, la soluzione del problema è rappresentata dall'elemento nella popolazione che presenta la migliore fitness
- **5.** [**Ciclo**] Vai al passo **2.**

Problemi Intrinseci

- Come creare i cromosomi
 - Che tipo di codifica utilizzare
- Come selezionare i genitori per la ricombinazione
 - Nella speranza che i genitori migliori produrranno migliore prole
- Come definire la ricombinazione e la mutazione

Codifica del Cromosoma

- Il primo passo per sviluppare un GA consiste nel definire la **codifica della soluzione**
 - Un cromosoma deve contenere l'informazione sulla soluzione che rappresenta
 - La codifica dipende dal problema da risolvere
 - ✓ Numeri interi, numeri reali, permutazione, alberi di parsing, ...

Codifica del Cromosoma

- La modalità usuale di codifica consiste nell'utilizzare una **stringa binaria**
 - Ciascun bit nella stringa rappresenta una qualche caratteristica della soluzione
- Esempio di cromosoma
 - Rappresentazione binaria di un numero intero

Cromosoma 1:

1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Cromosoma 2:

1	1	1	0	1	1	1	0	0	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Codifica Implicita

- La codifica binaria è la più comune codifica implicita
 - Ogni cromosoma è una stringa di bit
 - L'implementazione di operatori genetici è immediata
 - Tuttavia non è sempre naturale per molti problemi
- Esempio: Problema dello zaino
 - Dato un certo numero di oggetti caratterizzati da un valore e una dimensione, massimizzare il valore degli oggetti in uno zaino che ha una capacità fissata

Codifica Implicita

Capacità dello zaino 200

Oggetto	Valore	Dimensione
Anello 1	20	5
Anello 2	30	8
Anello 3	35	7
Orologio 1	30	10
Collana 1	50	15
Candelabro 1	100	55
Candelabro 2	150	65
Quadro 1	200	70
Quadro 2	250	80
Quadro 3	100	50
Quadro 4	120	60
Pelliccia	150	120
...

Cromosoma

Anello 1	1
Anello 2	1
Anello 3	0
Orologio 1	1
Collana 1	1
Candelabro 1	0
Candelabro 2	0
Quadro 1	1
Quadro 2	0
Quadro 3	0
Quadro 4	0
Pelliccia	0
...	...

Codifica di Permutazioni

- Problema del commesso viaggiatore
 - E' dato un insieme di città e le corrispondenti distanze a coppie
 - Il commesso viaggiatore deve visitarle tutte ma non vuole viaggiare più del necessario
 - Scopo:
 - ✓ Trovare una sequenza di città che minimizza la distanza percorsa
 - Il cromosoma descrive l'ordine delle città visitate

Complessità del Problema

- Sede di lavoro Springfield
- In programma la visita di
 - Oldtown
 - Midtown
 - Newtown
- Si parte in auto da Springfield e si torna a Springfield
- Scopo: minimizzare i costi (tempo e/o benzina)
 - Minimizzare la lunghezza totale del tragitto

Complessità del Problema

	Springfield	Oldtown	Midtown	Newtown
Springfield	0	54	17	79
Oldtown	54	0	49	104
Midtown	17	49	0	91
Newtown	79	109	91	0

Complessità del Problema

Itinerario	Distanza totale		
S-O-M-N-S	$54 + 49 + 91 + 79$	=	273
S-O-N-M-S	$54 + 104 + 91 + 17$	=	266
S-M-N-O-S	$17 + 91 + 109 + 54$	=	271
S-M-O-N-S	$17 + 49 + 104 + 79$	=	249
S-N-O-M-S	$79 + 109 + 49 + 17$	=	254
S-N-M-O-S	$79 + 91 + 49 + 54$	=	273



Complessità del Problema

■ Quanti sono i possibili itinerari?

■ 3 città

→3 possibilità per la prima città

→2 possibilità per la seconda città

→1 possibilità per la terza città

$$3 \times 2 \times 1 = 6$$

Complessità del Problema

■ Quanti sono i possibili itinerari?

■ 10 città

→ 10 possibilità per la prima città

→ 9 possibilità per la seconda città

→ 8 possibilità per la terza città

→ ...

→ 1 possibilità per la decima città

$$10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = \\ = 10! = \mathbf{3.628.800}$$

Complessità del Problema

Città	Numero di possibili itinerari
-------	-------------------------------

10	3628800
----	---------

11	39916800
----	----------

12	479001600
----	-----------

13	6227020800
----	------------

14	87178291200
----	-------------

15	1307674368000
----	---------------

16	20922789888000
----	----------------

17	355687428096000
----	-----------------

18	6402373705728000
----	------------------

19	121645100408832000
----	--------------------

20	2432902008176640000
----	---------------------

21	51090942171709400000
----	----------------------

22	1124000727777610000000
----	------------------------

23	25852016738885000000000
----	-------------------------

24	6204484017332390000000000
----	---------------------------

25	15511210043331000000000000
----	----------------------------

1 valutazione eseguita in 1 ms

← 491857243890506 anni!

Codifica Esplicita

- Il cromosoma è una sequenza di valori connessi al problema

Tempo di cottura (sec)	Temperatura di cottura (°C)	Quantità di sale (gr)	Quantità di pepe (gr)	Coperchio (Si/No)	Quantità di vino (ml)
------------------------------	-----------------------------------	-----------------------------	-----------------------------	----------------------	-----------------------------

30	120	15	5	No	100
----	-----	----	---	----	-----

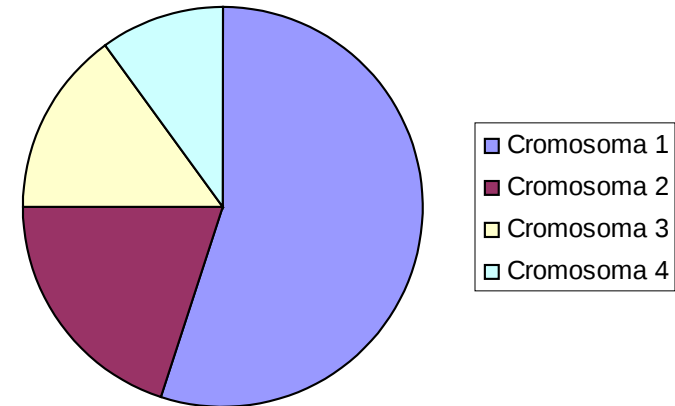
- Rappresenta la scelta migliore per l'ottimizzazione in spazi multidimensionali e/o multimodali
 - ➔ Gli operatori di ricombinazione e mutazione richiedono specifici controlli di consistenza

Selezione

- In accordo alla teoria evoluzionistica di Darwin il migliore cromosoma sopravvive per creare nuova prole
- Per selezionare il migliore cromosoma esistono diversi criteri
 - Ordinamento (priorità diretta ai cromosomi che si sono espressi meglio)
 - Casualità (metodo della roulette)
 - ...

Selezione Tramite Roulette

- Gli individui sono selezionati proporzionalmente alla loro fitness
 - Migliore essa è e più alta è la probabilità di selezione
- **1.** Si immagina una roulette dove sono sistemati tutti i cromosomi della popolazione
- **2.** La dimensione della sezione nella roulette è proporzionale al valore della fitness di ciascun cromosoma
- **3.** La pallina viene lanciata all'interno della roulette e il cromosoma in corrispondenza del quale si ferma è quello selezionato

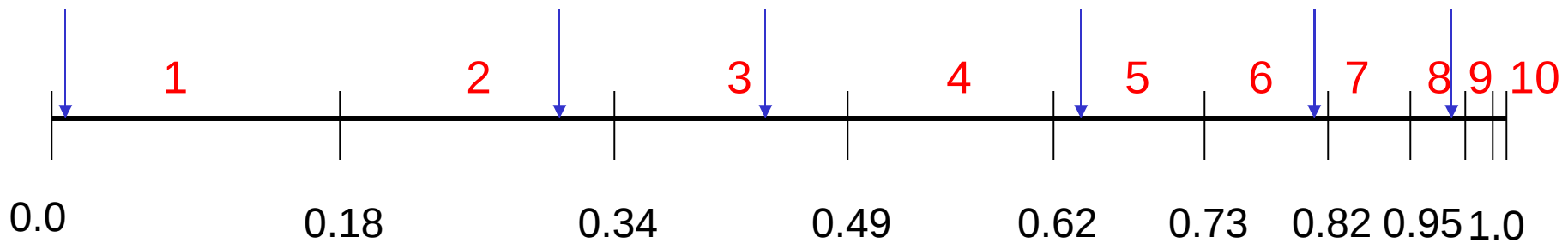


Roulette

Numero di individui	1	2	3	4	0	6	7	8	9	10	11
Fitness	2,0	1,8	1,6	1,4	1,2	1,0	0,8	0,6	0,4	0,2	0,1
Probabilità di selezione	0,18	0,16	0,10	0,13	0,11	0,09	0,07	0,06	0,03	0,02	0,01

■ Generazione casuale di 6 numeri

→ 0.81, 0.32, 0.96, 0.01, 0.65, 0.42



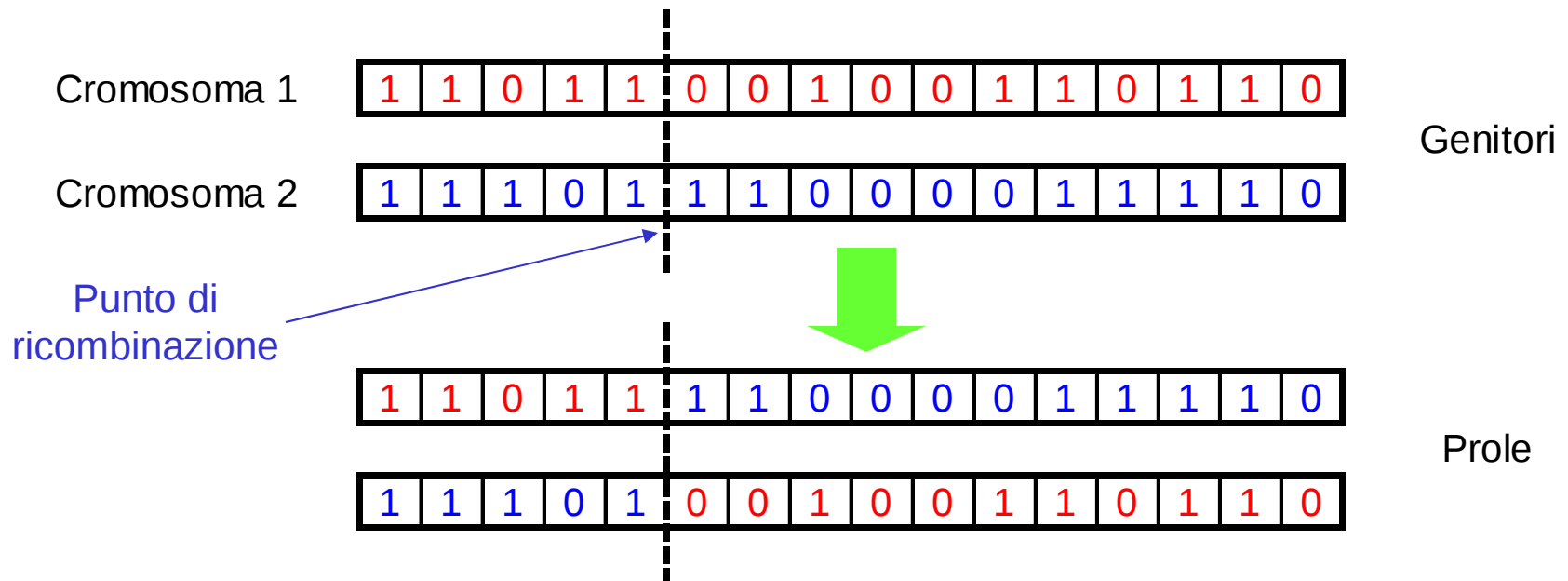
■ Dopo la selezione la nuova popolazione è costituita dai seguenti individui

→ 1, 2, 3, 5, 6, 9

Operatore di Ricombinazione

- Il processo di ricombinazione opera su geni selezionati dal cromosoma genitore e crea nuova prole
 - 1. Selezione casuale di un *punto di ricombinazione* all'interno del cromosoma
 - 2. Copia tutti i geni precedenti questo punto dal primo genitore e poi copia tutti i geni successivi a questo punto dal secondo genitore

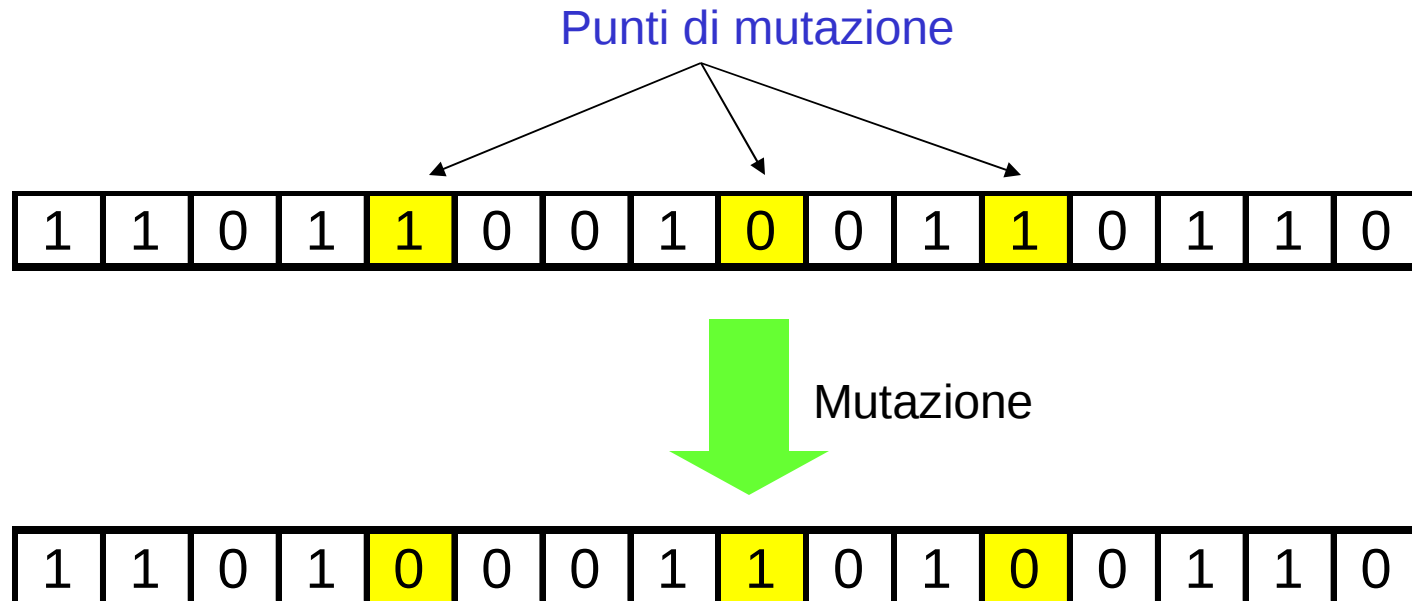
Operatore di Ricombinazione



- La ricombinazione dipende principalmente dalla codifica dei cromosomi
- Una ricombinazione specifica per un dato problema può migliorare o ridurre le prestazioni di un GA

Operatore di Mutazione

- Dopo la ricombinazione interviene il processo di mutazione
- Cambia in modo casuale un numero contenuto di bit da 0 a 1 o viceversa da 1 a 0



Operatore di Mutazione

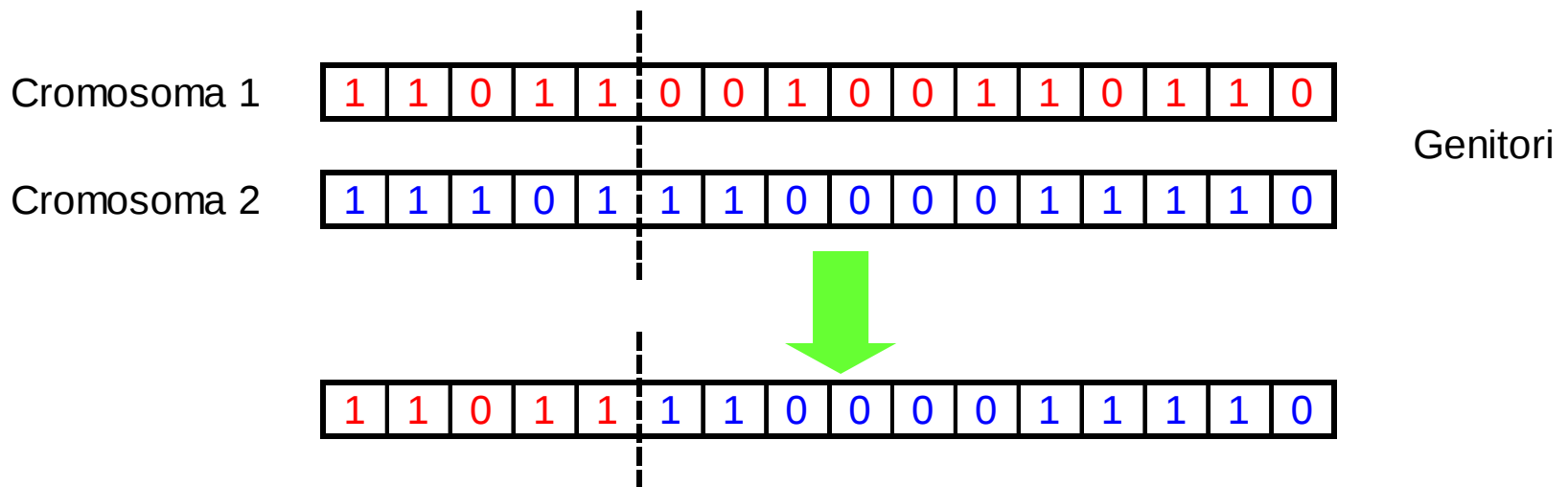
- La mutazione è concepita per evitare la possibile caduta delle soluzioni della popolazione in ottimi locali
- Anche la mutazione dipende dal tipo di codifica dei cromosomi
 - Es., se si codifica per permutazioni, la mutazione può essere effettuata come scambio tra due geni

Ricombinazione

- Per la codifica binaria esistono diverse modalità
 - Ricombinazione da un singolo punto
 - Ricombinazione da due punti
 - Ricombinazione uniforme
 - Ricombinazione aritmetica

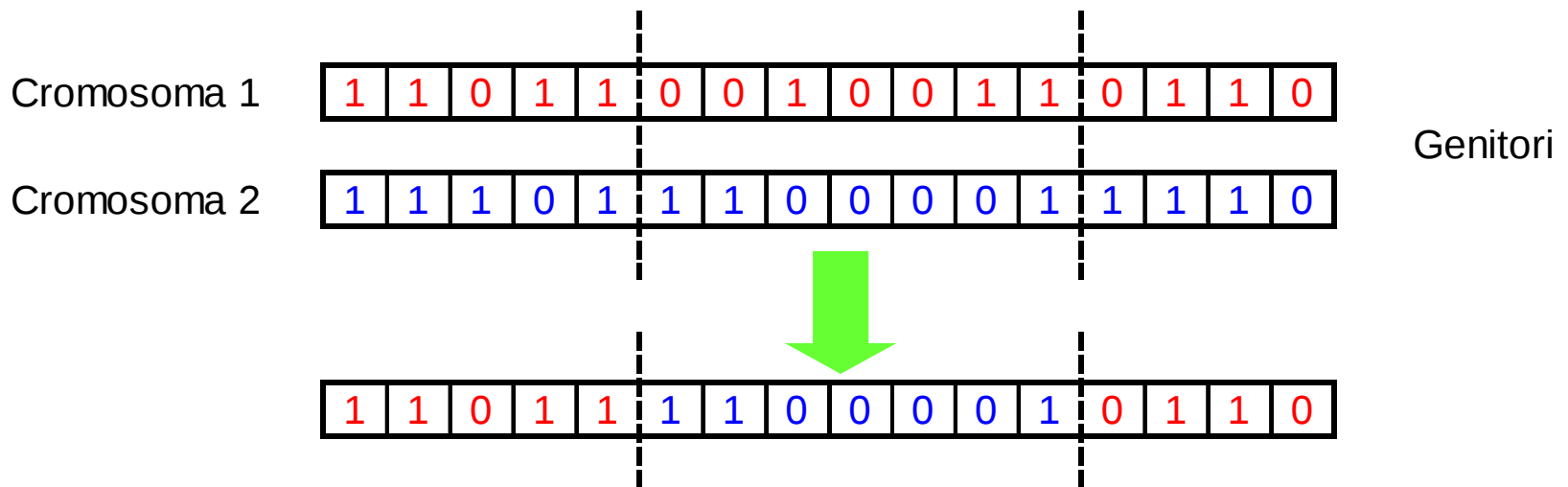
Ricombinazione da Singolo Punto

- Selezionato il punto all'interno del cromosoma, la stringa binaria compresa tra l'inizio e il punto di ricombinazione è copiata dal primo genitore, il resto è copiato dal secondo genitore



Ricombinazione da Due Punti

- Selezionati i due punti all'interno del cromosoma, la stringa binaria compresa tra l'inizio e il primo punto di ricombinazione è copiata dal primo genitore, la parte compresa tra il primo e il secondo punto è copiata dal secondo genitore, la parte compresa tra il secondo punto e la fine del cromosoma è copiata ancora dal primo genitore



Ricombinazione Uniforme

- I bit sono copiati casualmente dal primo o dal secondo genitore

Cromosoma 1

1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Cromosoma 2

1	1	1	0	1	1	1	0	0	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

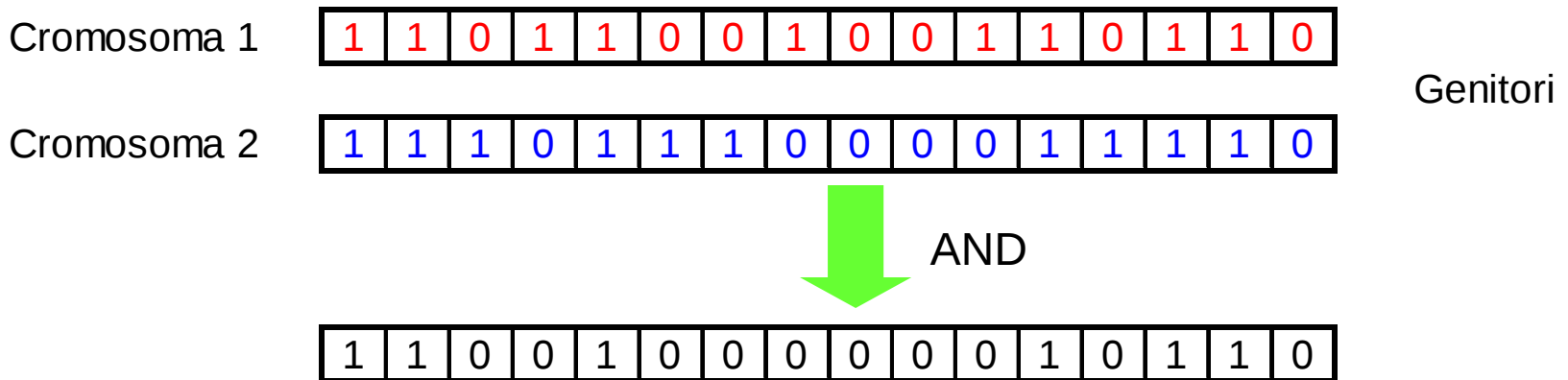
Genitori



1	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Ricombinazione Aritmetica

- La ricombinazione avviene attraverso l'uso degli operatori aritmetici e/o logici



Problemi e Regole Euristiche

- Funzione costo
- Dimensione della popolazione
 - Ridotta: convergenza veloce
 - Ampia: evitano ottimi locali ma il costo computazionale può essere critico
 - Di solito 30-50 individui rappresentano un compromesso adeguato
- Il tasso di ricombinazione dovrebbe essere alto
 - Intorno al 90%
- Il tasso di mutazione dovrebbe essere basso
 - Circa 1% per allele
- L'elitismo dovrebbe essere implementato se non si prevedono altri meccanismi che tengono memoria delle soluzioni migliori
 - Problema della convergenza prematura

Parte II

Reti Neurali Artificiali

Il Dilemma dell'IA

- I computer sono eccellenti nel calcolo, ma falliscono quando si cerca di riprodurre attività tipicamente umane
 - Percezione sensoriale
 - Coordinamento senso-motorio
 - Riconoscimento di immagini
 - Capacità di adattamento

Gli Scacchi

- Ogni posizione ammette in media 20 mosse legali
- Lo spazio di ricerca è immenso (10^{120})
- Potenza attuale dei computer
 - $10^8 - 10^9$ posizioni/sec
- Ricerca esaustiva
 - 10^{111} s ($U \approx 10^{17}$ s)
 - 10^{94} U

Bambino batte Computer 3 a 0

- Sebbene un computer possa battere il campione del mondo di scacchi, esso non è in grado di competere con un bambino di 3 anni nel
 - Costruire con il Lego
 - Riconoscere il volto di una persona
 - Riconoscere la voce dei genitori

Problema

- Le azioni complesse dipendono da molti fattori, che non possono essere previsti esattamente in un programma
- Tali fattori devono essere acquisiti con *l'esperienza*, in una fase di *apprendimento*

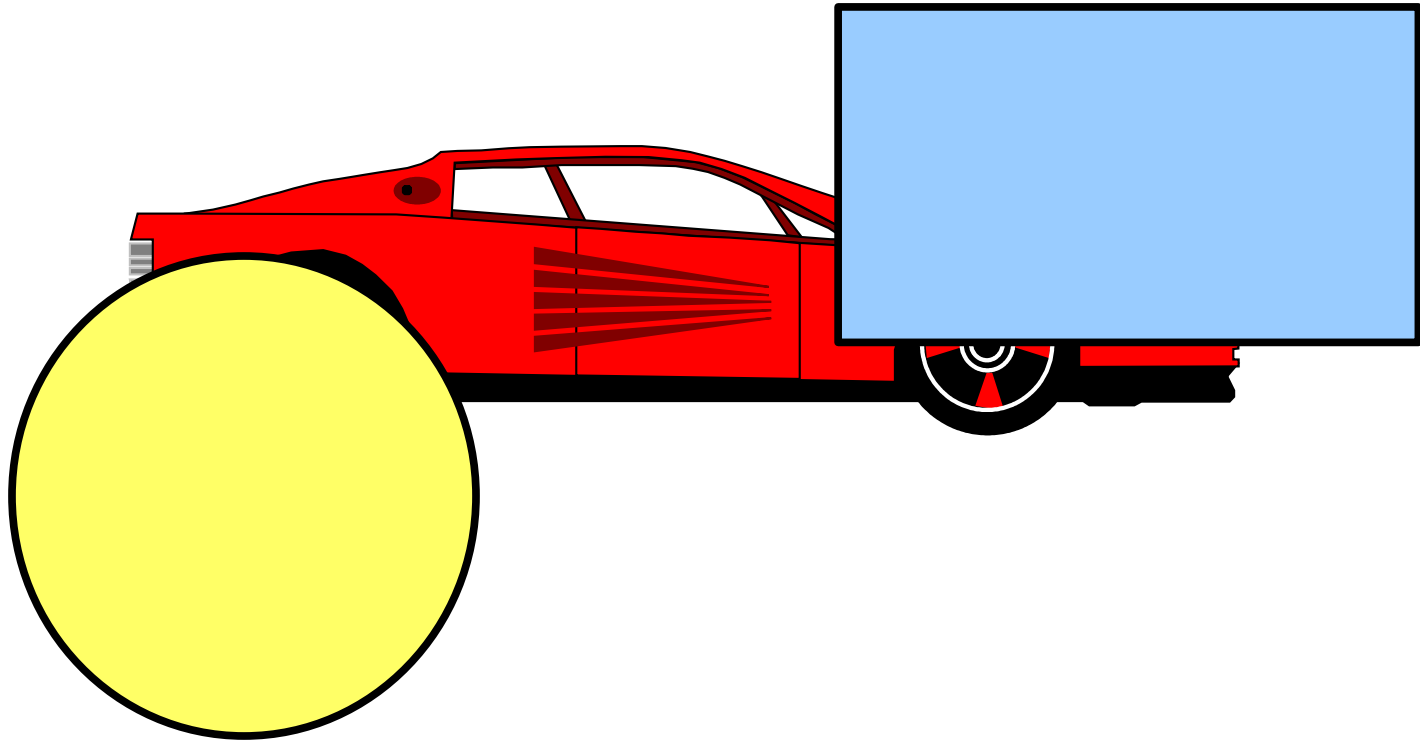
Esempi

- Afferraggio di un oggetto è determinato da numerosi fattori
 - La posizione dell'oggetto
 - La nostra postura
 - La dimensione e la forma dell'oggetto
 - Il peso previsto
 - Gli eventuali ostacoli interposti

Riconoscimento del Parlato

- Richiede una fase di apprendimento necessaria per
 - Adattarsi al soggetto che parla
 - Filtrare i rumori esterni
 - Separare eventuali altre voci

Riconoscimento delle Immagini



L'Approccio Neuronale

- L'estrema difficoltà di trattare questi problemi con il calcolatore ha fatto nascere l'esigenza di studiare nuove metodologie di calcolo, ispirate alle reti neurali

- **Medici**

- ✓ Studi sul cervello

- **Ingegneri**

- ✓ Risoluzioni di problemi

Evoluzione della Ricerca

- **1943**, **McCulloch** e **Pitts**: nasce il primo modello neurale: il neurone binario a soglia
- **1949**, **Hebb**: dagli studi sul cervello, emerge che l'apprendimento non è una proprietà dei neuroni, ma è dovuto a una modifica delle sinapsi
- **1962**, **Rosenblatt**: propone un nuovo modello di neurone capace di apprendere mediante esempi: il perceptron
- **1969**, **Minsky** e **Papert**: dimostrano i limiti del perceptron: crolla l'entusiasmo sulle reti neurali

Evoluzione della Ricerca

- 1982, Hopfield: propone un modello di rete per realizzare memorie associative
- 1982, Kohonen: propone un tipo di rete autoorganizzante (mappe recettive)
- 1985, Rumelhart, Hinton e Williams: formalizzano l'apprendimento di reti neurali con supervisione (Back-Propagation)

Introduzione alle Reti Neurali

- Il cervello è un calcolatore e come tale può essere riprodotto in una macchina
- E' un calcolatore che opera in parallelo, diversamente dai calcolatori tradizionali che operano in modo seriale

A Cosa Servono?

- Le reti neurali offrono i seguenti specifici vantaggi nell'elaborazione dell'informazione
 - **Apprendimento basato su esempi** (non è richiesta l'elaborazione di un modello aderente alla realtà)
 - **Autoorganizzazione** dell'informazione nella rete
 - **Robustezza ai guasti** (codifica ridondante dell'informazione)
 - **Funzionamento in tempo reale** (realizzazione HW)

Computer Seriale (1/2)

- Processore molto veloce (10^6 operazioni al secondo)
- Tre tipi di memoria
 - Memoria che contiene le istruzioni
 - Memoria temporanea per i dati
 - Memoria permanente
- Programma: istruzioni organizzate in modo gerarchico e tabelle di consultazione
- L'informazione viene elaborata sequenzialmente utilizzando regole e conoscenze predefinite

Computer Seriale (2/2)

- Molto rapido ed efficiente in particolare in compiti difficili per gli esseri umani
 - Calcoli complessi, rotazioni di immagini tridimensionali, memorizzazione di grandi quantità di dati
 - ...che possiedono una soluzione analitica
- Inefficienti in compiti facili da risolvere per gli esseri umani
 - Trasformazioni sensorimotorie, riconoscimento visivo

Cervello

- 10^{11} neuroni che comunicano in media con 10^4 altri neuroni
- Ciascun neurone è un elemento di elaborazione che emette una risposta in funzione del segnale globale in ingresso e della propria soglia di attivazione
- Non esiste una distinzione tra elementi di elaborazione e sistemi di immagazzinamento dell'informazione
- Le conoscenze sono *incorporate* nelle connessioni (sinapsi)

Differenze Cruciali

■ Cervello

- Elaborazione in parallelo
- Elaborazione distribuita: molti neuroni sono coinvolti nella stessa operazione
- Accesso alle conoscenze per contenuto
- Apprendimento: modifica delle connessioni sinaptiche
- Codifica sub-simbolica

■ Calcolatore tradizionale

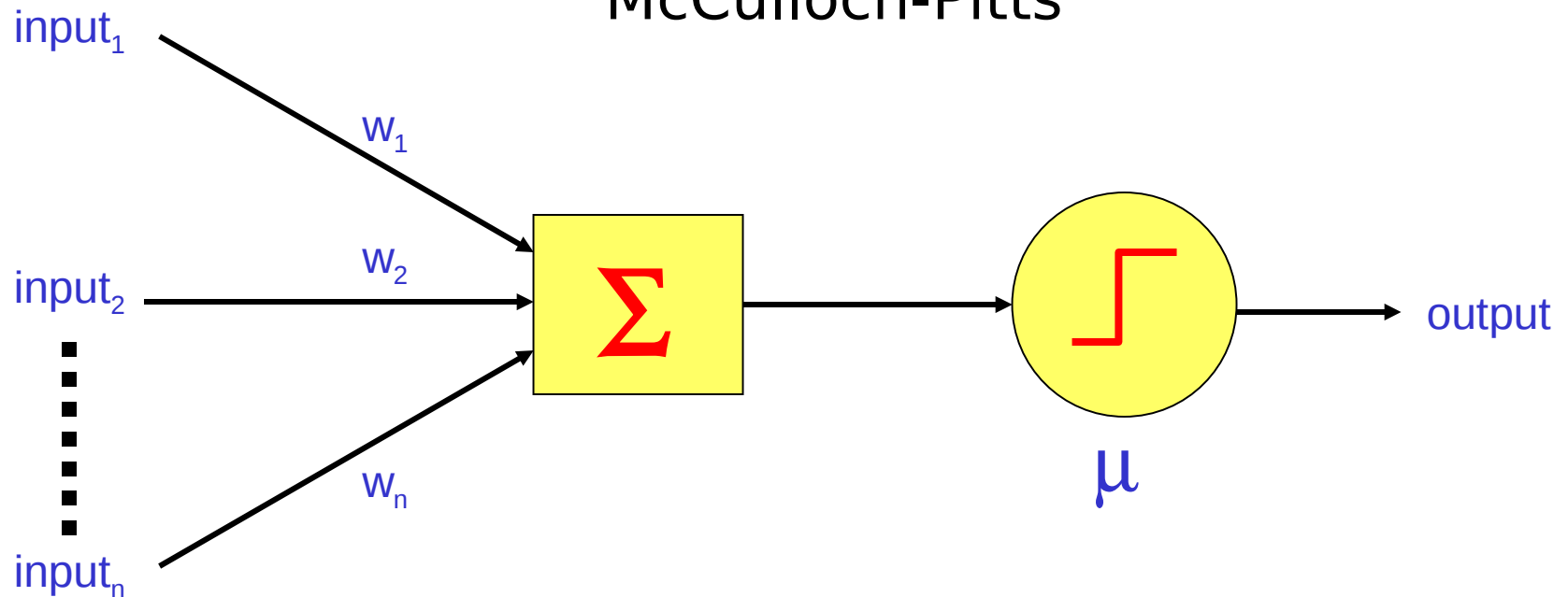
- Elaborazione seriale
- Accesso alle conoscenze per indirizzamento
- Richiedono programmi che contengano tutte le istruzioni necessarie a risolvere un compito
- Codifica simbolica
- Manipolazione di simboli

Le Reti Neurali

- Che cosa sono le Reti Neurali?
 - Sono sistemi di elaborazione composti da unità di calcolo (**nodi**, processori) operanti in parallelo, collegati da connessioni (**pesi**) attraverso cui viene trasferita l'informazione
 - Ogni unità può ricevere e inviare informazione ad una o più unità contemporaneamente, l'informazione quindi è distribuita nel sistema

Il Neurone Artificiale

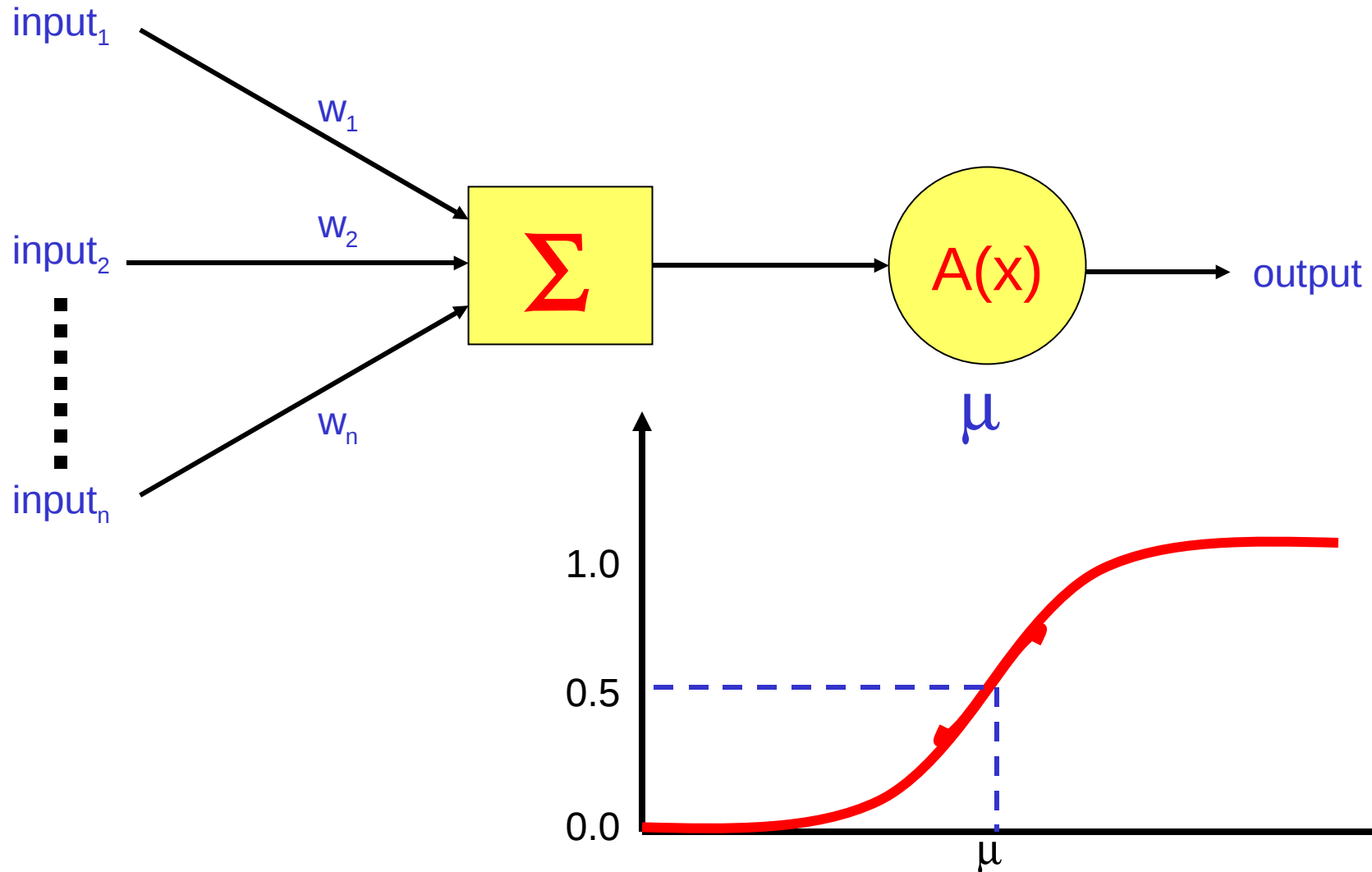
Modello di
McCulloch-Pitts



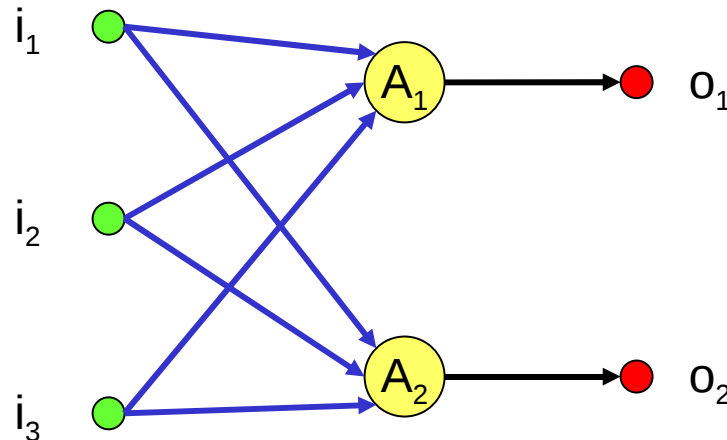
Critiche al Modello di McCulloch-Pitts

- I neuroni reali non possono essere ridotti ad un dispositivo a soglia
- Lo spike ha la sua forma continua che ha una durata di qualche millisecondo
- Il tempo di propagazione lungo i dendriti non viene considerato
- Gli input non sono sincroni
- Le interazioni tra input non sono lineari
- I pesi sono supposti costanti

Il Perceptrone (Rosemblatt, 1962)



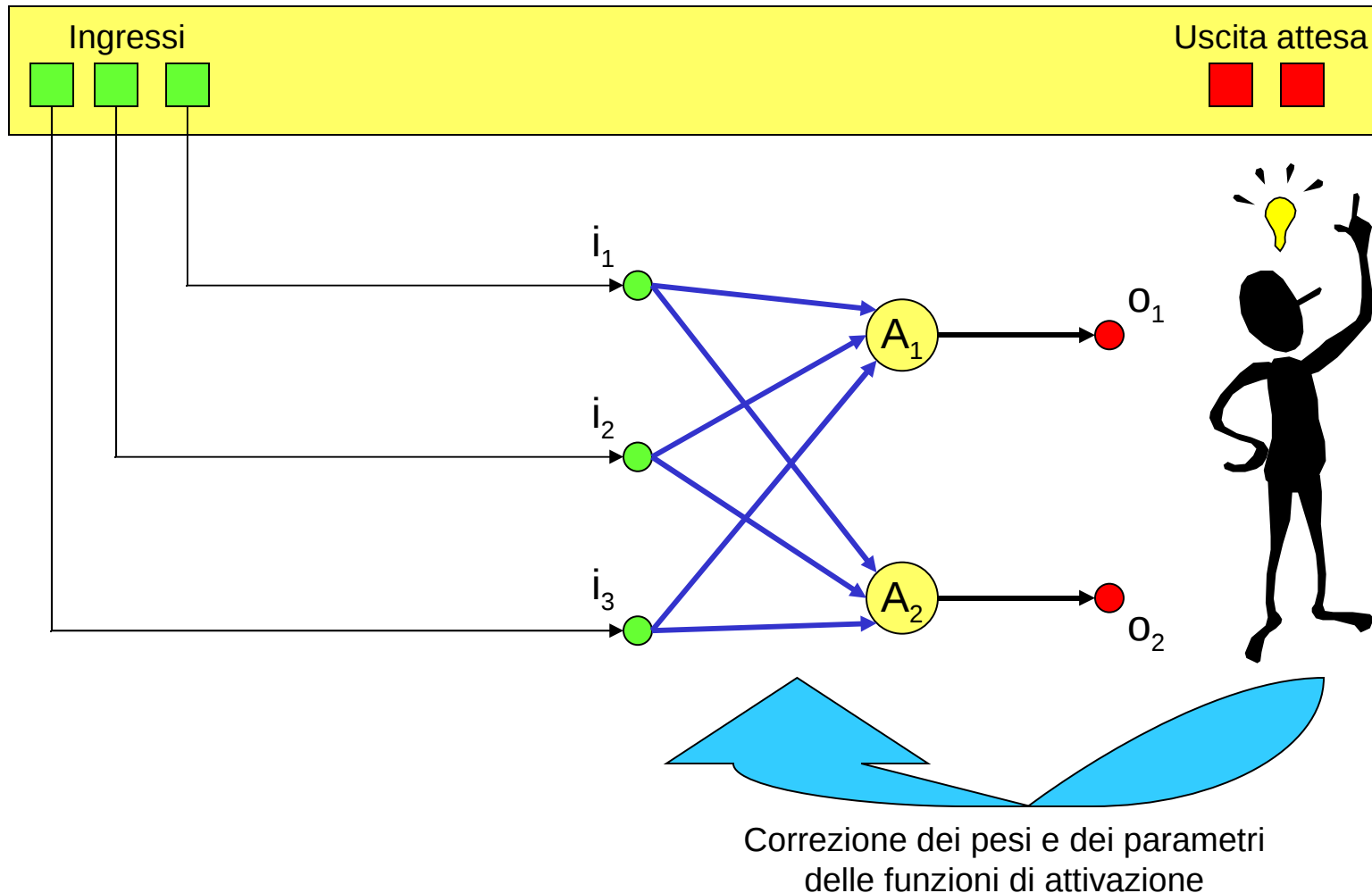
Le Reti di Perceptroni



- L'apprendimento è la modifica dei parametri in funzione dei parametri di input/output

Le Reti di Perceptroni

Pattern di allenamento



I Vari Tipi di Apprendimento

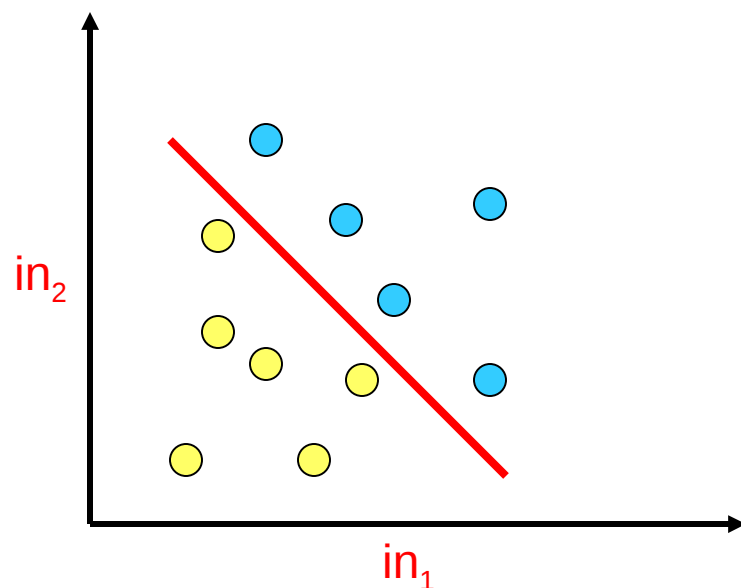
- **Supervisionato** (learning with a teacher)
 - Viene specificato per ogni pattern di input, il pattern desiderato in input
- **Non-supervisionato** (learning without a teacher)
 - I neuroni verranno associati a pattern di ingresso contigui
 - Clustering
 - Mappe neurali

Limiti del Perceptrone

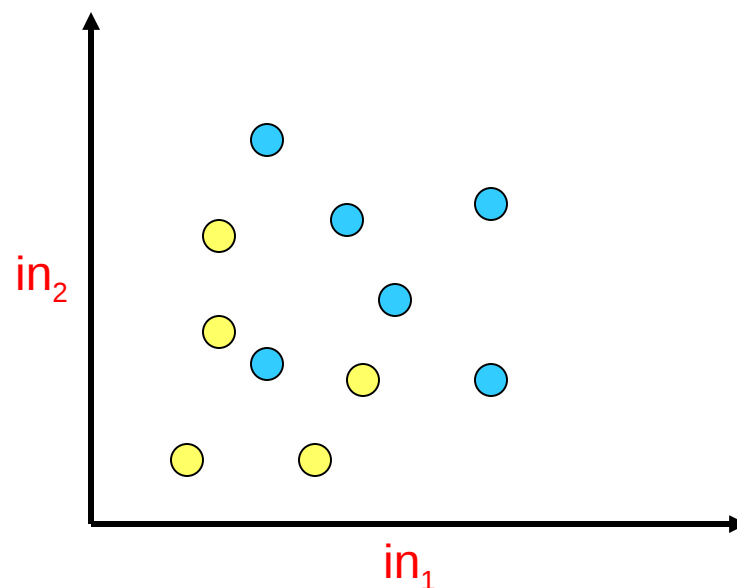
- Per apprendere una classificazione, il problema deve essere **linearmente separabile**
 - I pattern appartenenti alla classe C devono essere contenuti in un semipiano dello spazio d'ingresso
- Con n ingressi, lo spazio d'ingresso diventa n -dimensionale e i pattern vengono separati da un iperpiano

Funzioni Linearmente Separabili

- Le reti di perceptroni sono efficaci soltanto se gli insiemi da classificare sono *linearmente separabili*



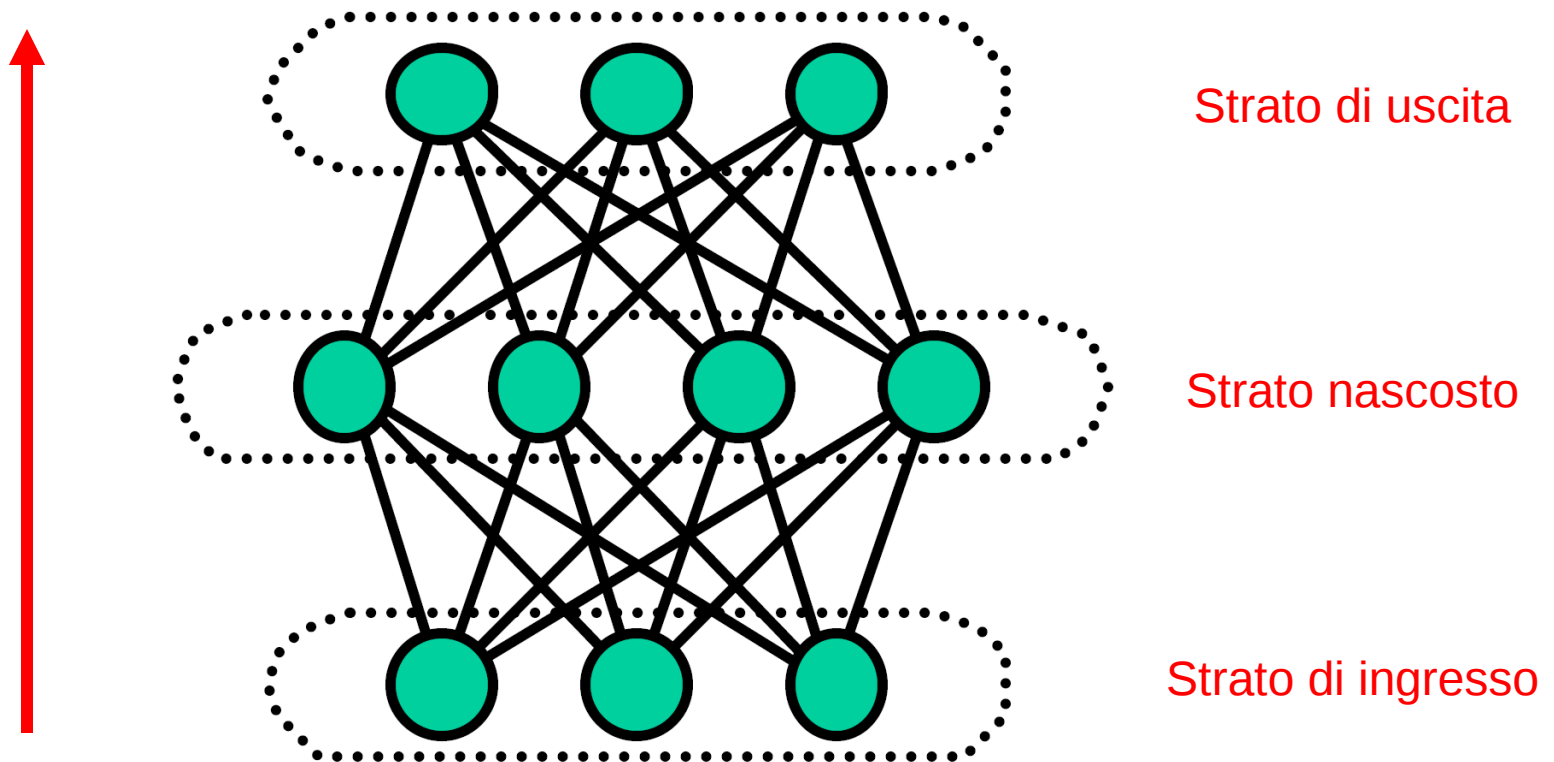
Linearmente separabile



Non linearmente separabile
(non è possibile determinare una retta che
separi i due insiemi)

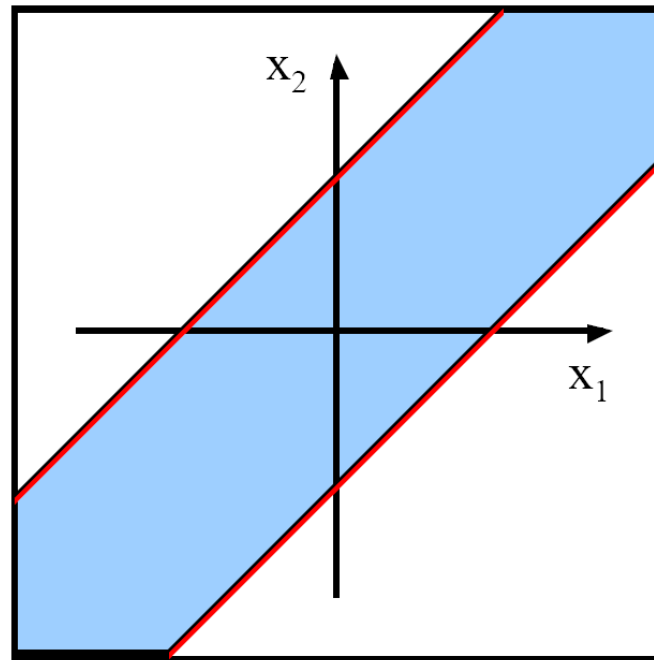
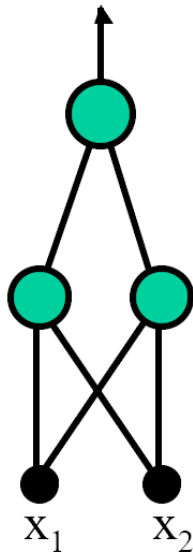
Reti Multistrato

- Tutti i neuroni di uno strato sono connessi con tutti i neuroni dello strato successivo
- Non esistono connessioni tra neuroni dello stesso strato



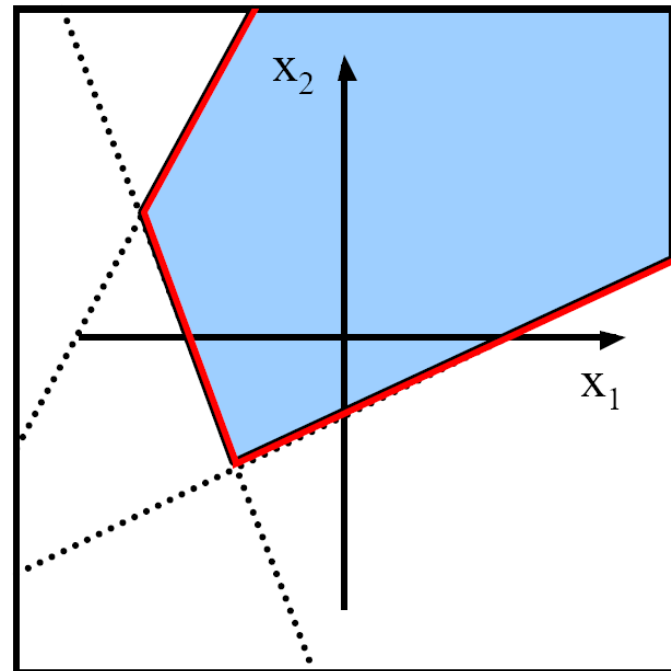
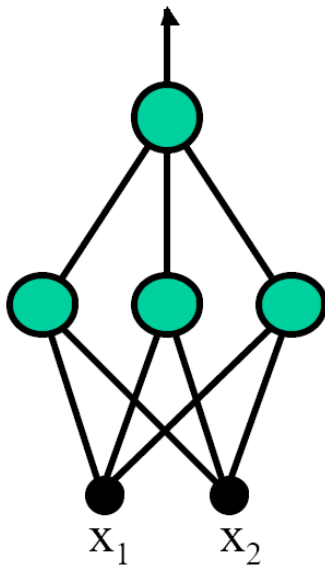
Reti a 3 Strati

- Sono in grado di separare regioni convesse numero di lati \leq numero neuroni nascosti



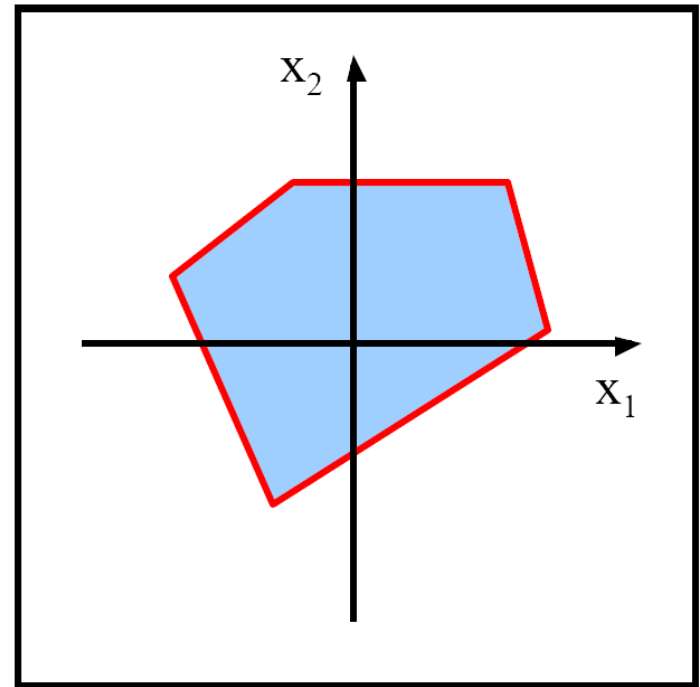
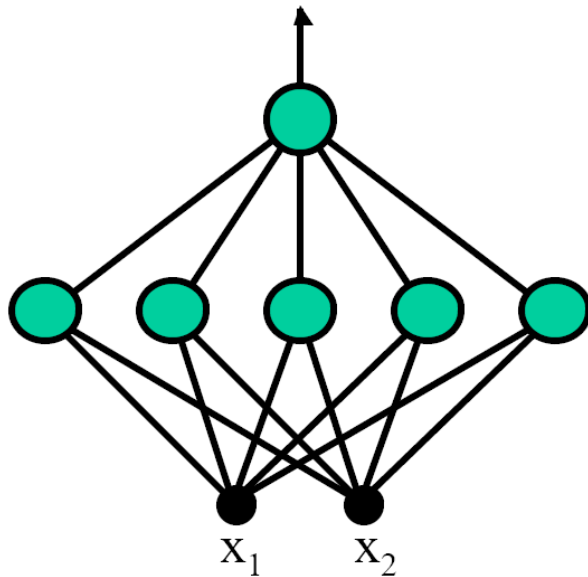
Reti a 3 Strati

- Sono in grado di separare regioni convesse numero di lati \leq numero neuroni nascosti



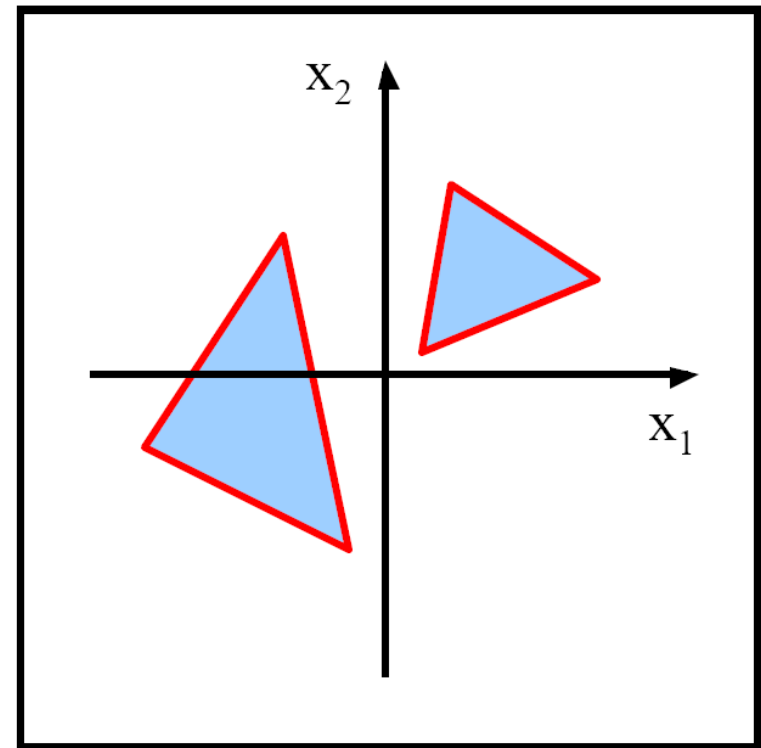
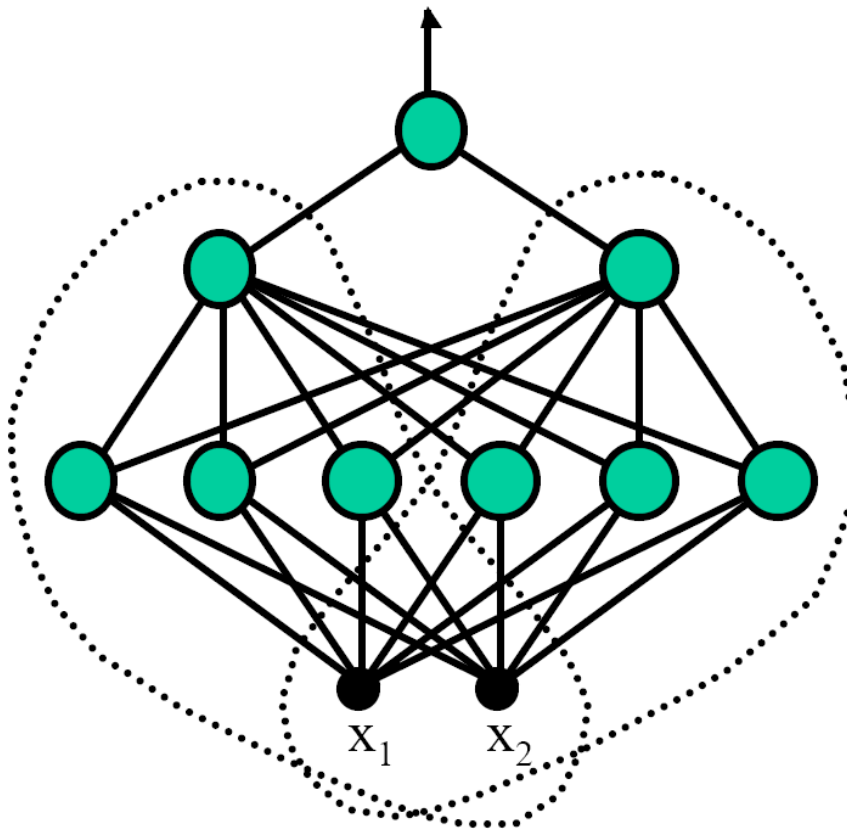
Reti a 3 Strati

- Sono in grado di separare regioni convesse numero di lati \leq numero neuroni nascosti



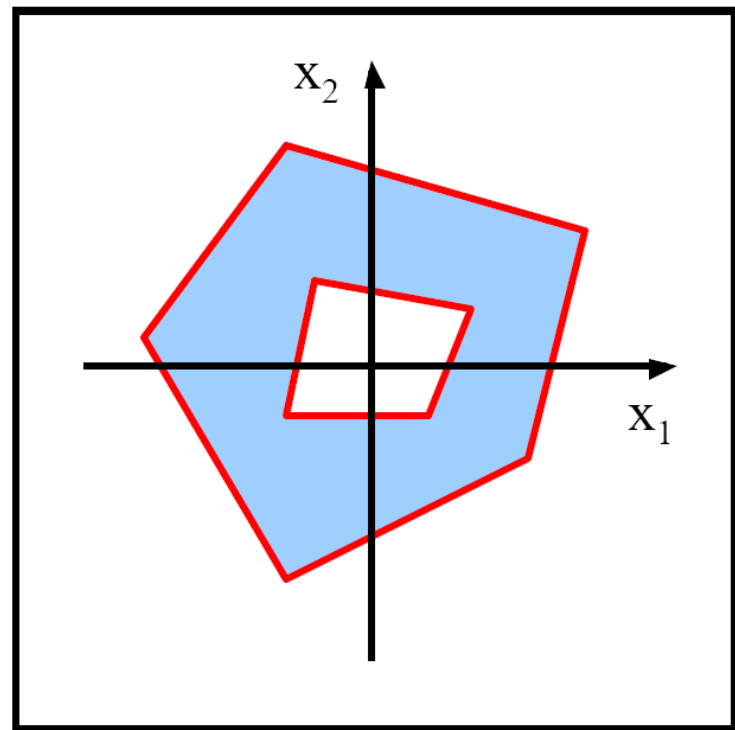
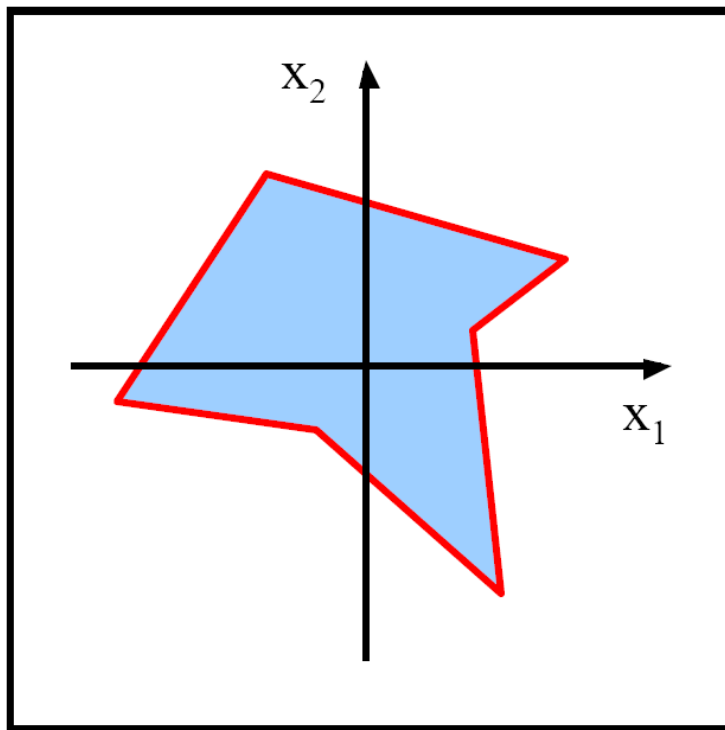
Reti a 4 Strati

- Sono in gradi di separare regioni qualsiasi



Reti a 4 Strati

- L'aggiunta di altri strati non migliora la capacità di classificazione



Implicazioni

- Per effettuare classificazioni complesse, i neuroni devono essere *non lineari* ed essere organizzati su *più strati*
- Problemi
 - Come si addestra una rete multistrato?
 - Qual è l'uscita desiderata dei neuroni nascosti?

Generalizzazione

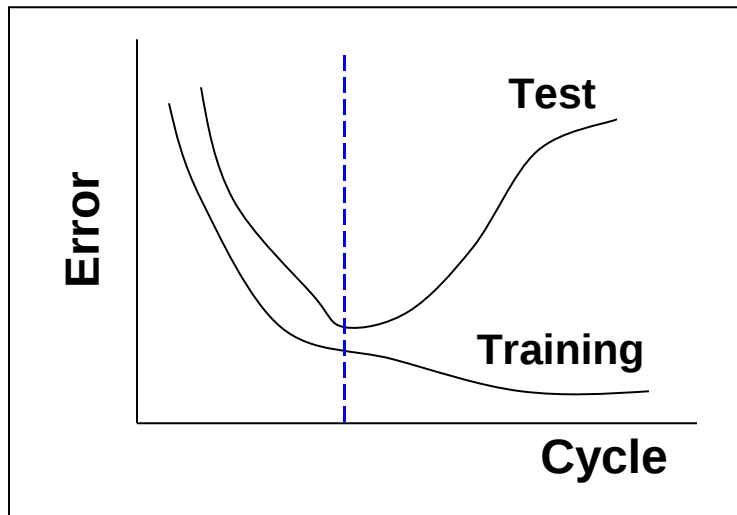
- E' la capacità della rete di riconoscere stimoli leggermente diversi da quelli con cui è stata addestrata
- Per valutare la capacità della rete di generalizzare gli esempi del **TS**, si definisce un altro insieme di esempi, detto *Validation Set* (**VS**)
- Terminato l'apprendimento sul **TS** ($E_{TS} < \varepsilon$), si valuta l'errore sul **VS** (E_{VS})

Evaluation Overfitting

■ Quando il modello è valido?

→ Rischio di Overfitting:

- ✓ Sovrastima sui dati di training: il modello si comporta molto bene sui dati di training solo perchè è diventato troppo specifico



Tipicamente procedendo nel training

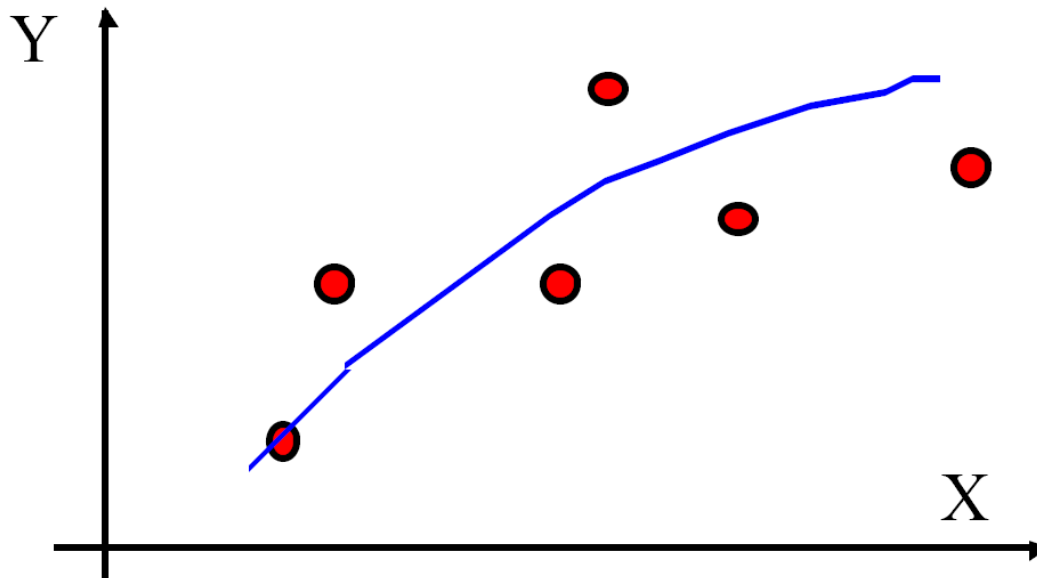
- Errore su dati training continua a diminuire.
- Error su dati di test prima diminuisce e poi cresce

Soluzione:

fermarsi nel training quando l'errore nell'insieme di test inizia a crescere

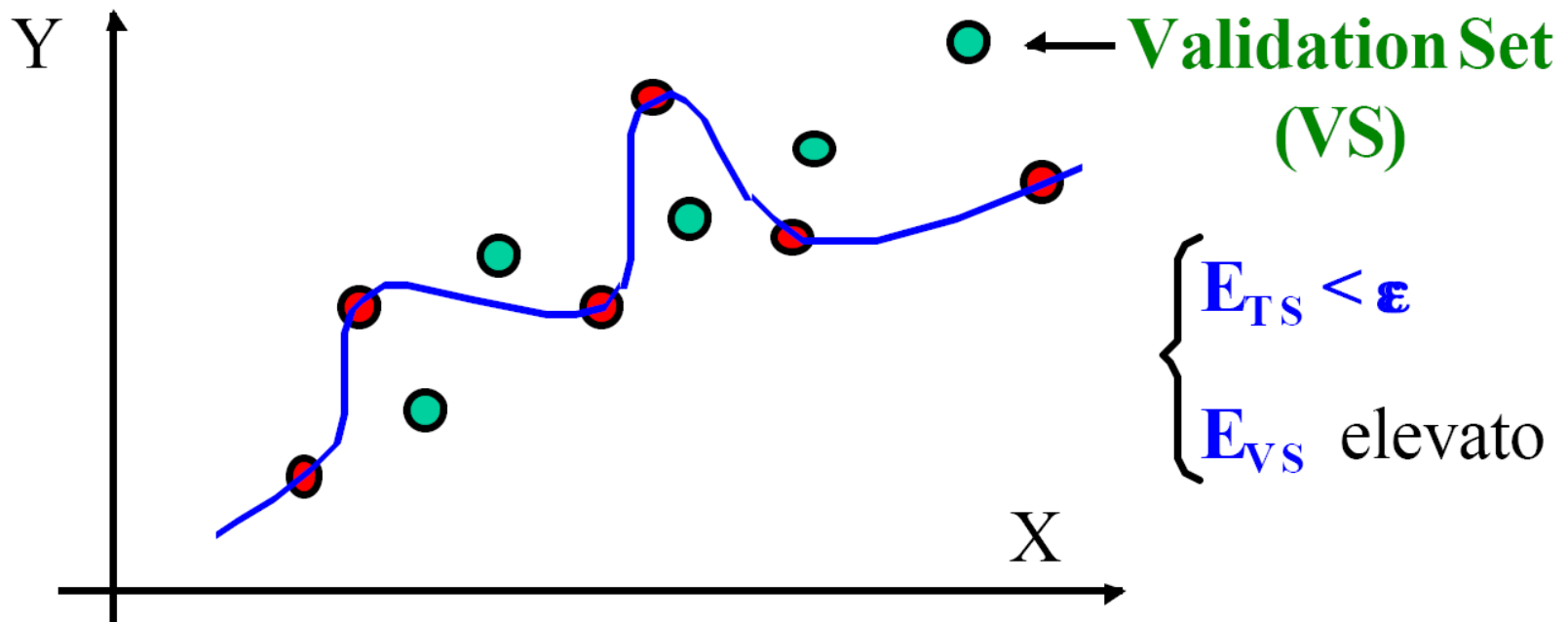
Generalizzazione (1/4)

- Il numero di parametri da regolare dipende dal numero di neuroni nascosti della rete
- Pochi neuroni nascosti potrebbero non essere sufficienti a ridurre l'errore globale



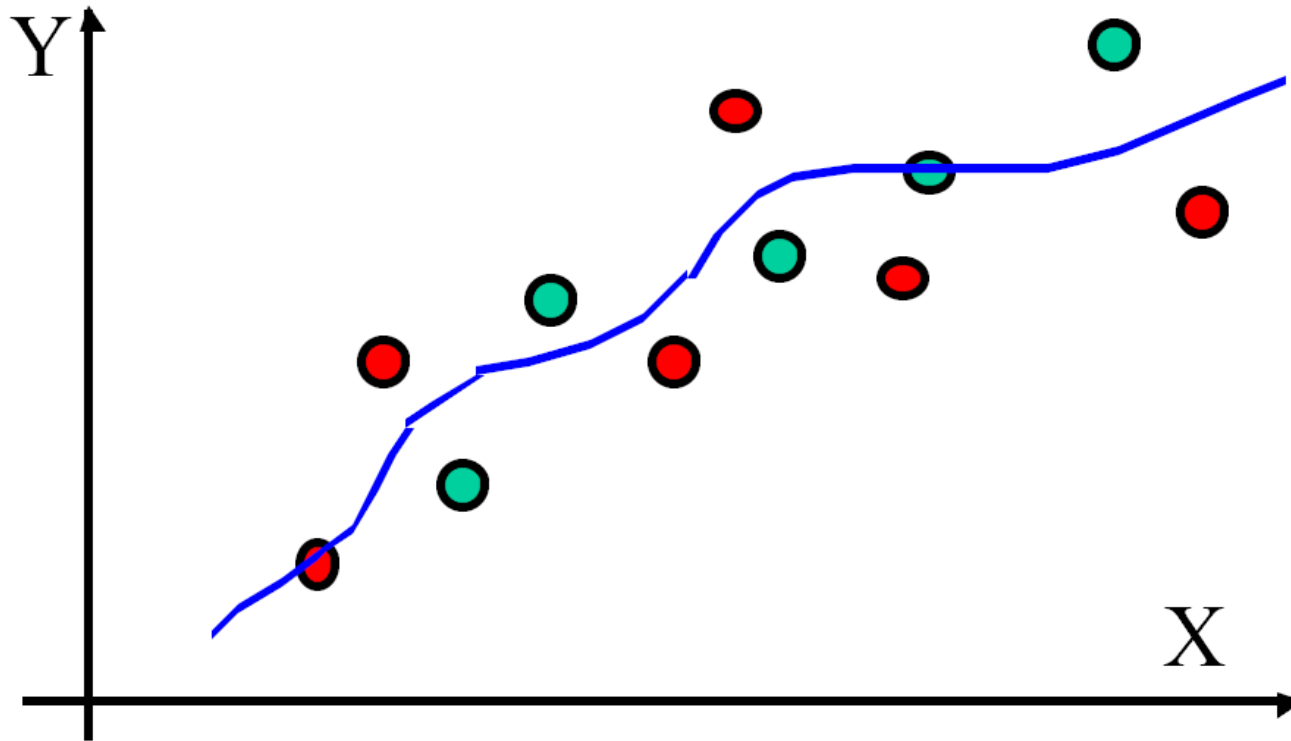
Generalizzazione (2/4)

- Troppi neuroni nascosti potrebbero fossilizzare eccessivamente la rete sugli esempi specifici del **TS**
- La rete risponderebbe bene sul **TS**, ma l'errore sarebbe elevato su altri esempi (*overtraining*)

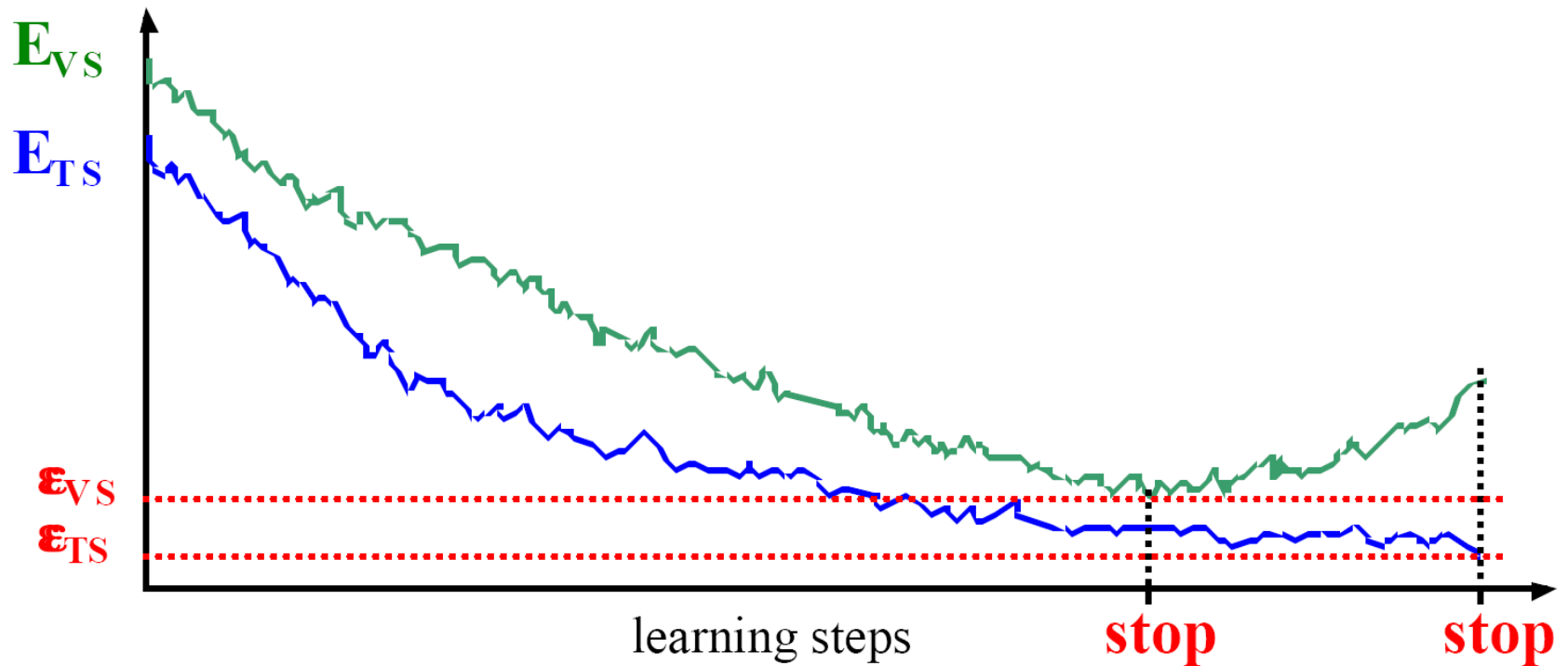


Generalizzazione (3/4)

- Per migliorare la capacità di generalizzazione si può addestrare la rete sul **TS**, monitorare l'errore sul **VS** (E_{VS}) e fermare l'apprendimento quando $E_{VS} < \epsilon_{VS}$



Generalizzazione (4/4)



Sommario

■ Algoritmi Genetici

- Introduzione e applicazioni
- Terminologia
- Mutazione, ricombinazione, selezione
- Problemi e regole euristiche

■ Reti Neurali Artificiali

- Introduzione e applicazioni
- Differenze con computer seriale
- Perceptrone e perceptrone multistrato
- Valutazione