Unione europea
Fondo sociale europeo

RegioneEmilia-Romagna

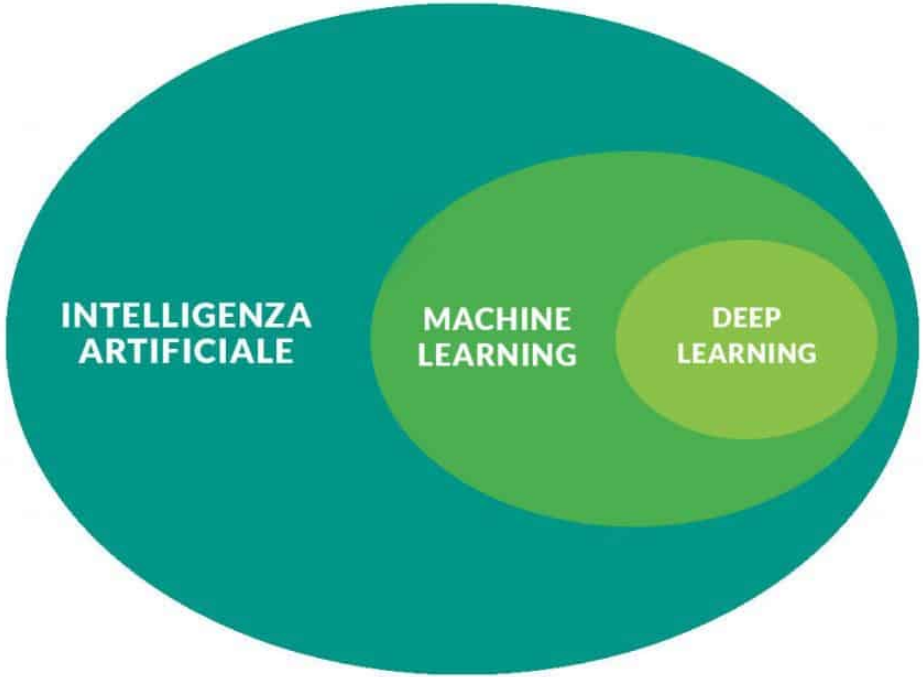
RETE POLITECNICA
La tecnica per crescere

Materiale didattico per partecipante al corso "**TECNICO ESPERTO NELL'ANALISI E NELLA VISUALIZZAZIONE DEI DATI**" – Rif.P.A. 2021-15998/RER – approvata con DGR n. 1263 del 02/08/2021 di IFOA – Istituto Formazione Operatori Aziendali

1.
INTRODUZIONE AL MODULO

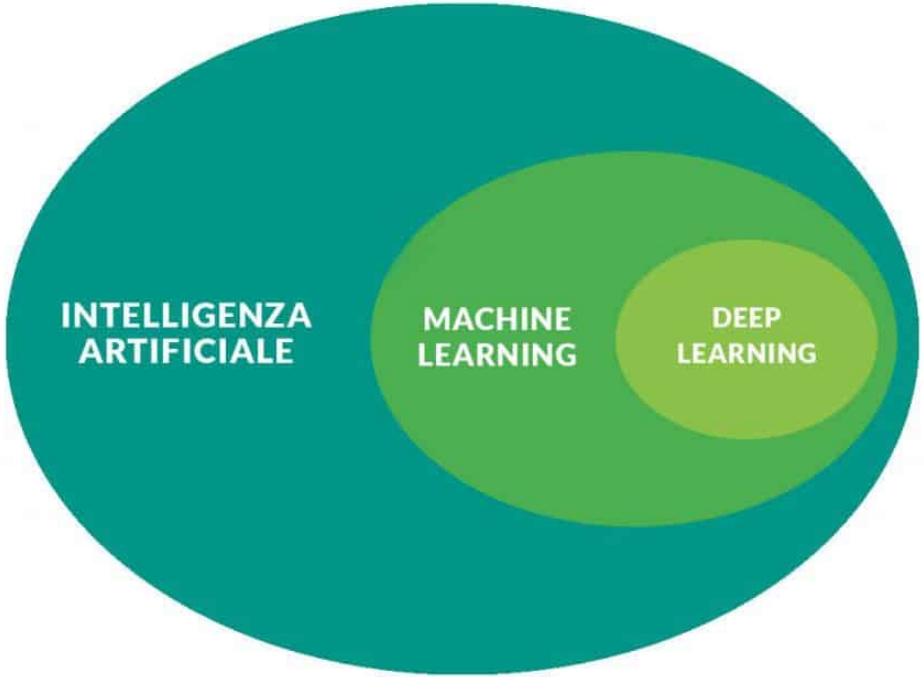
Cosa si dice del ML

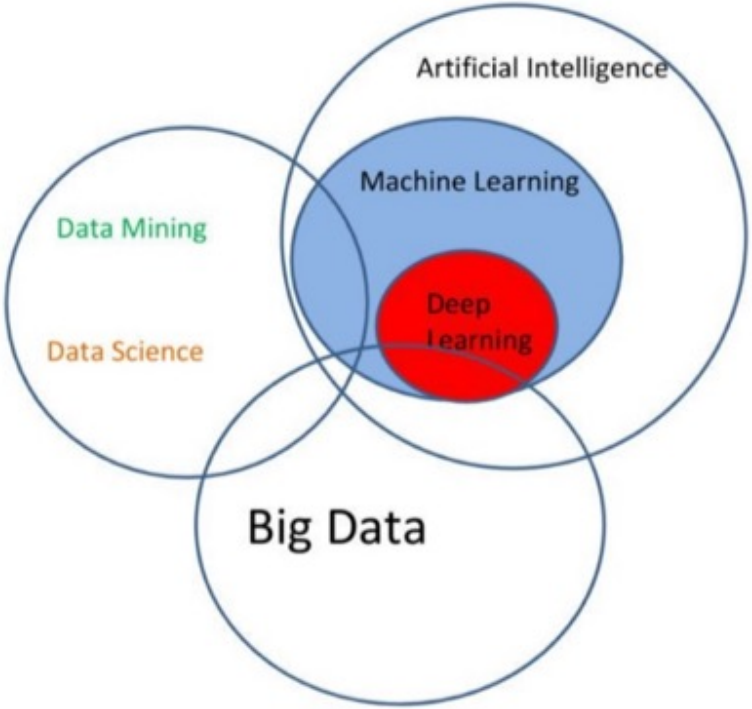
Dove si colloca nel panorama delle discipline



Cosa si dice del ML

Dove si colloca nel panorama delle discipline





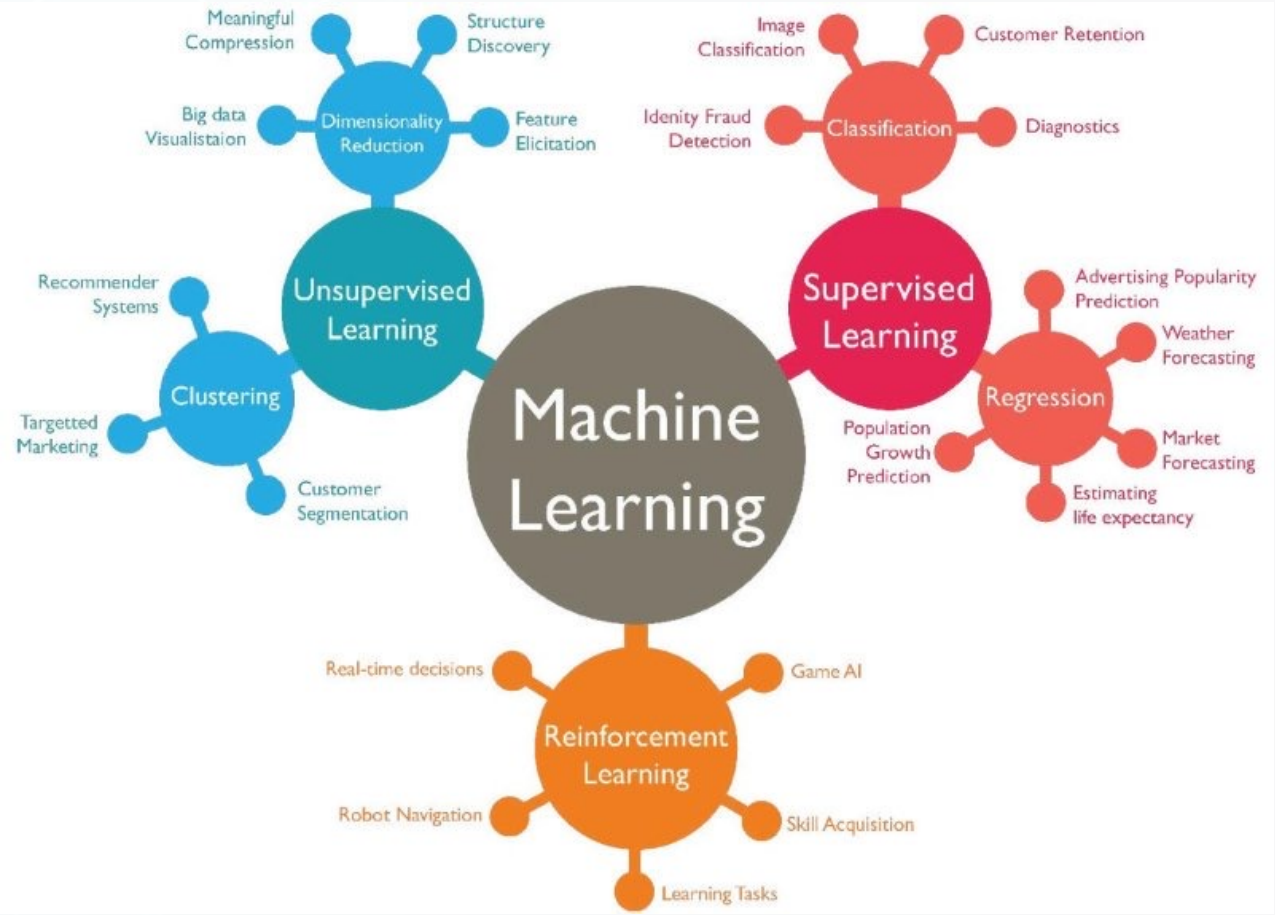
Cosa si dice del ML

Le principali diramazioni:

1. **Supervised**: apprendimento da un dataset ‘etichettato’ (labeled), cioè di cui si hanno i target

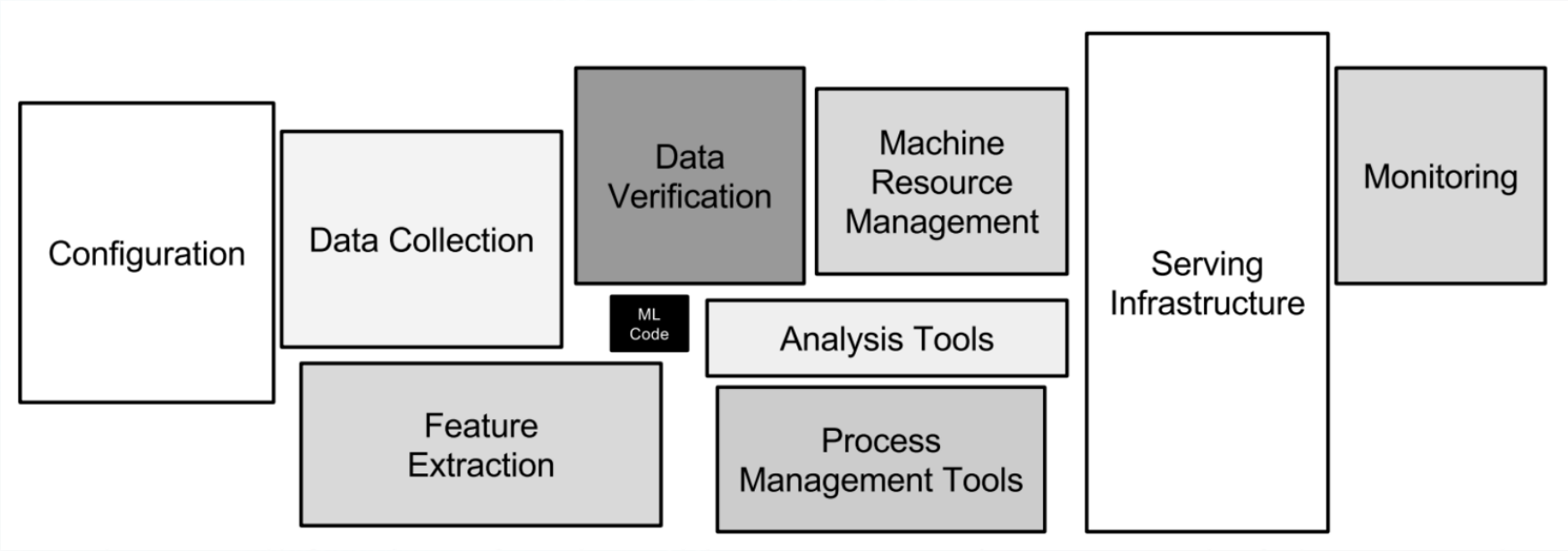
2. **Unsupervised**: ricerca di pattern in dataset senza target

3. **Reinforcement Learning**: apprendimento tramite meccanismi di feedback o ‘ricompensa’



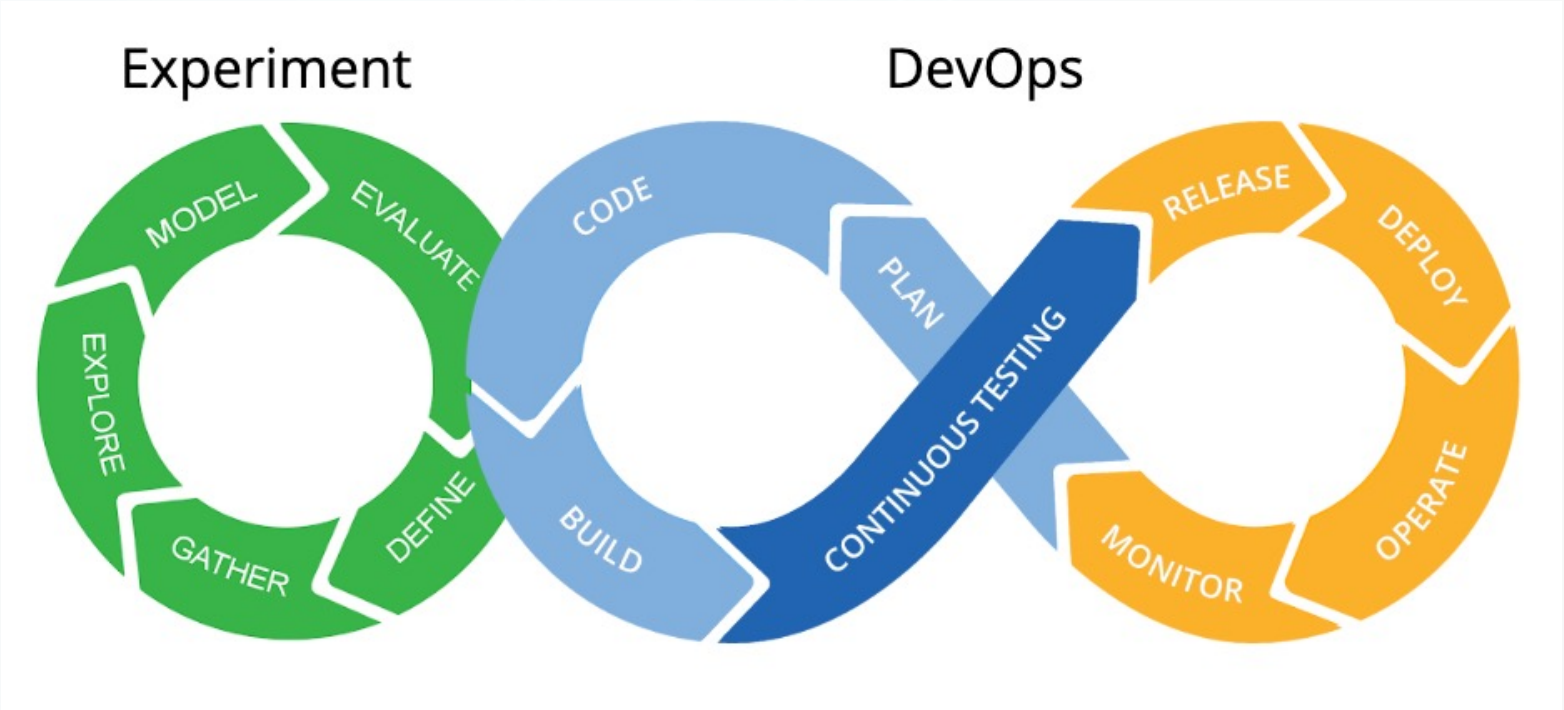
Cosa si dice del ML

In una struttura operativa, la parte di ML vero e proprio occupa una piccolissima parte



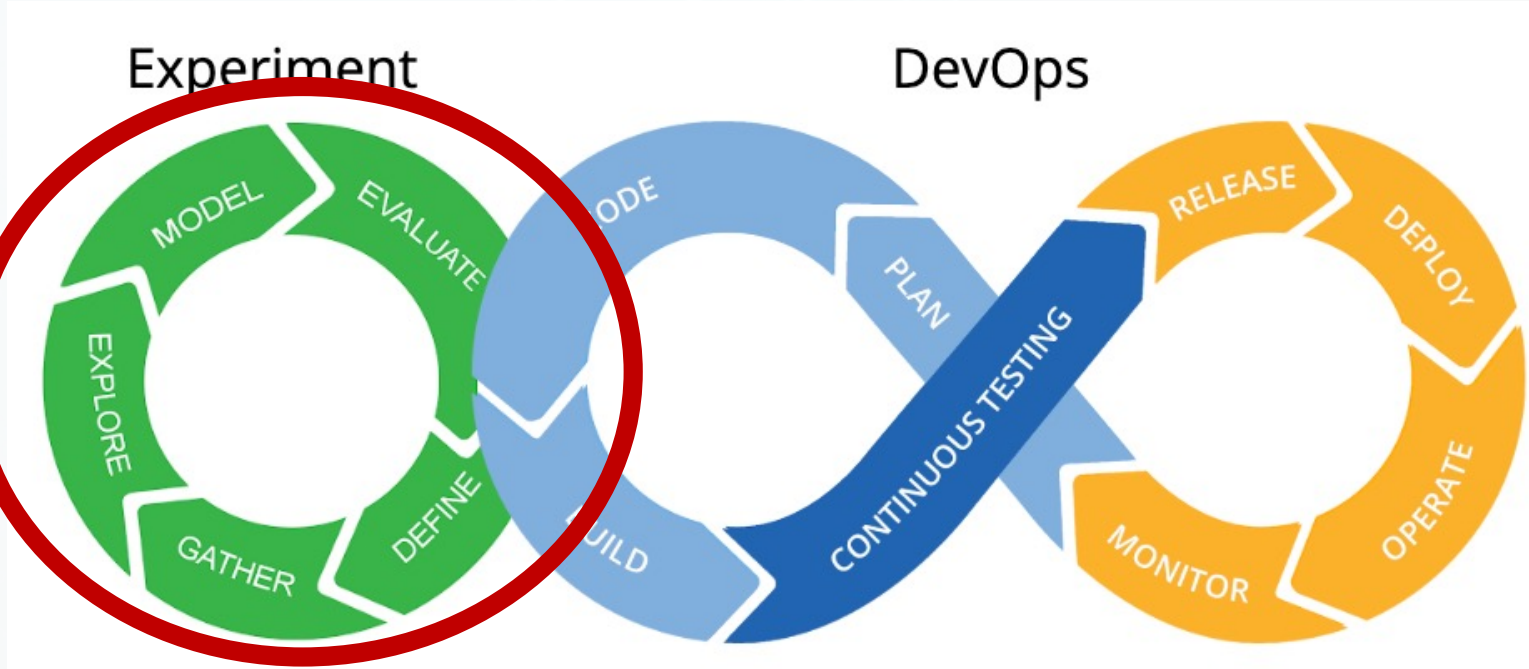
Cosa si dice del ML

Il Machine Learning è tipicamente un **processo ciclico**, affiancato e strutturato sulla base di una serie di Development Operations (DevOps), conosciute ad oggi come MLops



Cosa si dice del ML

Il Machine Learning è tipicamente un **processo ciclico**, affiancato e strutturato sulla base di una serie di Development Operations (DevOps), conosciute ad oggi come MLops



In questo corso ci occuperemo solo della parte di Experiment

Argomenti del modulo di ML

1. Supervised ML: problemi di regressione
2. Unsupervised ML: problemi di clustering
3. Reinforcement learning (?)

Toccheremo i principali snodi di un progetto di Machine Learning:

Dataset Exploration:
- analisi grafiche
- analisi statistiche

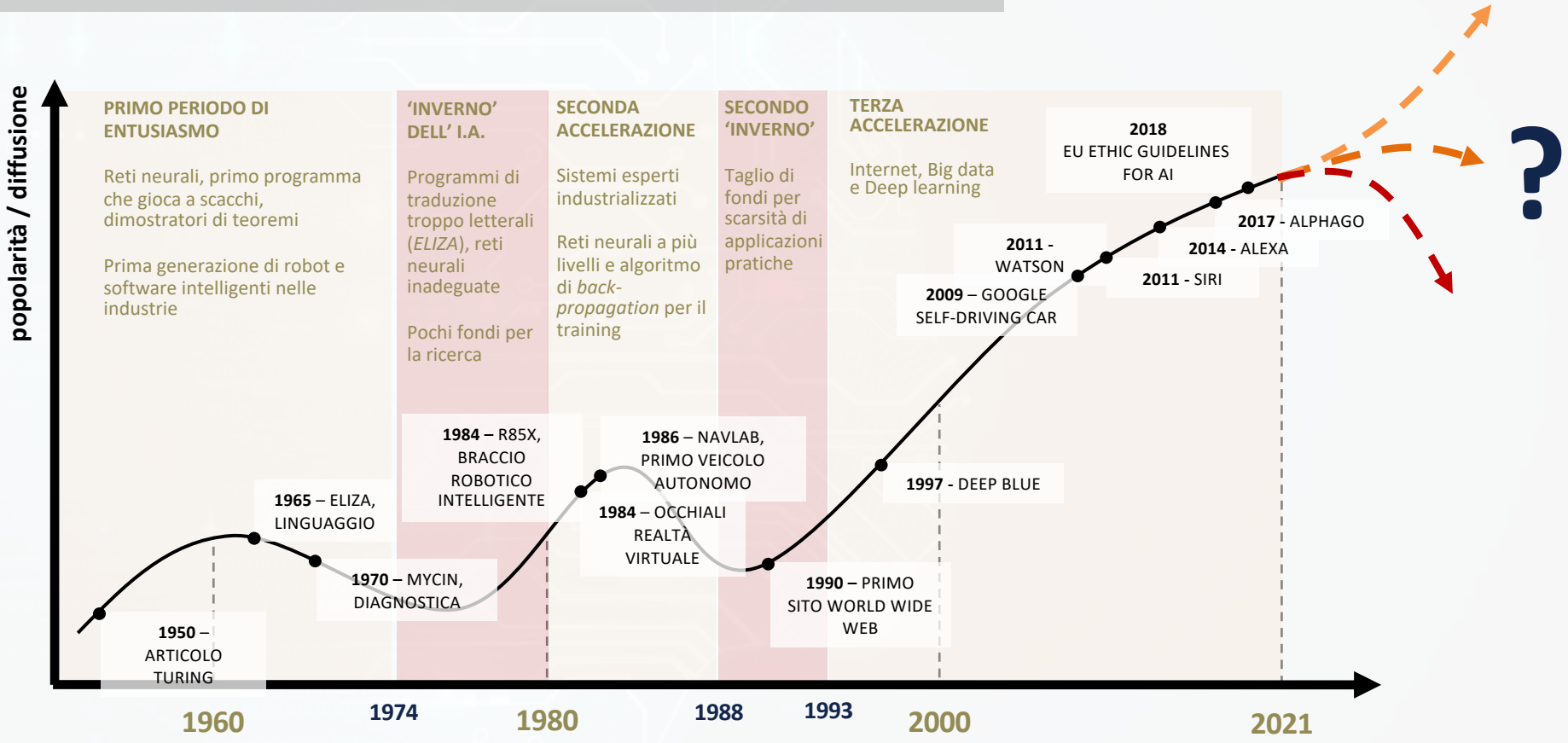
Pre-processing:
- Gestione valori mancanti
- Normalizzazione
- Feature selection / dimensionality reduction

Model selection:
- Regressione: regressioni Lineari / Alberi Decisionali / GradientBoosting…
- Clustering: K-means / Hierarchical clustering…

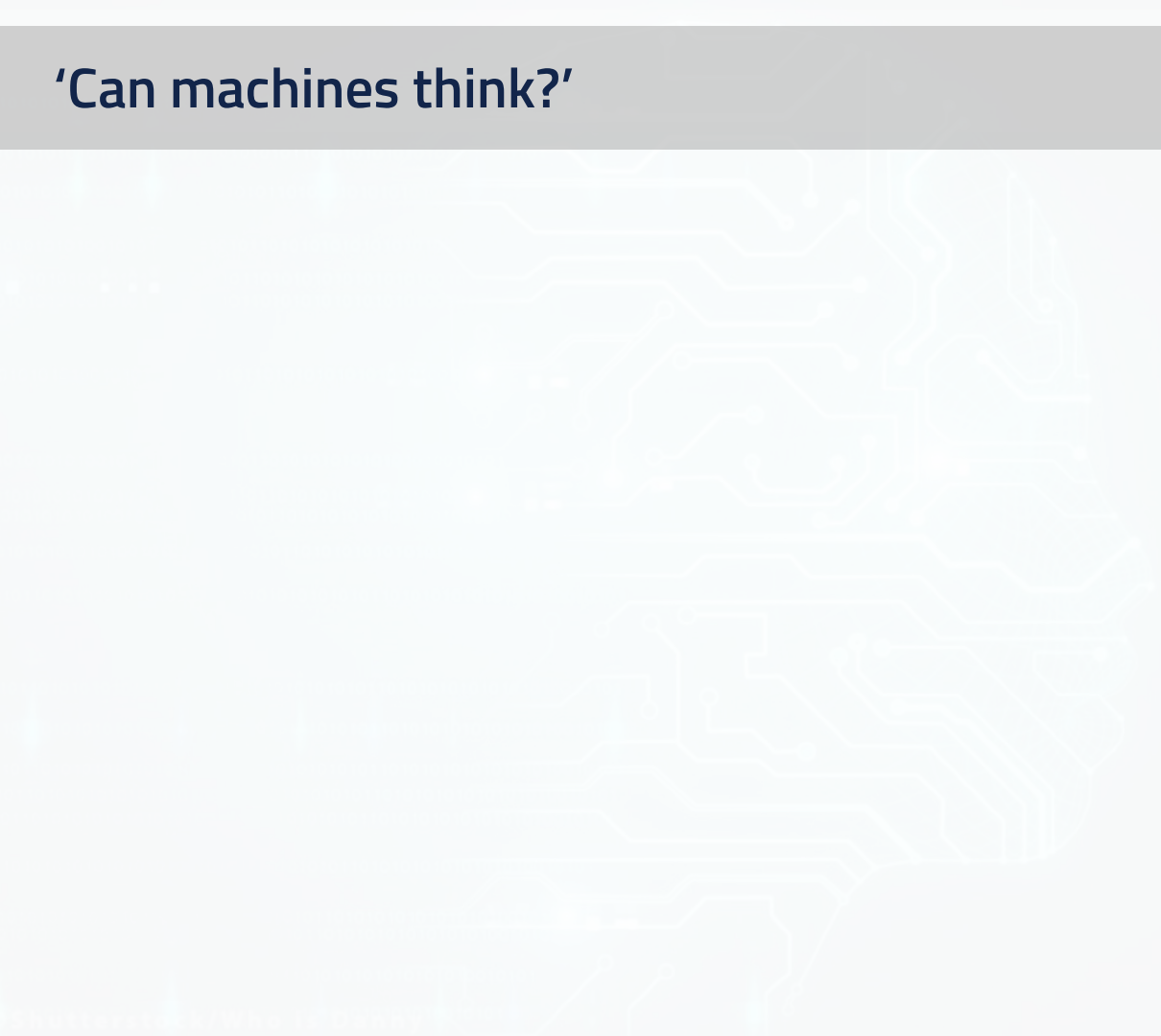
Model training/testing:
- evaluation metrics
- overfitting / cross-validation

2.
I.A. &
MACHINE LEARNING

I.A. – BREVE STORIA



‘Can machines think?’



‘Can machines think?’

‘Le macchine possono pensare’?

Londra, 1950

“Credo che la domanda […] sia troppo priva di senso per meritare una discussione.

Ciononostante, credo che alla fine del secolo l’uso delle parole e l’opinione corrente saranno talmente mutate che chiunque potrà parlare di machine pensanti senza aspettarsi di essere contraddetto” *

*_Mind_, Volume LIX, Issue 236, October 1950, Pages 433–460, https://doi.org/10.1093/mind/LIX.236.433

‘Le macchine possono pensare’?

Londra, 1950

“Credo che la domanda […] sia troppo priva di senso per meritare una discussione.

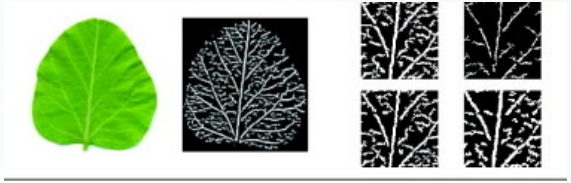
Ciononostante, credo che alla fine del secolo l’uso delle parole e l’opinione corrente saranno talmente mutate che chiunque potrà parlare di machine pensanti senza aspettarsi di essere contraddetto” *

Alan M. Turing

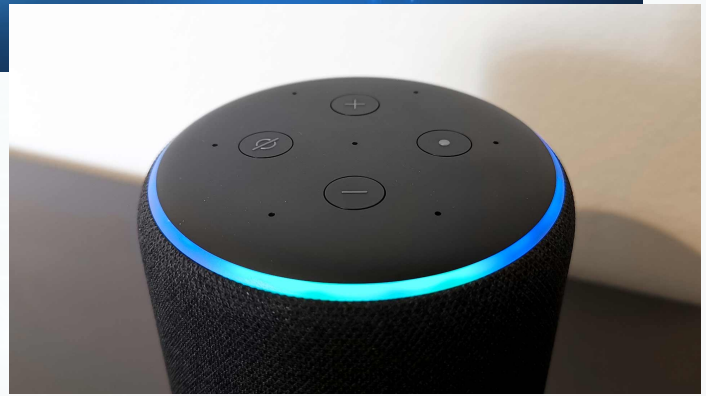
**Mind*, Volume LIX, Issue 236, October 1950, Pages 433–460, https://doi.org/10.1093/mind/LIX.236.433

I.A. – APPLICAZIONI DEL PRESENTE

RICONOSCIMENTO IMMAGINI



I.A. – APPLICAZIONI DEL PRESENTE



RICONOSCIMENTO IMMAGINI

INTERPRETAZIONE DEL LINGUAGGIO

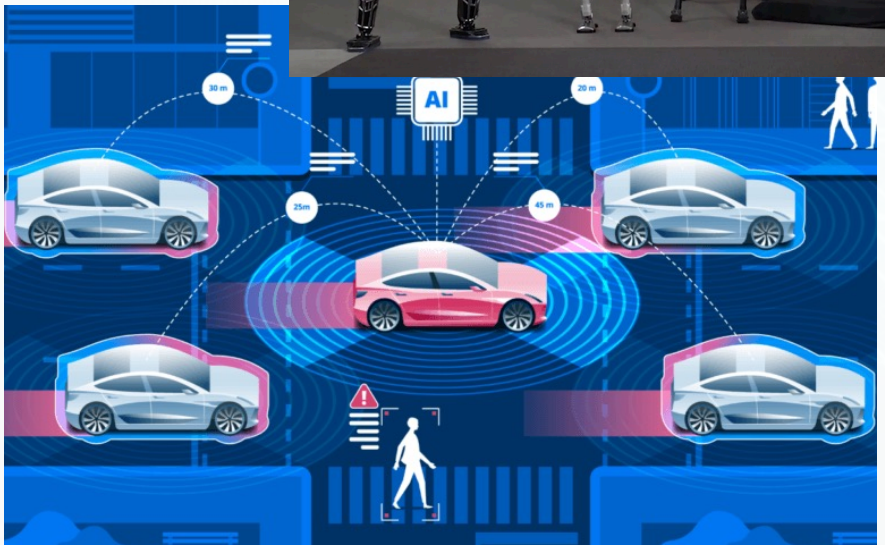
I.A. – APPLICAZIONI DEL PRESENTE

RICONOSCIMENTO IMMAGINI

INTERPRETAZIONE DEL LINGUAGGIO

DECISION MAKING

I.A. – APPLICAZIONI DEL PRESENTE



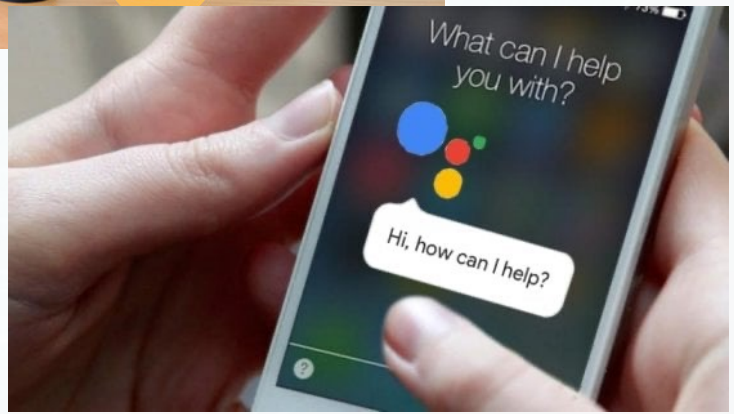
RICONOSCIMENTO IMMAGINI

INTERPRETAZIONE DEL LINGUAGGIO

DECISION MAKING

ROBOTICA E SISTEMI AUTONOMI

I.A. – APPLICAZIONI DEL PRESENTE



RICONOSCIMENTO IMMAGINI

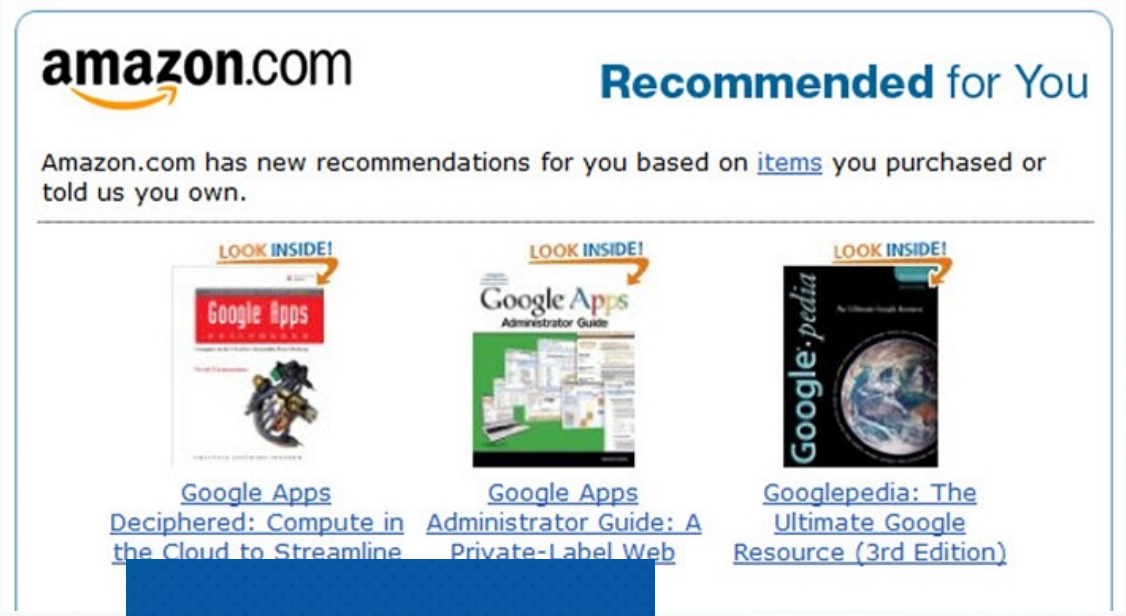
INTERPRETAZIONE DEL LINGUAGGIO

DECISION MAKING

ROBOTICA E SISTEMI AUTONOMI

ASSISTENTI VIRTUALI

I.A. – APPLICAZIONI DEL PRESENTE



RICONOSCIMENTO IMMAGINI

INTERPRETAZIONE DEL LINGUAGGIO

DECISION MAKING

ROBOTICA E SISTEMI AUTONOMI

ASSISTENTI VIRTUALI

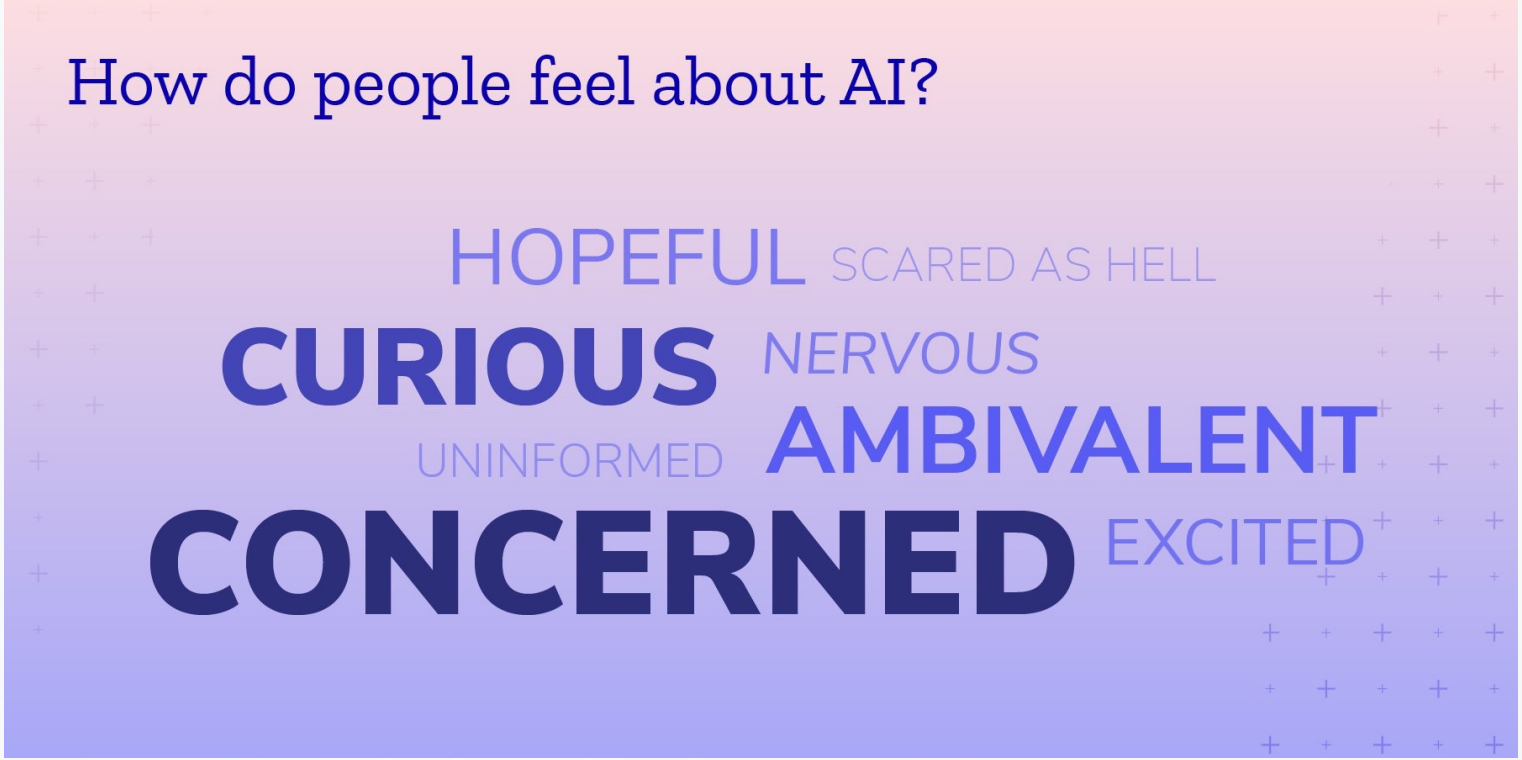
RECOMMENDATION

I.A. – APPLICAZIONI DEL PRESENTE

- RICONOSCIMENTO IMMAGINI
- INTERPRETAZIONE DEL LINGUAGGIO
- DECISION MAKING
- ROBOTICA E SISTEMI AUTONOMI
- ASSISTENTI VIRTUALI
- RECOMMENDATION
- SMART OBJECTS
- ...

I.A. - COME LA PERCEPIAMO OGGI

Da un sondaggio del 2019 (circa 70.000 partecipanti da tutto il mondo)

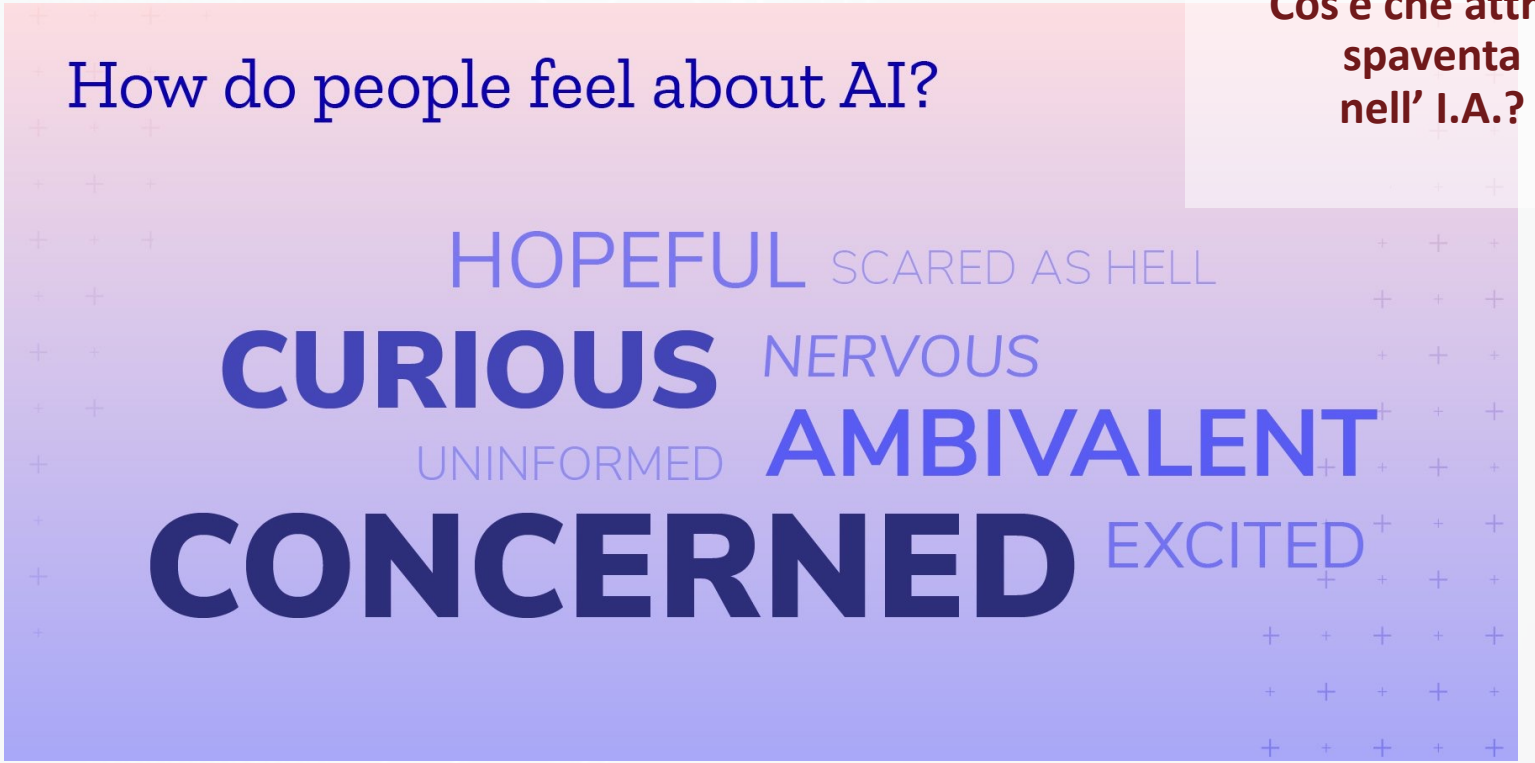


https://foundation.mozilla.org/en/blog/we-asked-people-around-the-world-how-they-feel-about-artificial-intelligence-heres-what-we-learned/

I.A. - COME LA PERCEPIAMO OGGI

Da un sondaggio del 2019 (circa 70.000 partecipanti da tutto il mondo)

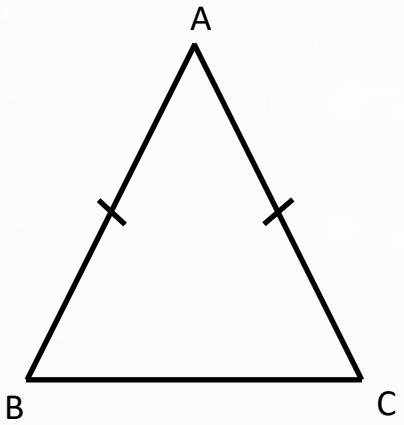
Cos'è che attre e spaventa nell' I.A.?



https://foundation.mozilla.org/en/blog/we-asked-people-around-the-world-how-they-feel-about-artificial-intelligence-heres-what-we-learned/

I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA

IL ‘PONS ASINORUM’

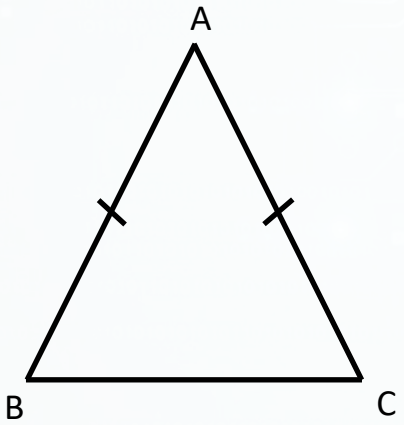


Dato un triangolo isoscele (AB = AC), dimostrare che:

gli angoli alla base sono congruenti.

I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA

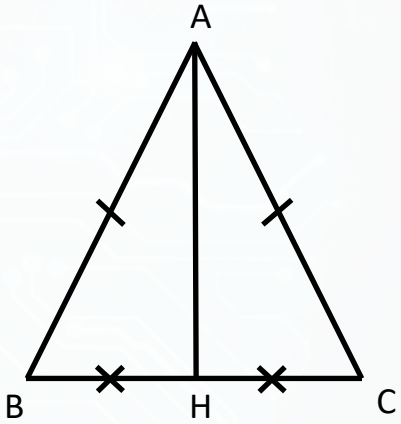
IL ‘PONS ASINORUM’



Dato un triangolo isoscele (AB = AC), dimostrare che:

gli angoli alla base sono congruenti.

DIMOSTRAZIONE STANDARD

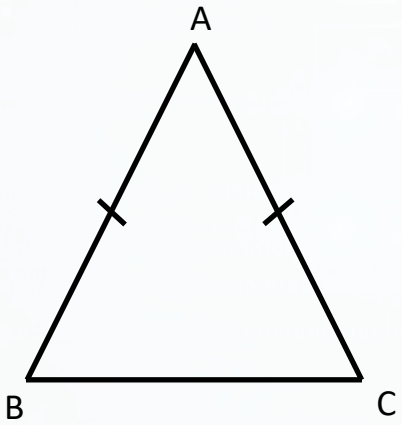


Avendo 3 lati congruenti per costruzione,
il triangolo AHB e il triangolo AHC sono congruenti

→ anche gli angoli saranno congruenti

I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA

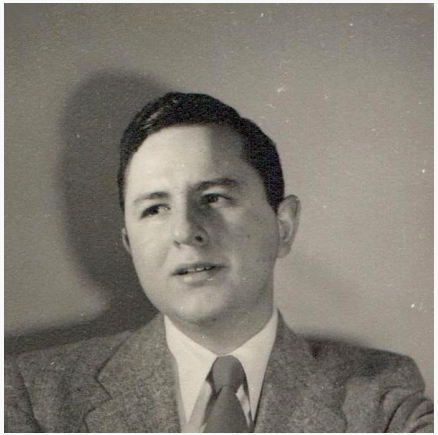
IL ‘PONS ASINORUM’



Dato un triangolo isoscele (AB = AC), dimostrare che:

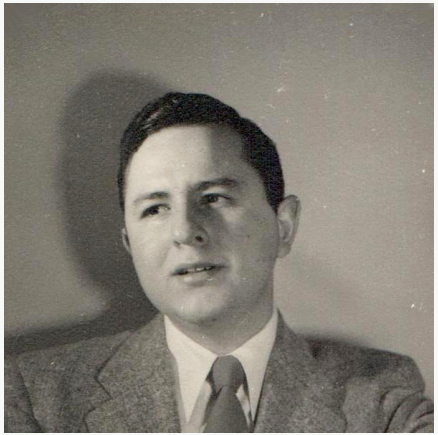
gli angoli alla base sono congruenti.

H. GELERNTER, 1959
prima macchina per la dimostrazione di teoremi su un computer IBM *



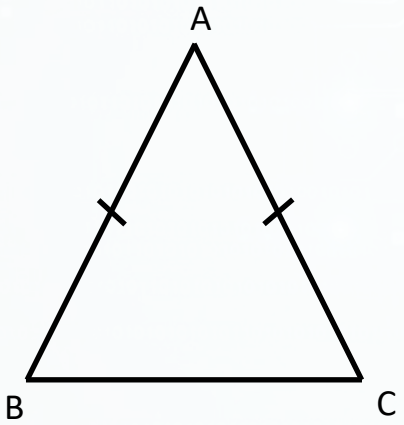
* H. Gelernter, "Realization of a Geometry Theorem-Proving Machine", Proc. of the International Conference on Information Pro
1959

I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA



H. GELERNTER, 1959
prima macchina per la dimostrazione di teoremi su un computer IBM *

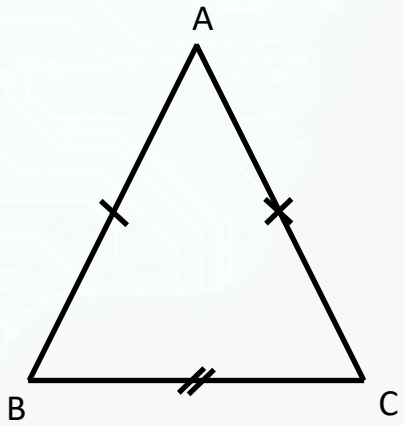
IL 'PONS ASINORUM'



Dato un triangolo isoscele (AB = AC), dimostrare che:

gli angoli alla base sono congruenti.

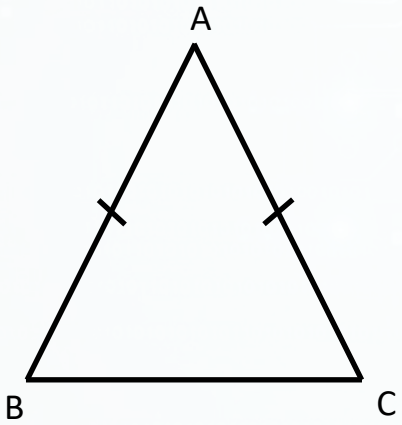
DIMOSTRAZIONE PROPOSTA DALLA MACCHINA



* H. Gelernter, "Realization of a Geometry Theorem-Proving Machine", Proc. of the International Conference on Information Pro
1959

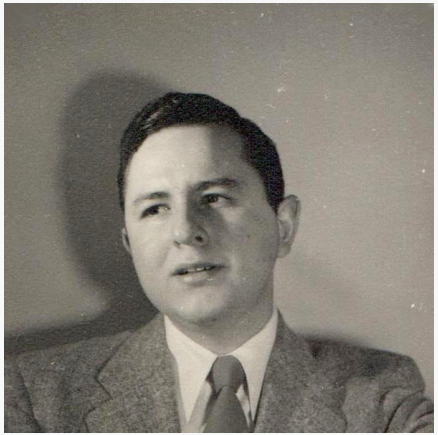
I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA

IL 'PONS ASINORUM'



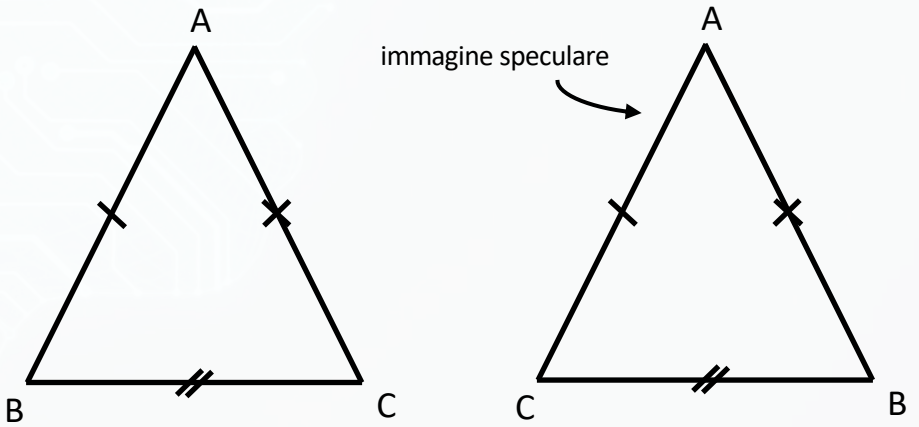
Dato un triangolo isoscele (AB = AC), dimostrare che:

gli angoli alla base sono congruenti.



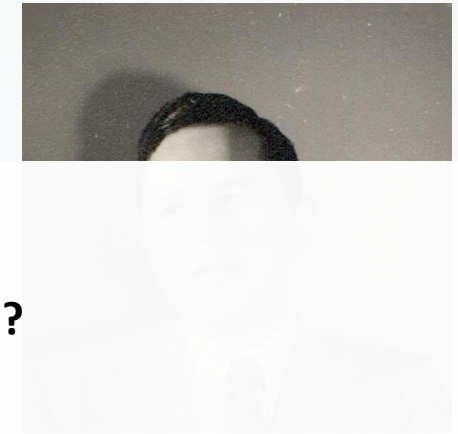
H. GELERNTER, 1959
prima macchina per la dimostrazione di teoremi su un computer IBM *

DIMOSTRAZIONE PROPOSTA DALLA MACCHINA



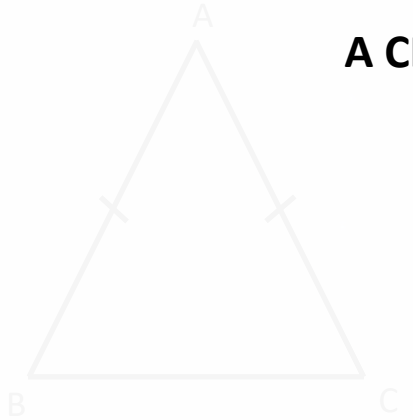
* H. Gelernter, "Realization of a Geometry Theorem-Proving Machine", Proc. of the International Conference on Information Pro
1959

I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA



A CHI VA IL MERITO DELLA DIMOSTRAZIONE?

IL 'PONS ASINORUM'



Dato un triangolo isoscele (AB = AC), dimostrare che:

gli angoli alla base sono uguali.

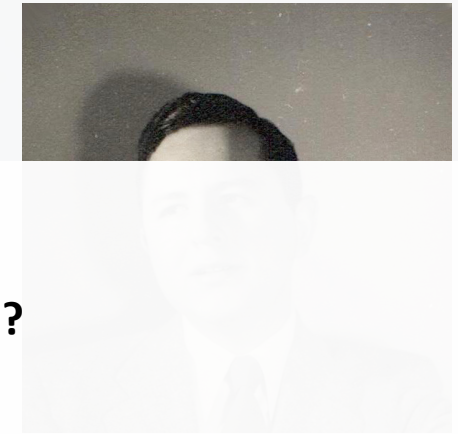
H. GELERNTER, 1959

dimostrazione di teoremi su un computer IBM *

DIMOSTRAZIONE PROPOSTA DALLA MACCHINA

* H. Gelernter, "Realization of a Geometry Theorem-Proving Machine", Proc. of the International Conference on Information Pro
1959

I.A. CREATIVA (1) - GEOMETRIA EUCLIDEA



A CHI VA IL MERITO DELLA DIMOSTRAZIONE?

Profondità di una soluzione

- la soluzione era già nascosta nella mente del programmatore e il programma l'ha solo svelata?

- quanto è facile capire perchè il programma fece ciò che fece?

* H. Gelernter, "Realization of a Geometry Theorem-Proving Machine", Proc. of the International Conference on Information Pro
1959

I.A. CREATIVA

COSA È LA CREATIVITÀ?

COSA SIGNIFICA ‘PENSANTE’?

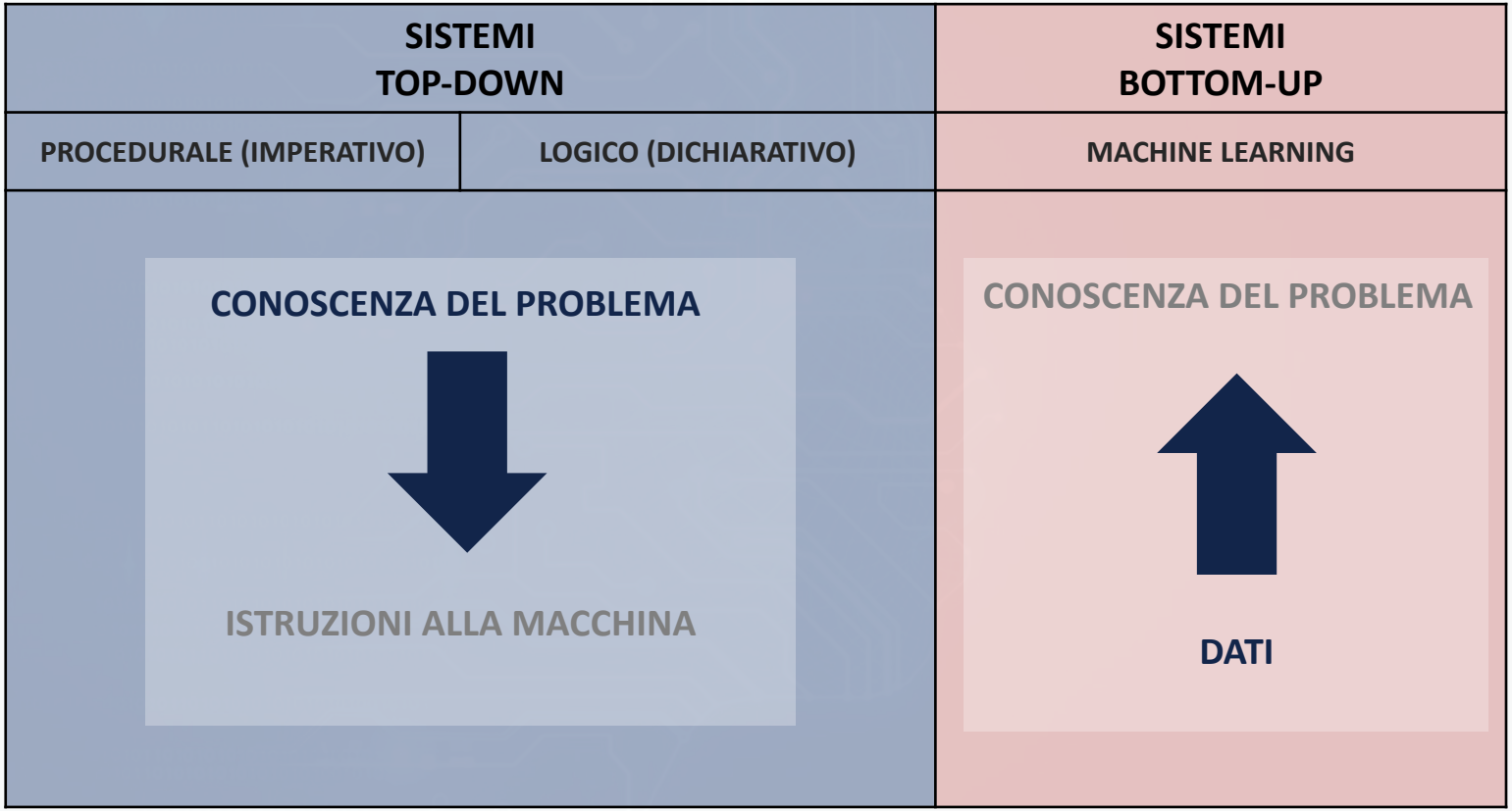
QUAL È IL LIMITE OLTRE IL QUALE LA CAPACITÀ DIVENTA GENIALITÀ?

‘COME’ PENSA UNA MACCHINA ?

I.A. – COME PENSA UNA MACCHINA

SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
PROCEDURALE (IMPERATIVO)	LOGICO (DICHIARATIVO)	MACHINE LEARNING

I.A. – COME PENSA UNA MACCHINA



I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore			

I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare		

I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare	creare un **sistema logico** dal quale la macchina possa inferire la soluzione a un problema	

I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare	creare un **sistema logico** dal quale la macchina possa inferire la soluzione a un problema	

- problemi **ben conosciuti**

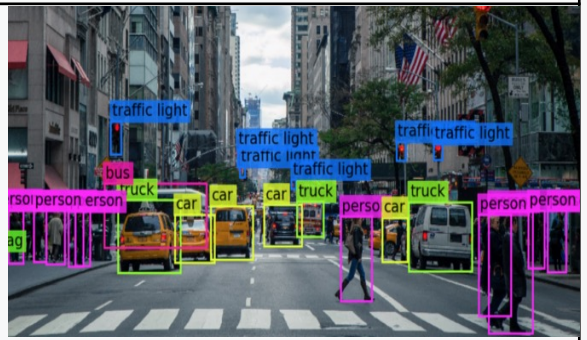
I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare	creare un **sistema logico** dal quale la macchina possa inferire la soluzione a un problema	creare un **database** di dati dai quali la macchina (tipicamente una rete neurale) possa estrarre conoscenza significativa
	▪ problemi **ben conosciuti**		

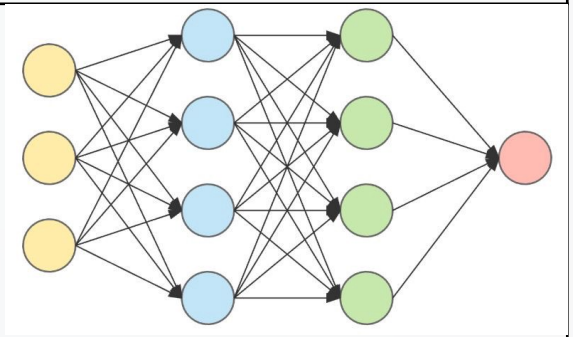
I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare	creare un **sistema logico** dal quale la macchina possa inferire la soluzione a un problema	creare un **database** di dati dai quali la macchina (tipicamente una rete neurale) possa estrarre conoscenza significativa
	▪ problemi **ben conosciuti**		▪ problemi **non sistematici**

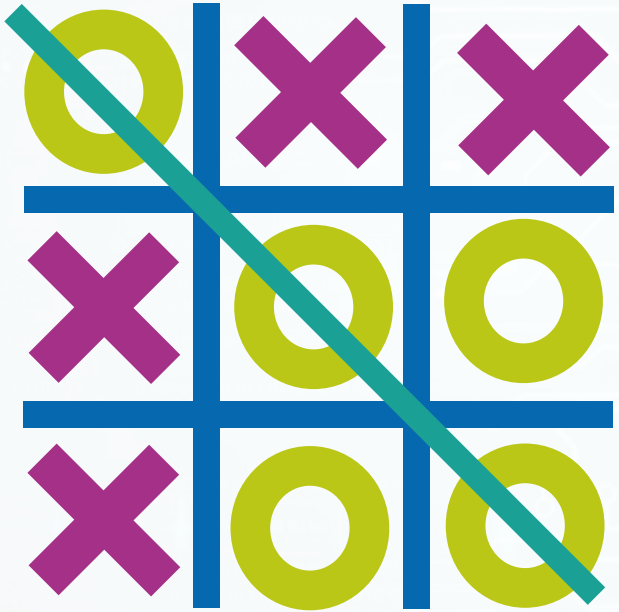
I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare	creare un **sistema logico** dal quale la macchina possa inferire la soluzione a un problema	creare un **database** di dati dai quali la macchina (tipicamente una rete neurale) possa estrarre conoscenza significativa
	▪ problemi **ben conosciuti**		

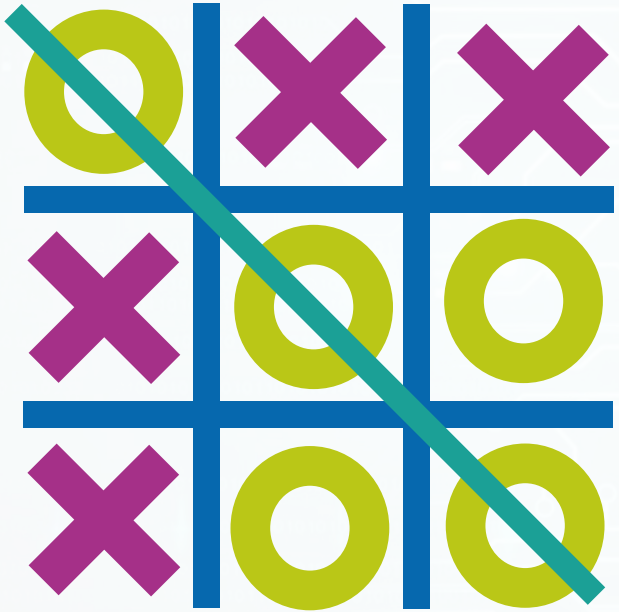
I.A. – COME PENSA UNA MACCHINA

	SISTEMI TOP-DOWN		SISTEMI BOTTOM-UP
	PROCEDURALE	LOGICO	MACHINE LEARNING
Ruolo del programmatore	scrivere un insieme di **procedure** per ogni possibile situazione che la macchina deve affrontare	creare un **sistema logico** dal quale la macchina possa inferire la soluzione a un problema	creare un **database** di dati dai quali la macchina (tipicamente una rete neurale) possa estrarre conoscenza significativa
	▪ problemi **ben conosciuti**		

I.A. – GIOCO DEL TRIS



I.A. – GIOCO DEL TRIS

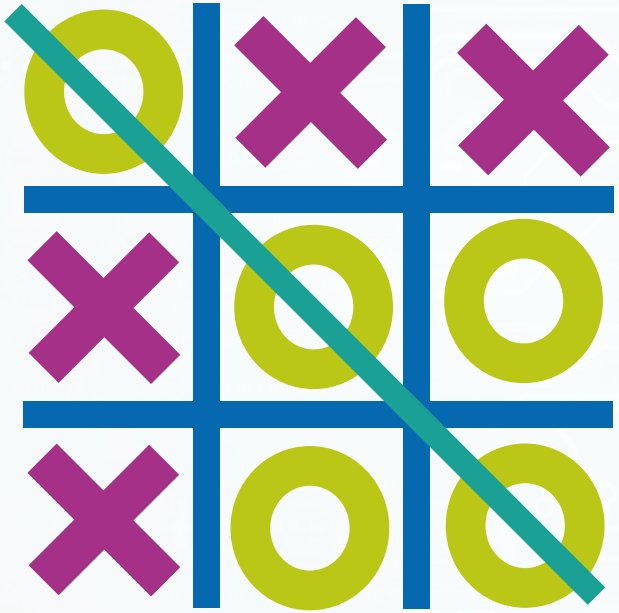


	Numero di ‘posizioni’ possibili	Numero di partite possibili
TRIS	5.478	26.830
Dama	~100.000.000.000.000.000.000 (10^{20})	Ordine di 10^{31}
Scacchi	order of 10^{47}	Ordine di 10^{123}
Go	order of 10^{170}	Ordine di 10^{360}

I.A. – GIOCO DEL TRIS

	di ‘posizioni’ ossibili	Numero di partite possibili
	5.478	26.830
	0.000.000.00 .000 (10^{20})	Ordine di 10^{31}
	er of 10^{47}	Ordine di 10^{123}
	er of 10^{170}	Ordine di 10^{360}

I.A. – GIOCO DEL TRIS



Quali sono le prime tre regole che daresti a una macchina (LUCY) per insegnarle vincere a TRIS?

TRIS

APPROCCIO PROCEDURALE (IMPERATIVO)

TRIS – APPROCCIO IMPERATIVO

Newell and Simon’s (1972) algorithm

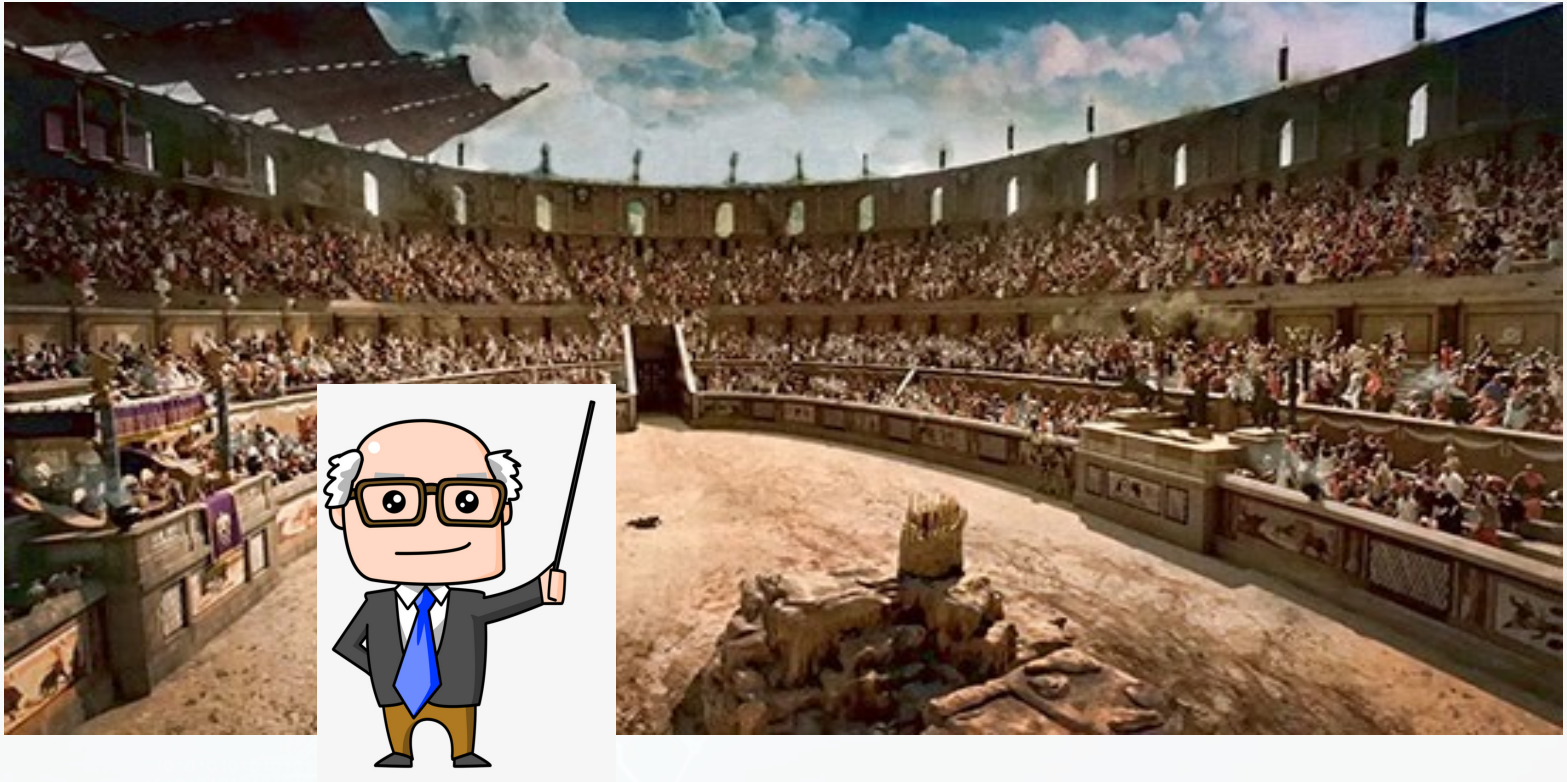
1. **Win**: If you have two in a row, play the third to get three in a row.
2. **Block**: If the opponent has two in a row, play the third to block them.
3. **Fork**: Create an opportunity where you can win in two ways.
4. **Block Opponent's Fork**:

   Option 1: Create two in a row to force the opponent into defending, as long as it doesn't result in them creating a fork or winning. For example, if "X" has a corner, "O" has the center, and "X" has the opposite corner as well, "O" must not play a corner in order to win. (Playing a corner in this scenario creates a fork for "X" to win.)

   Option 2: If there is a configuration where the opponent can fork, block that fork.
5. **Center**: Play the center.
6. **Opposite Corner**: If the opponent is in the corner, play the opposite corner.
7. **Empty Corner**: Play an empty corner.
8. **Empty Side**: Play an empty side.

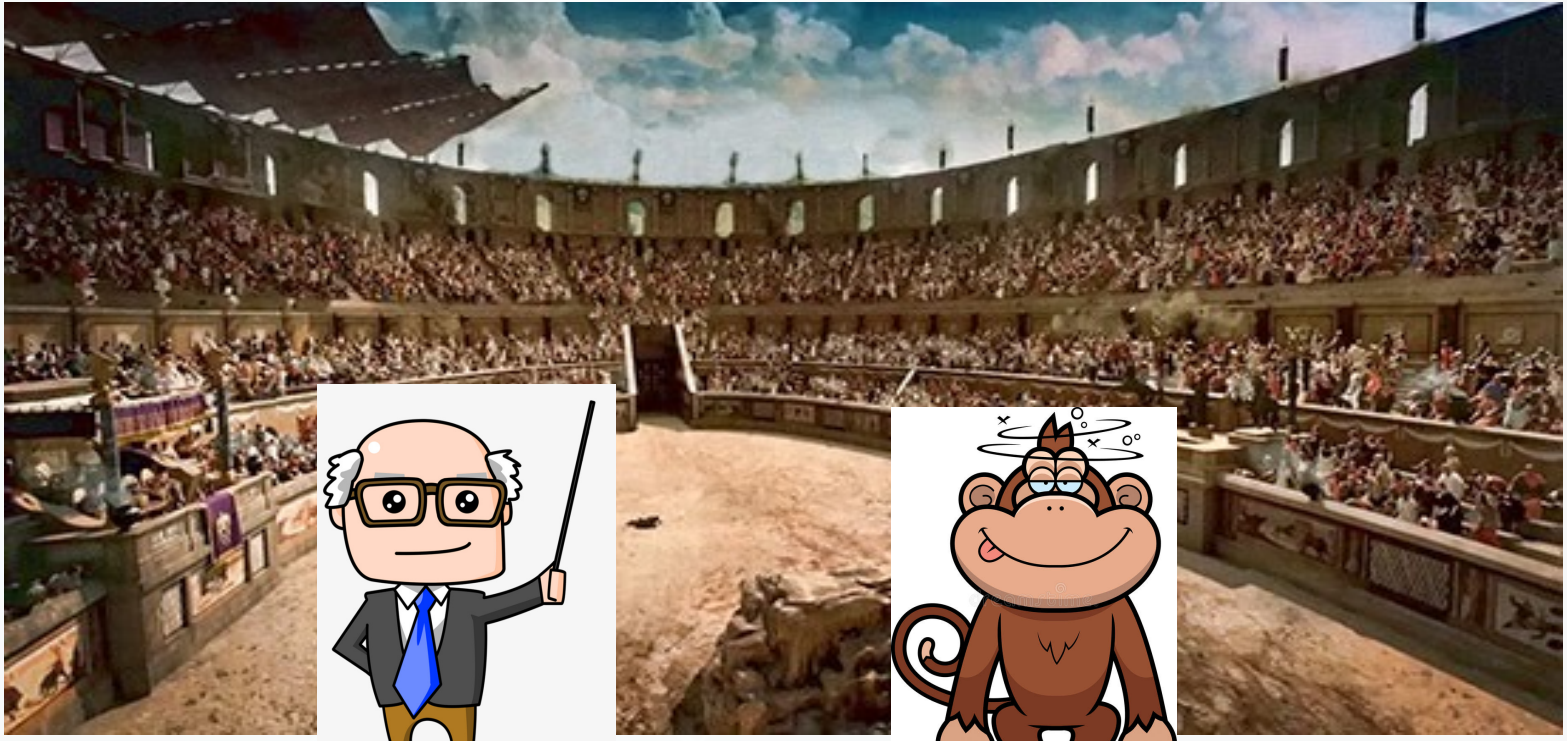
TRIS – PRIMI DUE GIOCATORI

TRIS – PRIMI DUE GIOCATORI



Algoritmo IMPERATIVO

TRIS – PRIMI DUE GIOCATORI

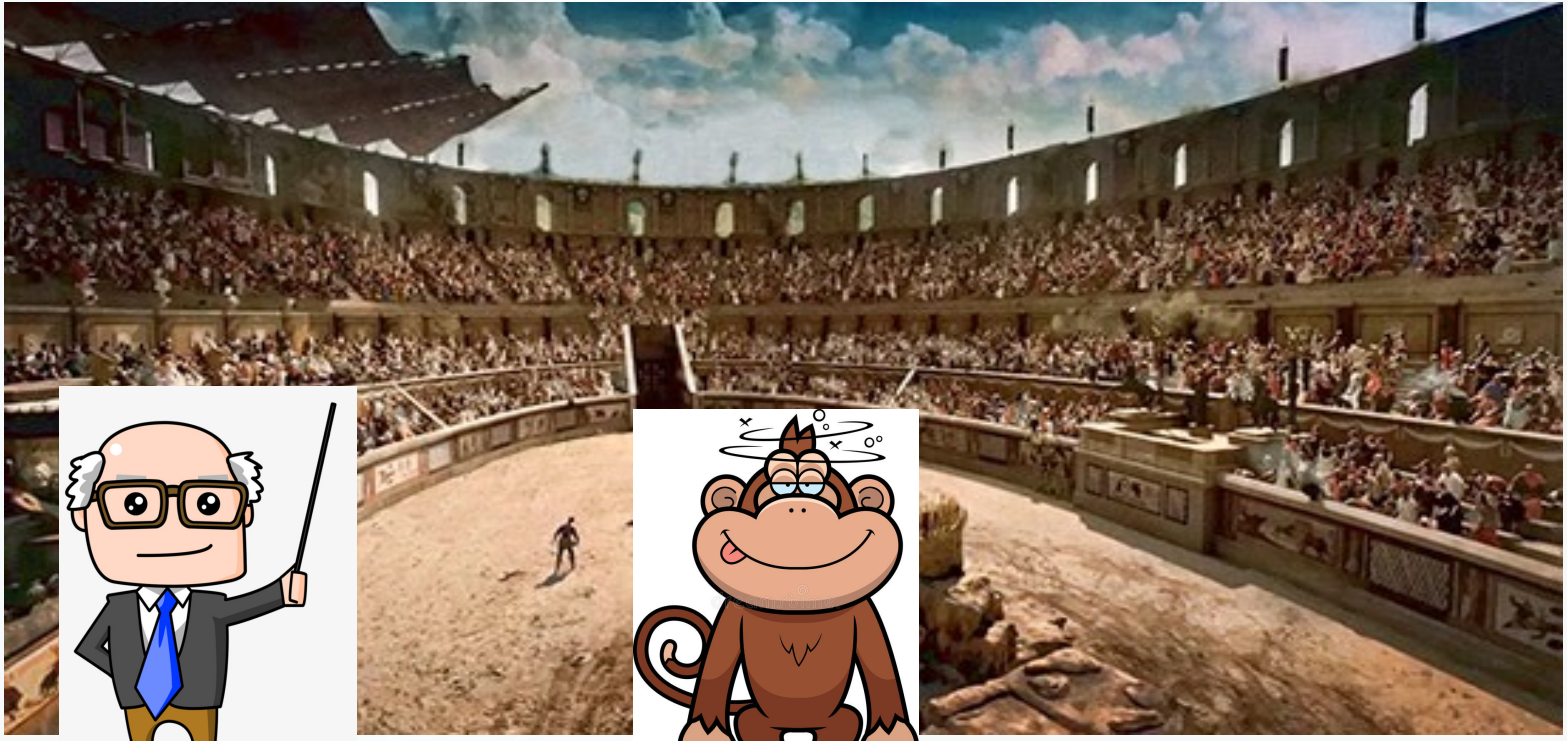


Algoritmo IMPERATIVO

Algoritmo RANDOM


MATLAB

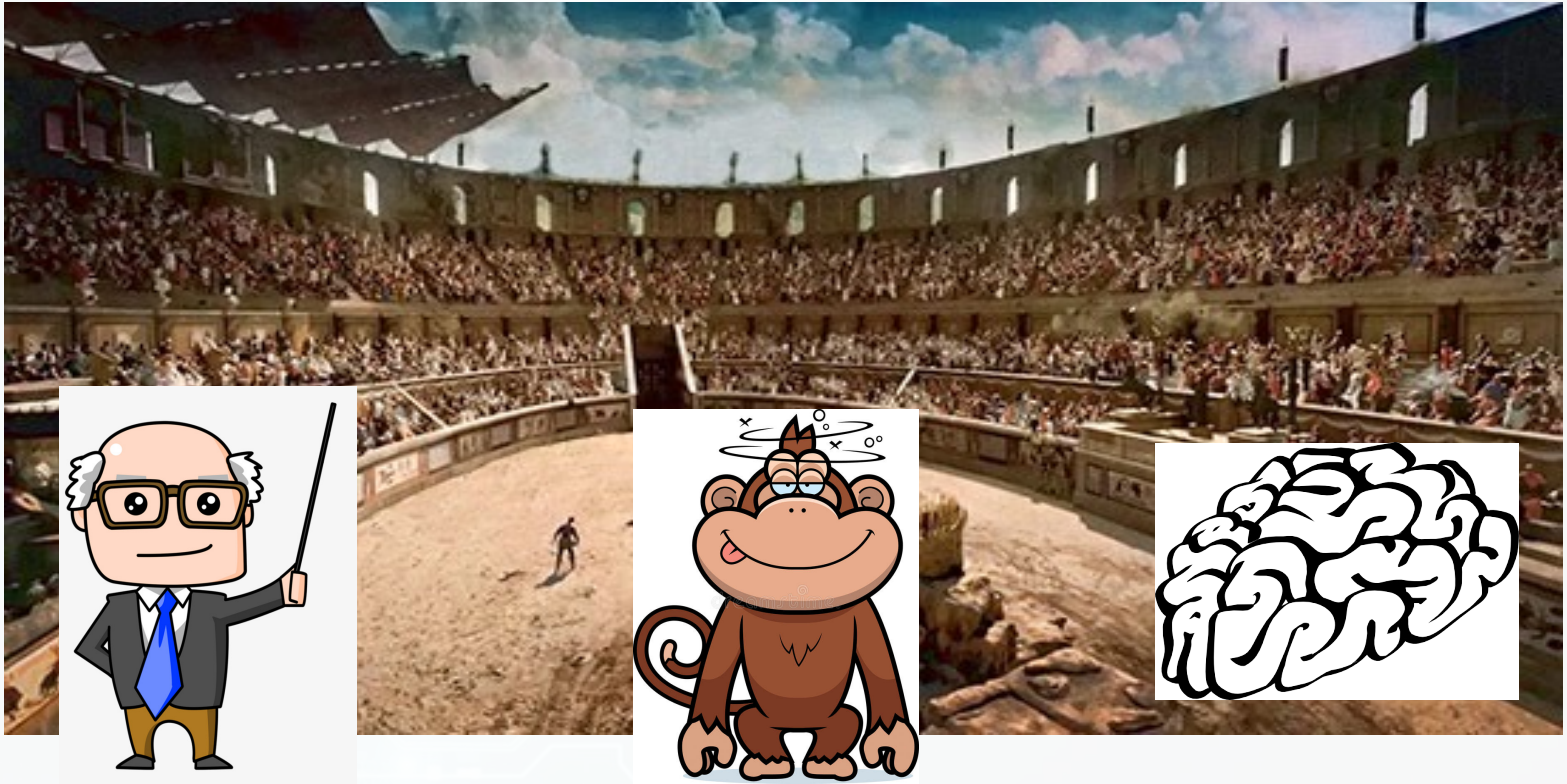
- **TRIS –** TERZO GIOCATORE



Algoritmo IMPERATIVO

Algoritmo RANDOM

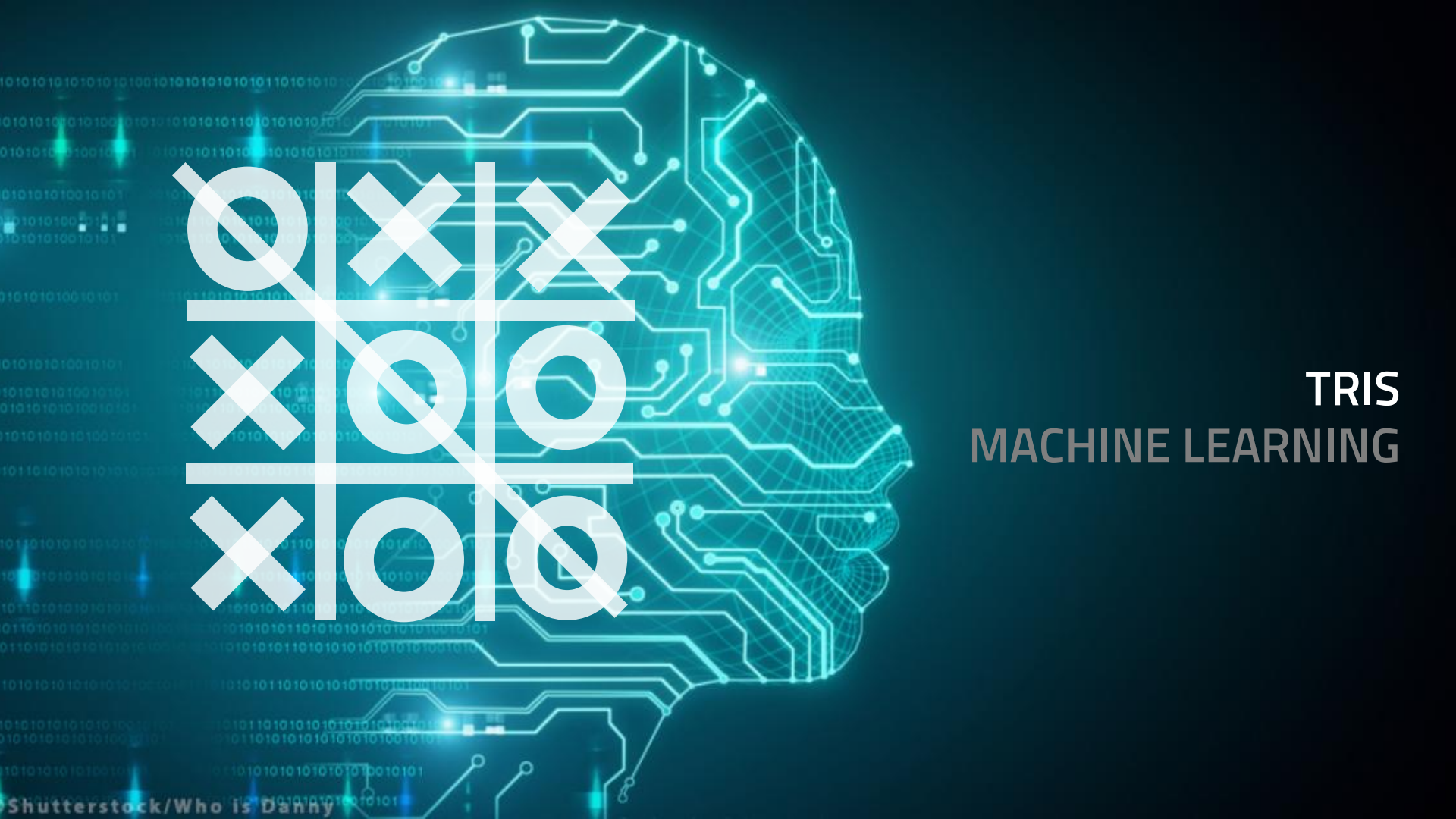
- **TRIS –** TERZO GIOCATORE



Algoritmo IMPERATIVO

Algoritmo RANDOM

Algoritmo BOTTOM-UP


TRIS
MACHINE LEARNING
Shutterstock/Who is Danny

TRIS – APPROCCIO MACHINE LEARNING

Quali sono gli step?

TRIS – APPROCCIO MACHINE LEARNING

Quali sono gli step?

1. costruzione di un **database** di partite da cui apprendere

TRIS – APPROCCIO MACHINE LEARNING

Quali sono gli step?

1. costruzione di un **database** di partite da cui apprendere
2. **training** di un modello di Machine Learning

TRIS – APPROCCIO MACHINE LEARNING

Quali sono gli step?

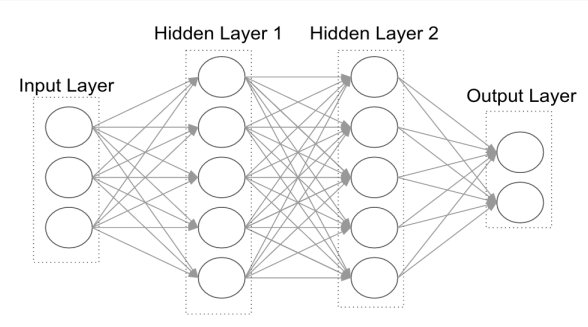
1. costruzione di un **database** di partite da cui apprendere
2. **training** di un modello di Machine Learning
3. **test** delle capacità di Lucy nel gioco del tris

TRIS – APPROCCIO MACHINE LEARNING

Quali sono gli step?

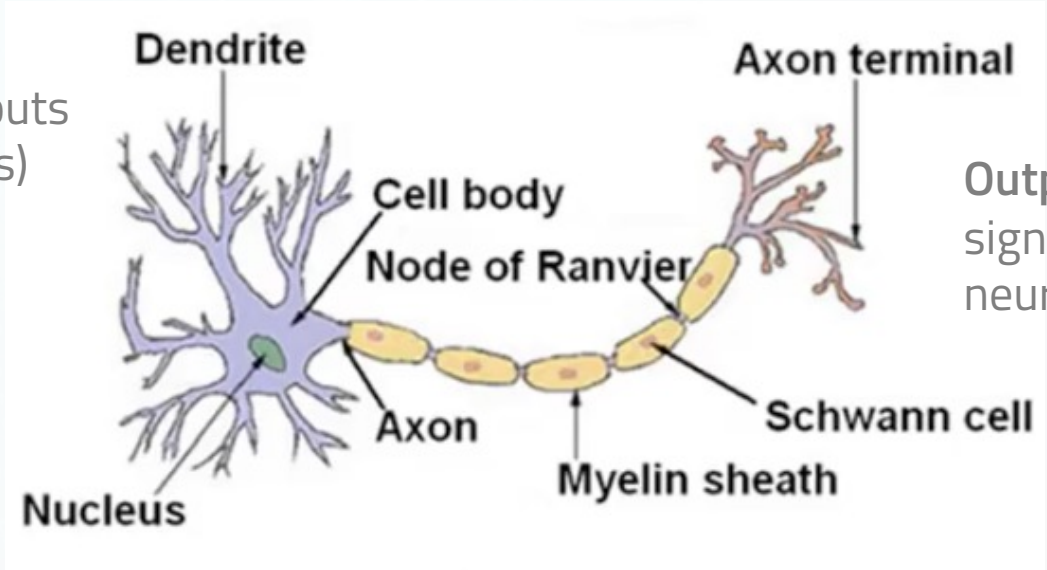
1. costruzione di un **database** di partite da cui apprendere
2. **training** di un modello di Machine Learning
3. **test** delle capacità di Lucy nel gioco del tris

Reti Neurali



Reti Neurali

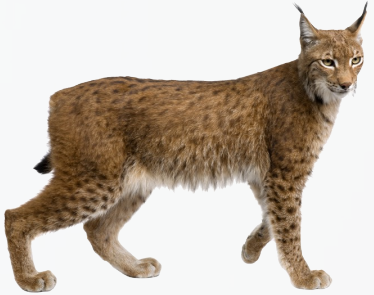
Input wires (receive the inputs from the others)



Output wire (sends the signals processed by the neuron to other neurons)

RETI NEURALI – GATTO vs LINCE

GATTO



LINCE

RETI NEURALI – GATTO vs LINCE

Coda
(cm)

Peso
(kg)

n°
gambe

RETI NEURALI – GATTO vs LINCE

Coda (cm)	27.3
Peso (kg)	3.2
n° gambe	4

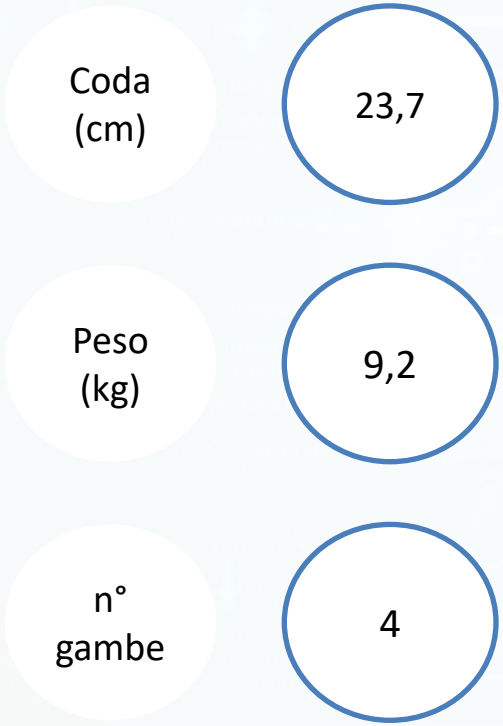
RETI NEURALI – GATTO vs LINCE

Coda (cm)	22.5
Peso (kg)	4.8
n° gambe	4

RETI NEURALI – GATTO vs LINCE

Coda (cm)	17,7
Peso (kg)	8,1
n° gambe	4

RETI NEURALI – GATTO vs LINCE

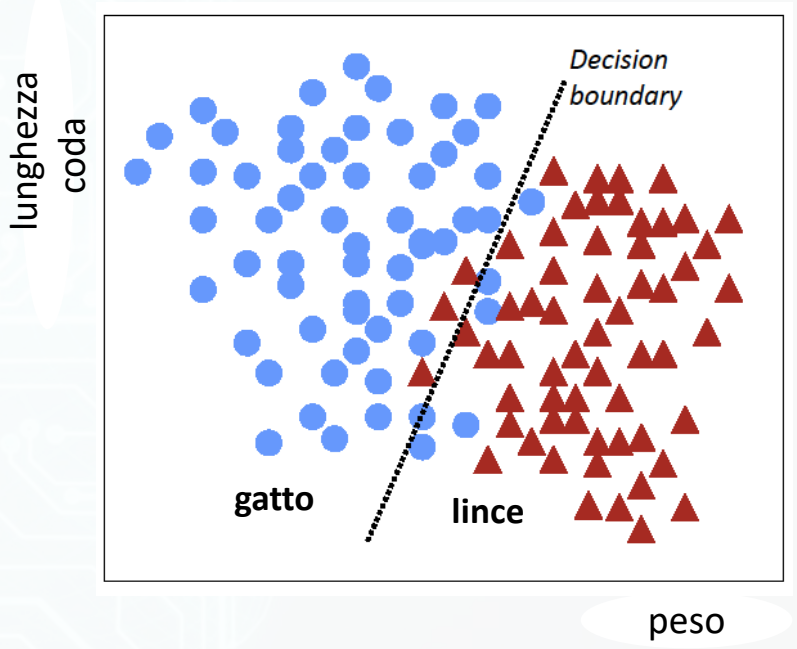


Is it a **cat** or a **lynx**?

RETI NEURALI – GATTO vs LINCE

Coda (cm) — 23,7

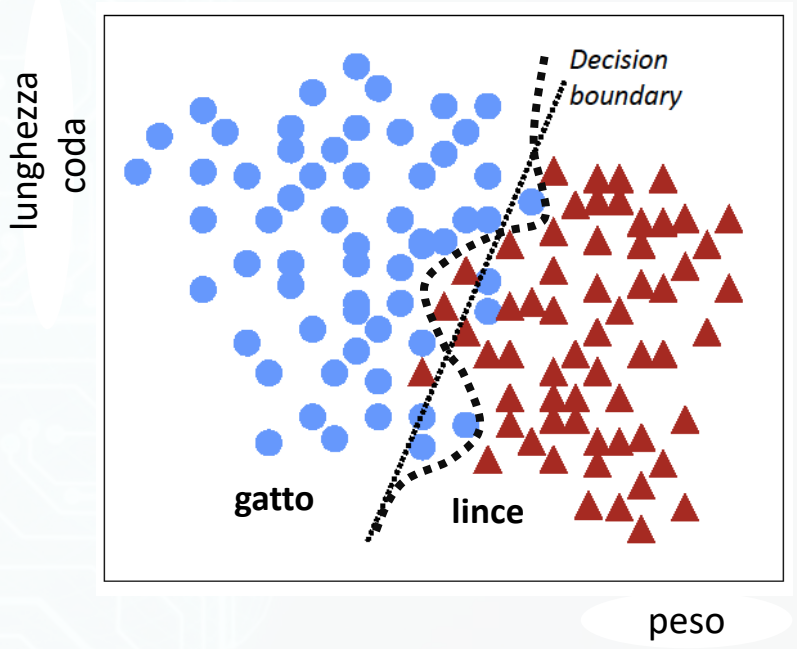
Peso (kg) — 9,2



RETI NEURALI – GATTO vs LINCE

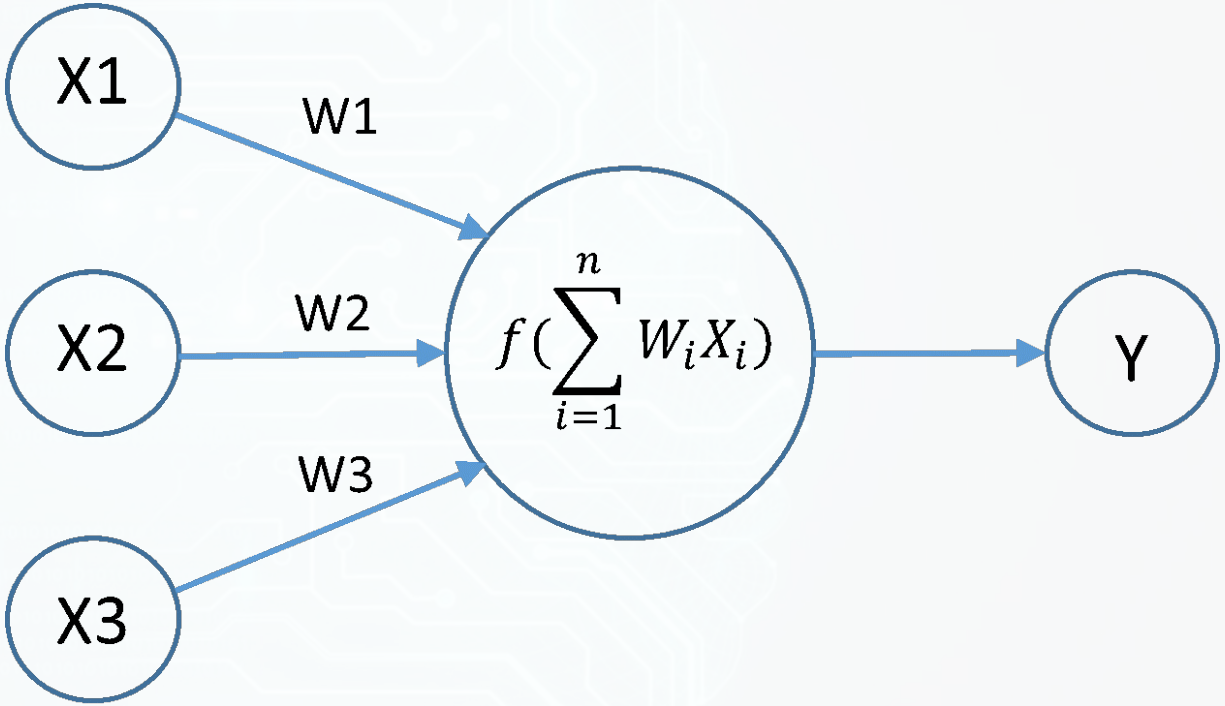
Coda (cm) — 23,7

Peso (kg) — 9,2

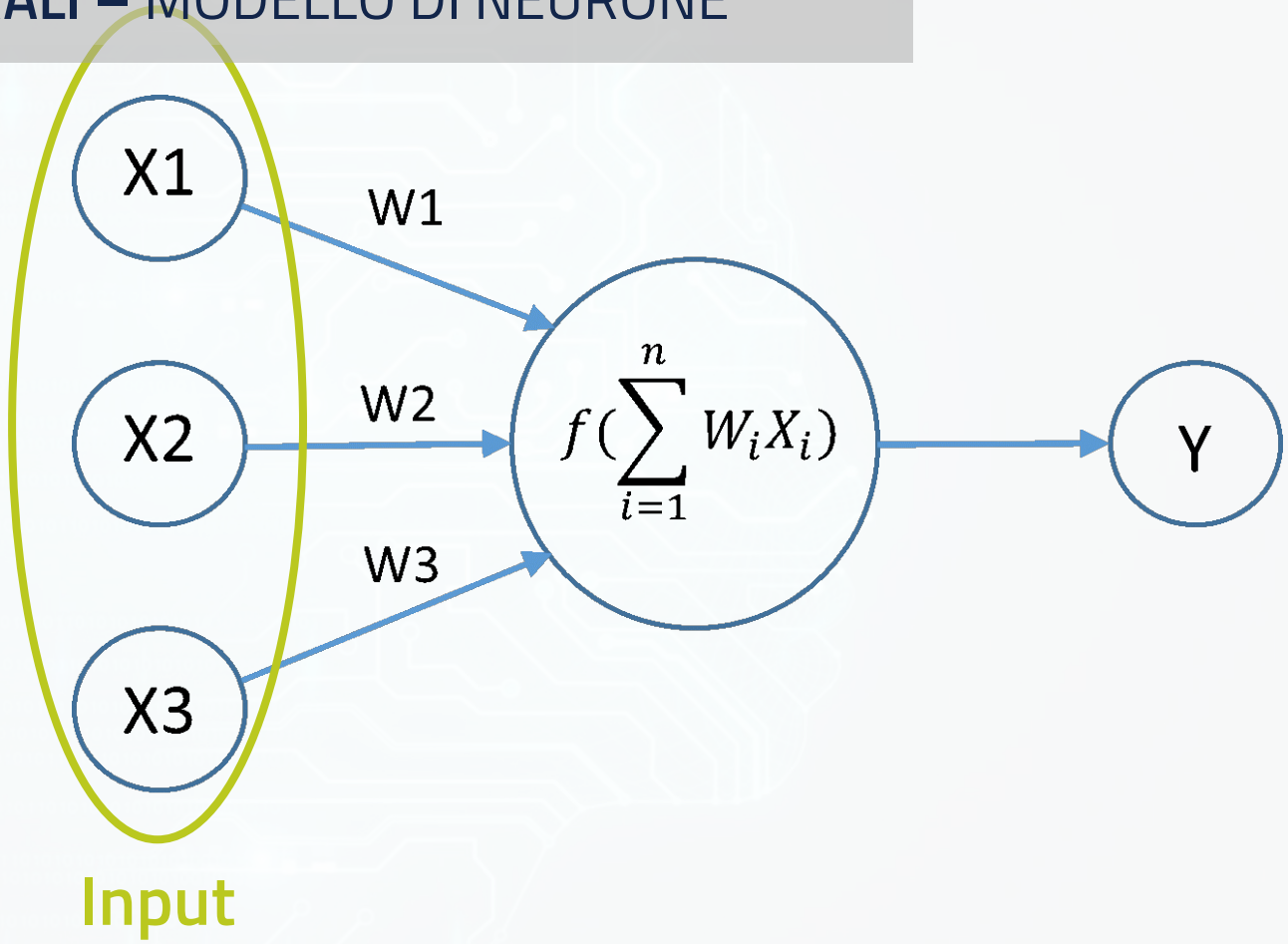


Come si allena un rete neurale a distinguere tra gatto e lince?

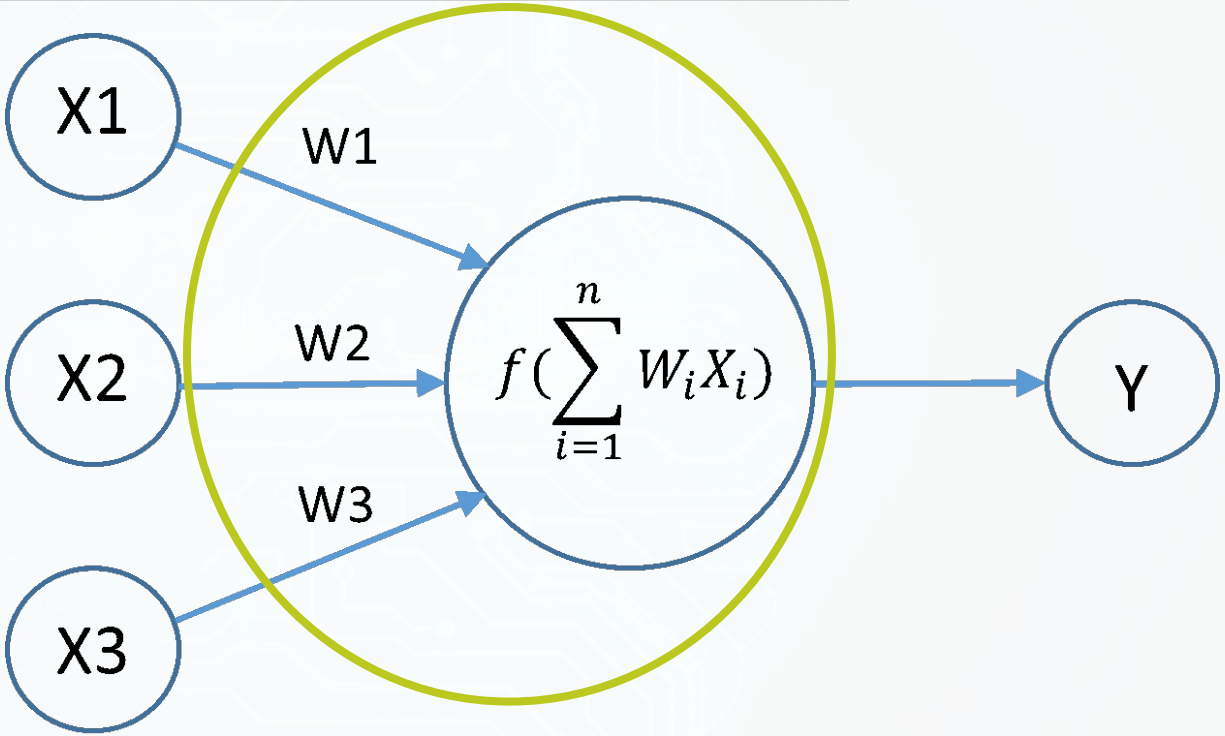
RETI NEURALI – MODELLO DI NEURONE



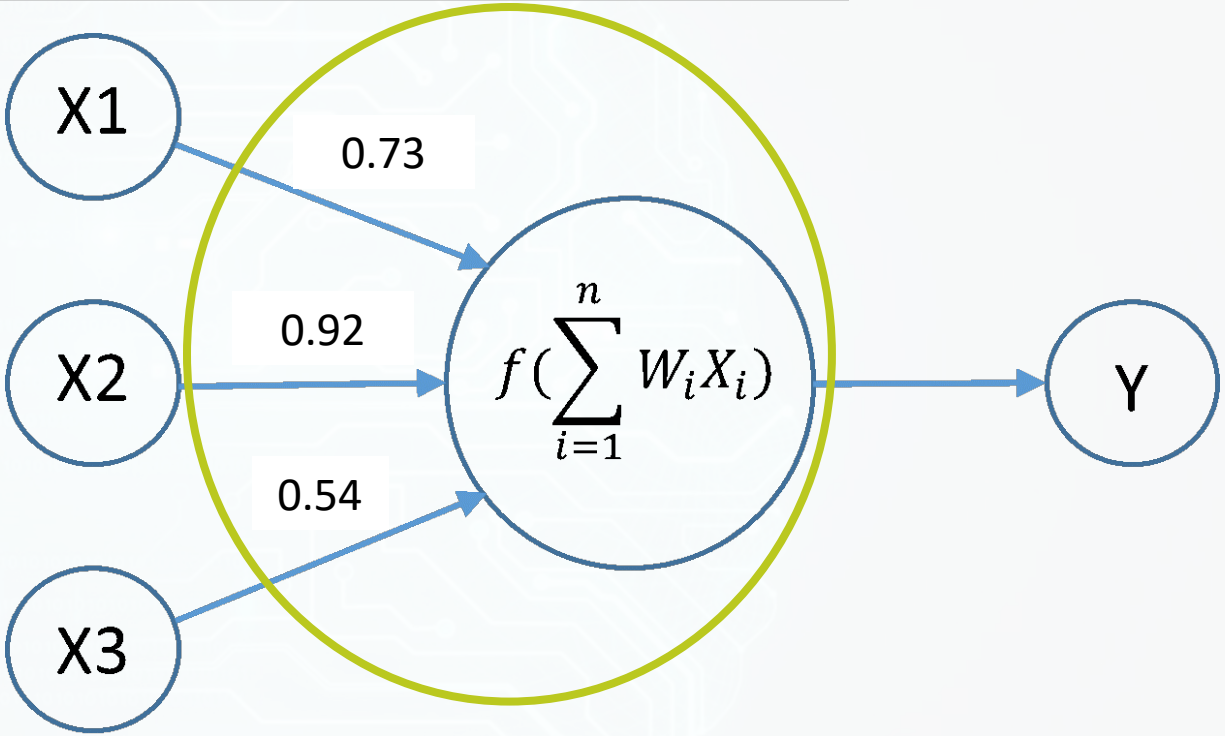
RETI NEURALI – MODELLO DI NEURONE



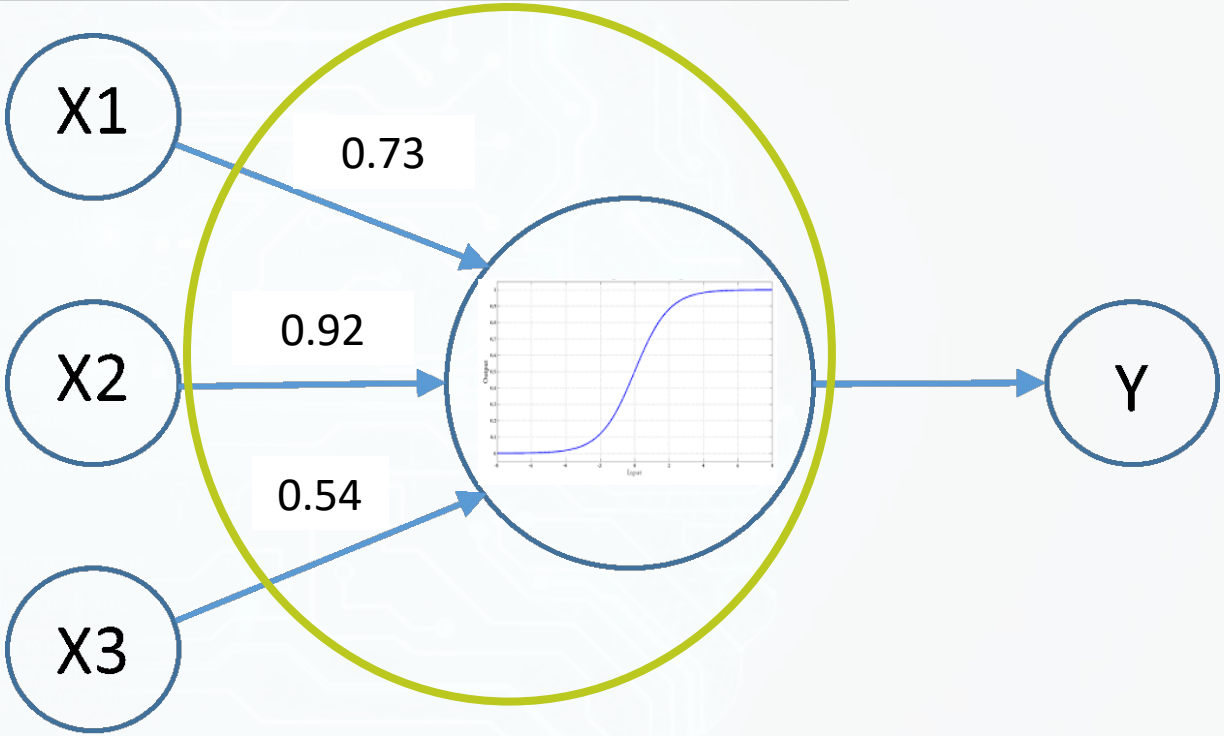
RETI NEURALI – MODELLO DI NEURONE



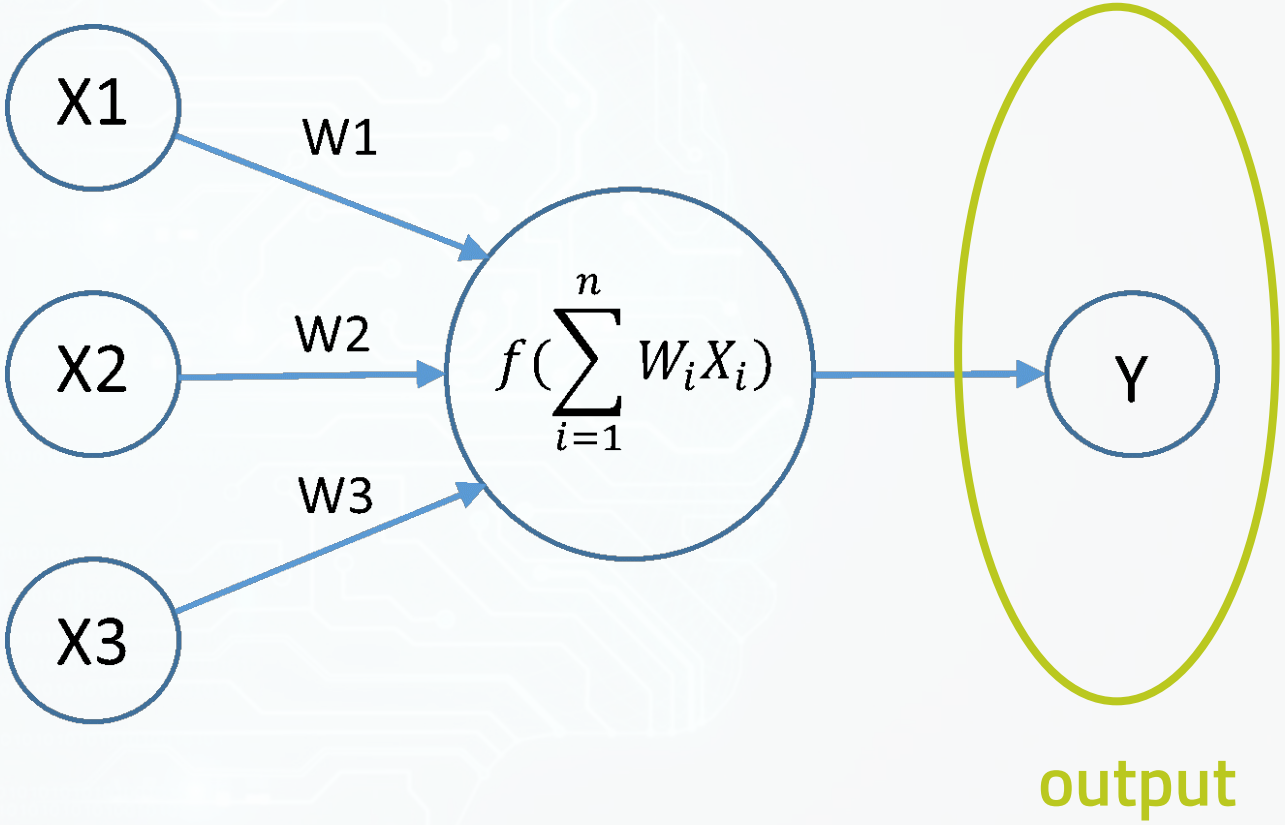
RETI NEURALI – MODELLO DI NEURONE



RETI NEURALI – MODELLO DI NEURONE



RETI NEURALI – MODELLO DI NEURONE



RETI NEURALI – PROCESSO DI TRAINING

27.3

3.2

4

0.73

0.92

0.54

$$f(\sum_{i=1}^{n} W_i X_i)$$

???

Example

RETI NEURALI – PROCESSO DI TRAINING

27.3

3.2

4

0.73

0.92

0.54

$$f(\sum_{i=1}^{n} W_i X_i)$$

???

Example

Etichetta target: gatto

RETI NEURALI – PROCESSO DI TRAINING

27.3

3.2

4

0.73

0.92

0.54

$$f\left(\sum_{i=1}^{n} w_i x_i\right)$$

???

Example

Etichetta target: gatto

RETI NEURALI – PROCESSO DI TRAINING

27.3

3.2

4

0.73

0.92

0.54

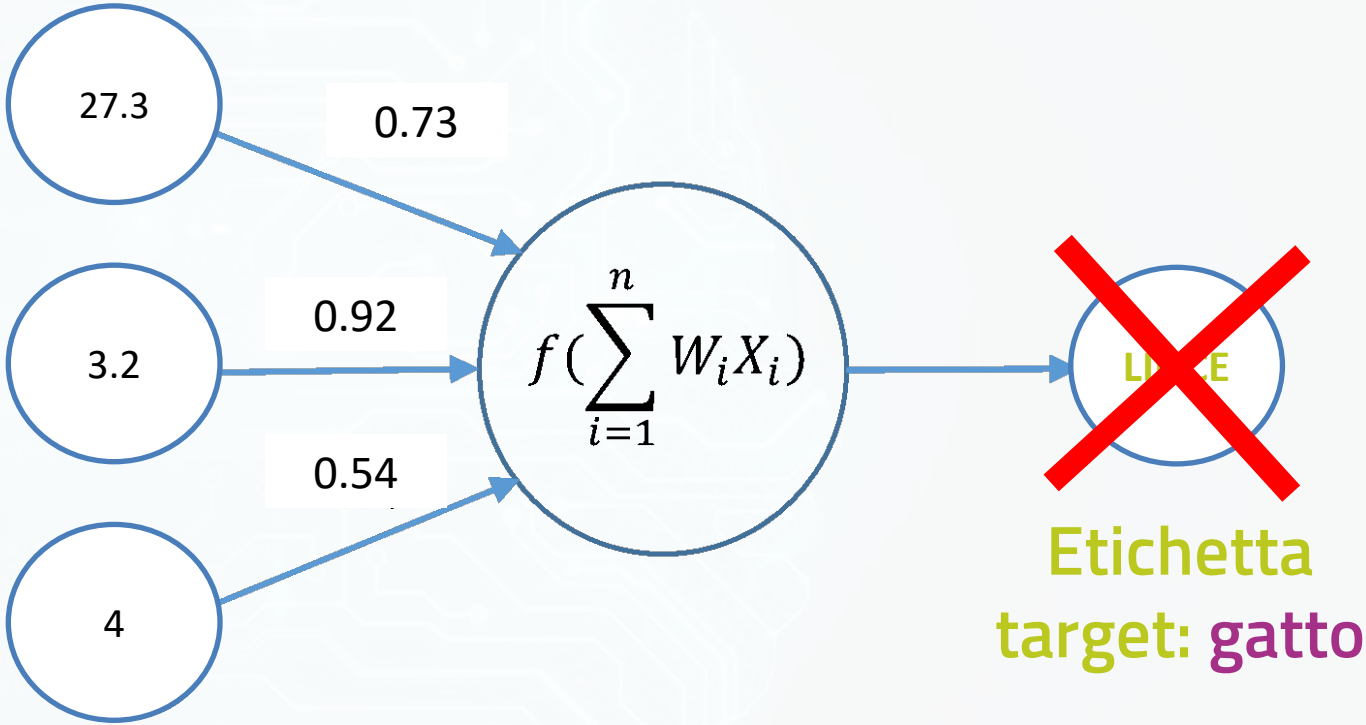
$$f\left(\sum_{i=1}^{n} w_i x_i\right)$$

LINCE

Etichetta target: gatto

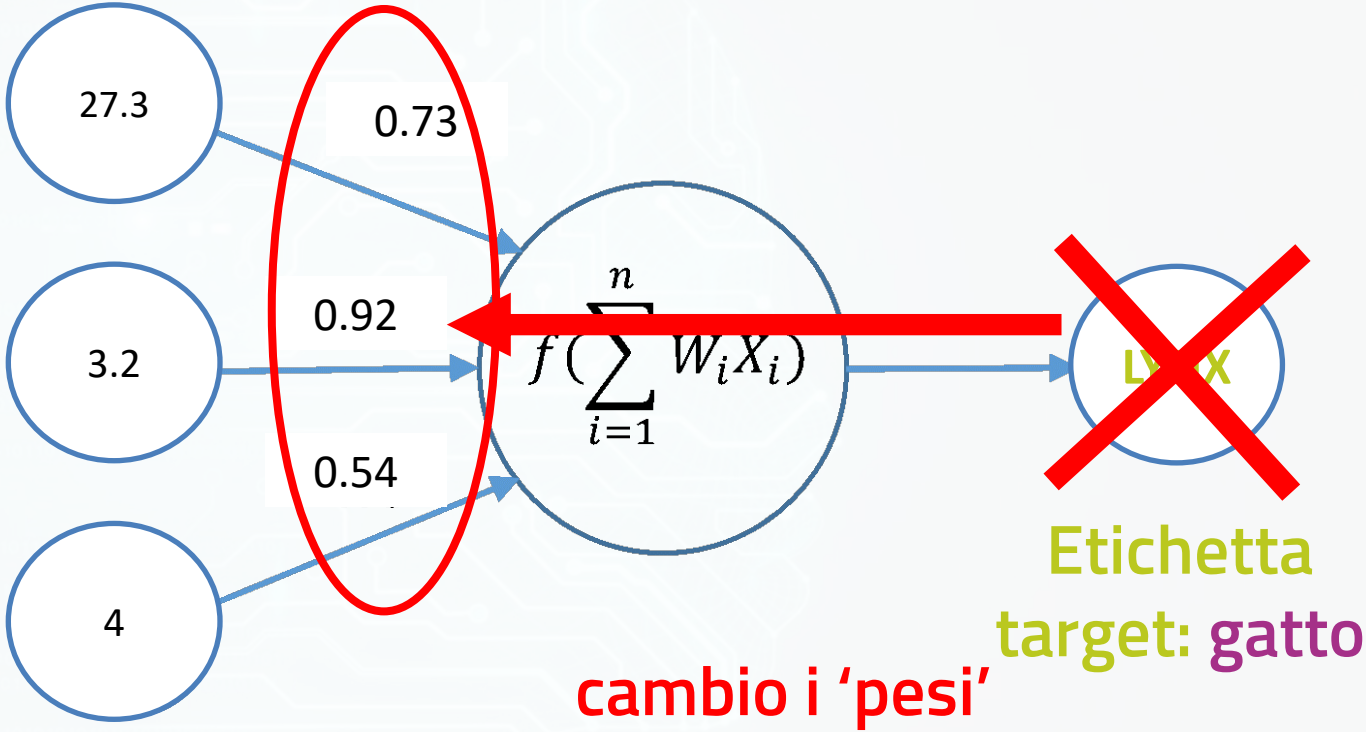
Example

RETI NEURALI – PROCESSO DI TRAINING

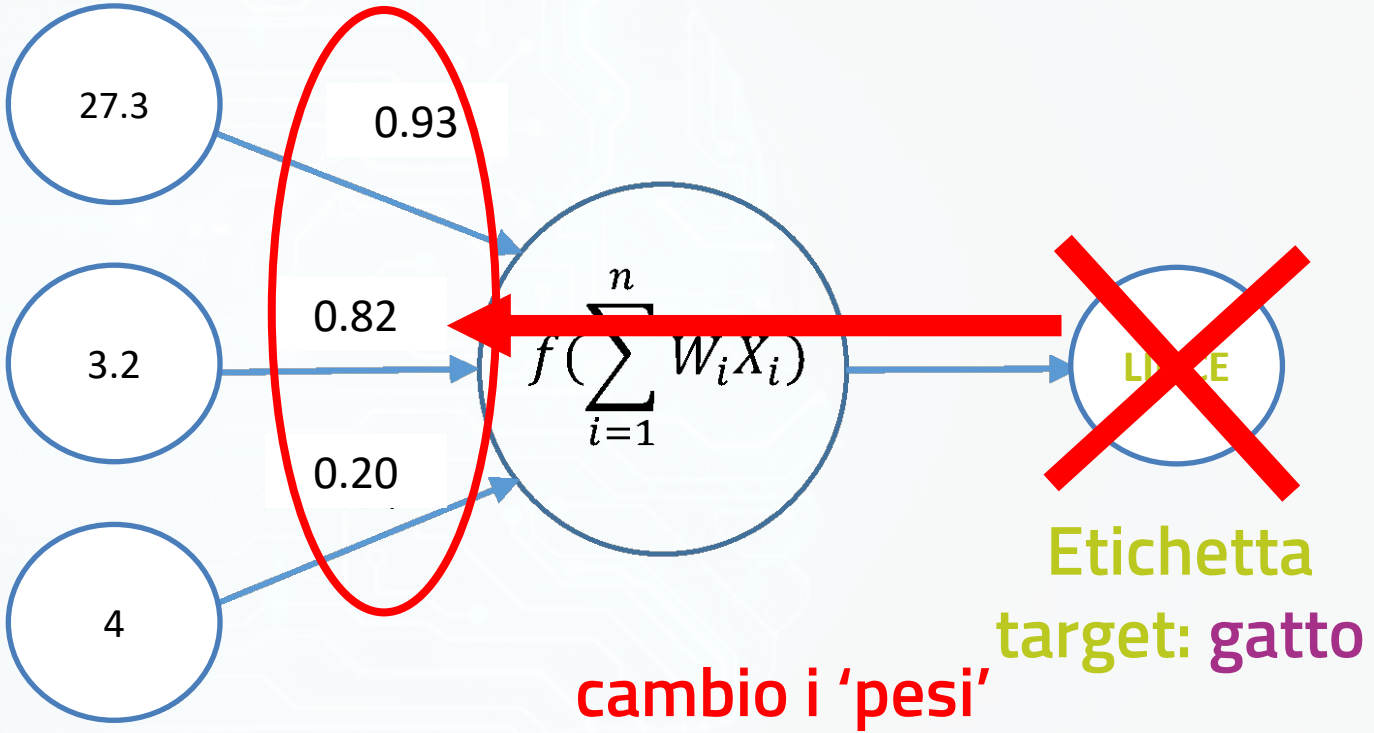


Etichetta
target: gatto

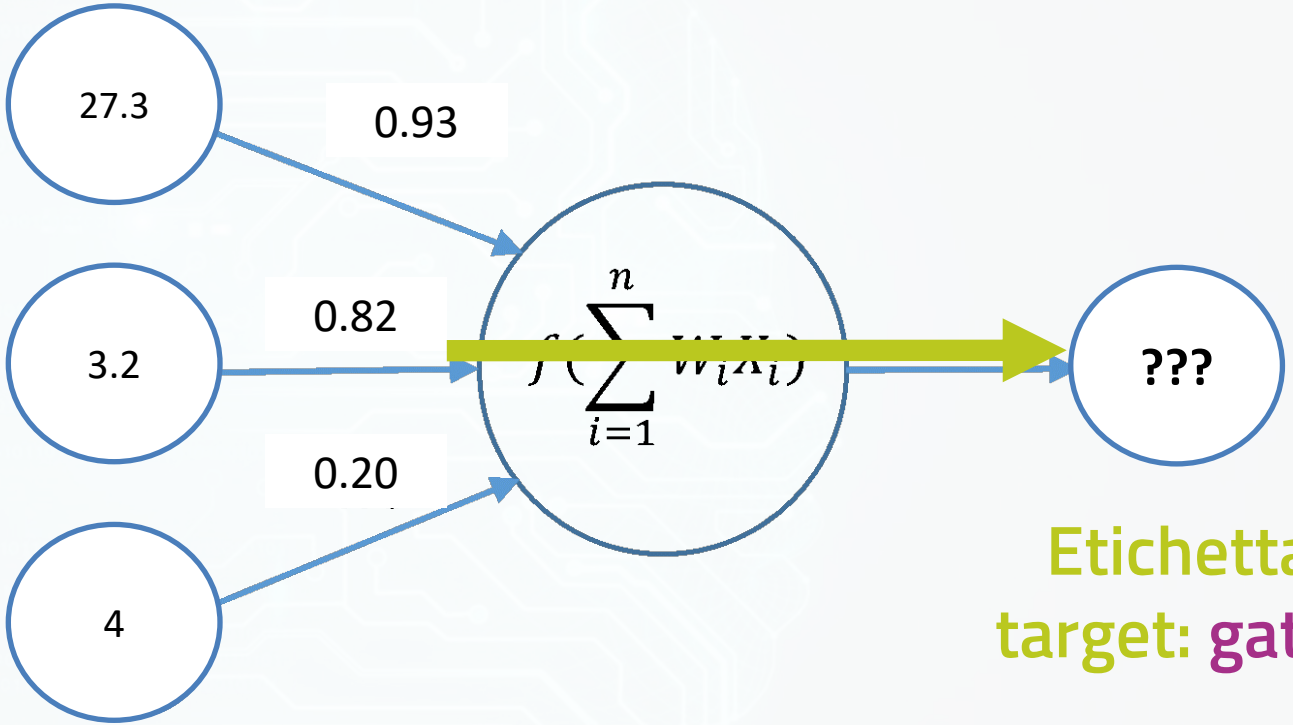
RETI NEURALI – PROCESSO DI TRAINING



RETI NEURALI – PROCESSO DI TRAINING

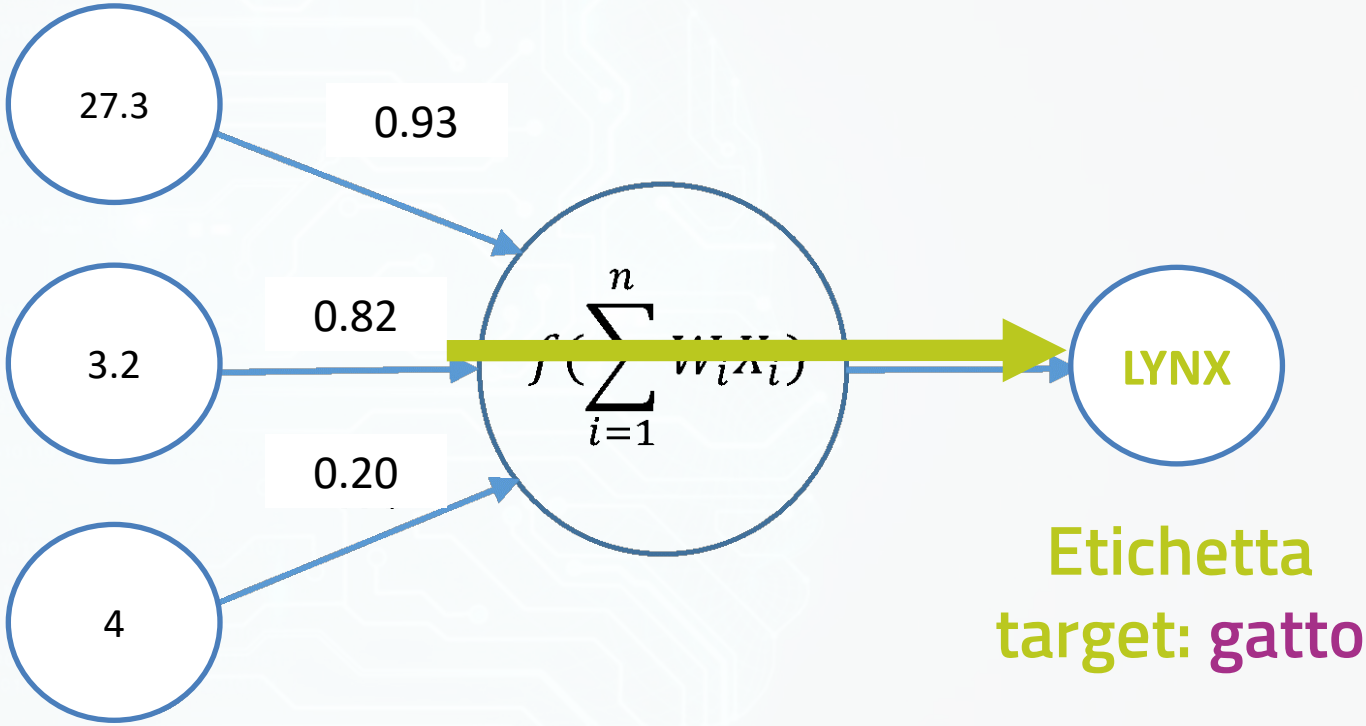


RETI NEURALI – PROCESSO DI TRAINING



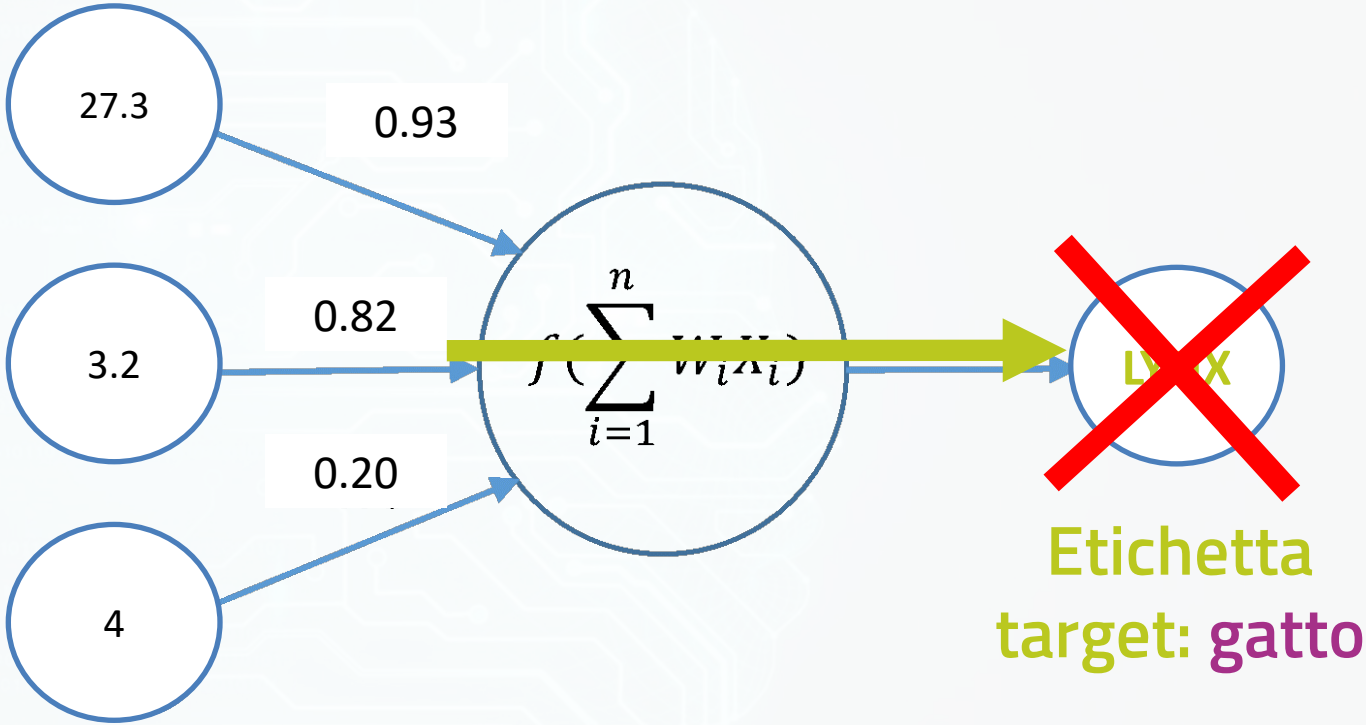
Etichetta
target: gatto

RETI NEURALI – PROCESSO DI TRAINING

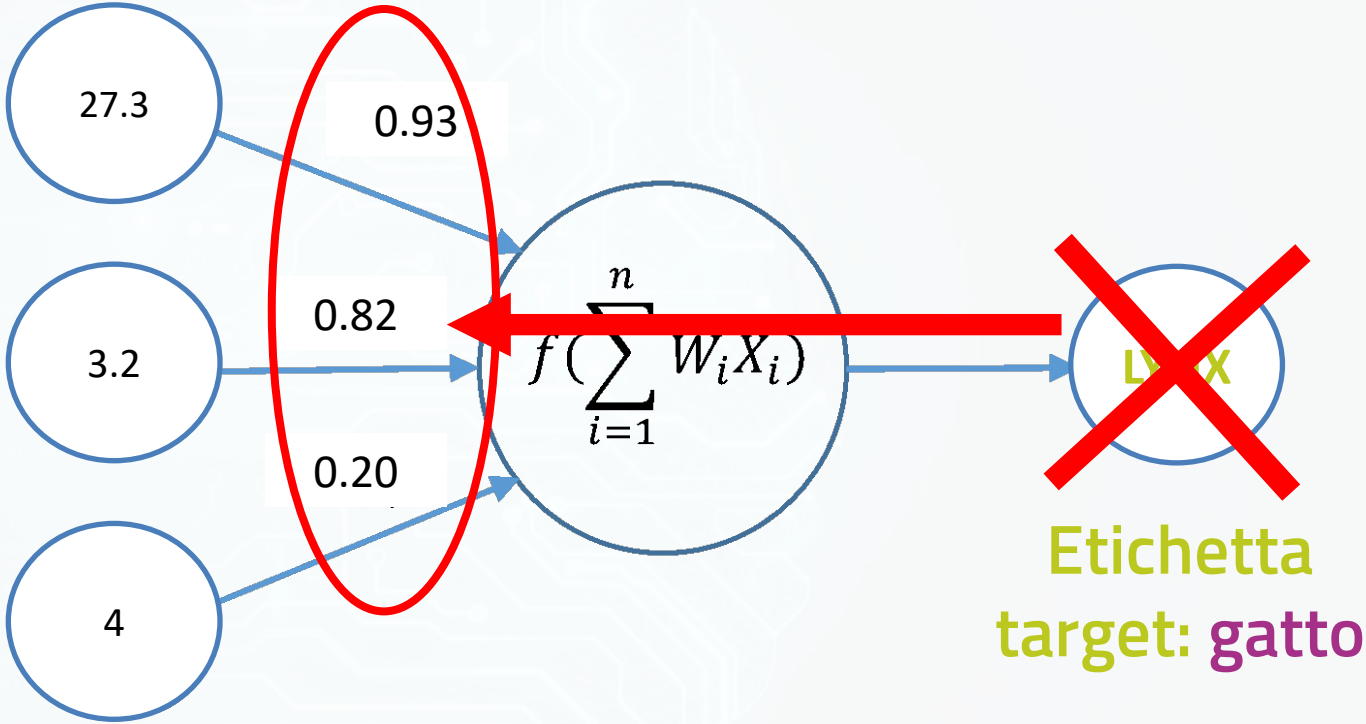


Etichetta target: gatto

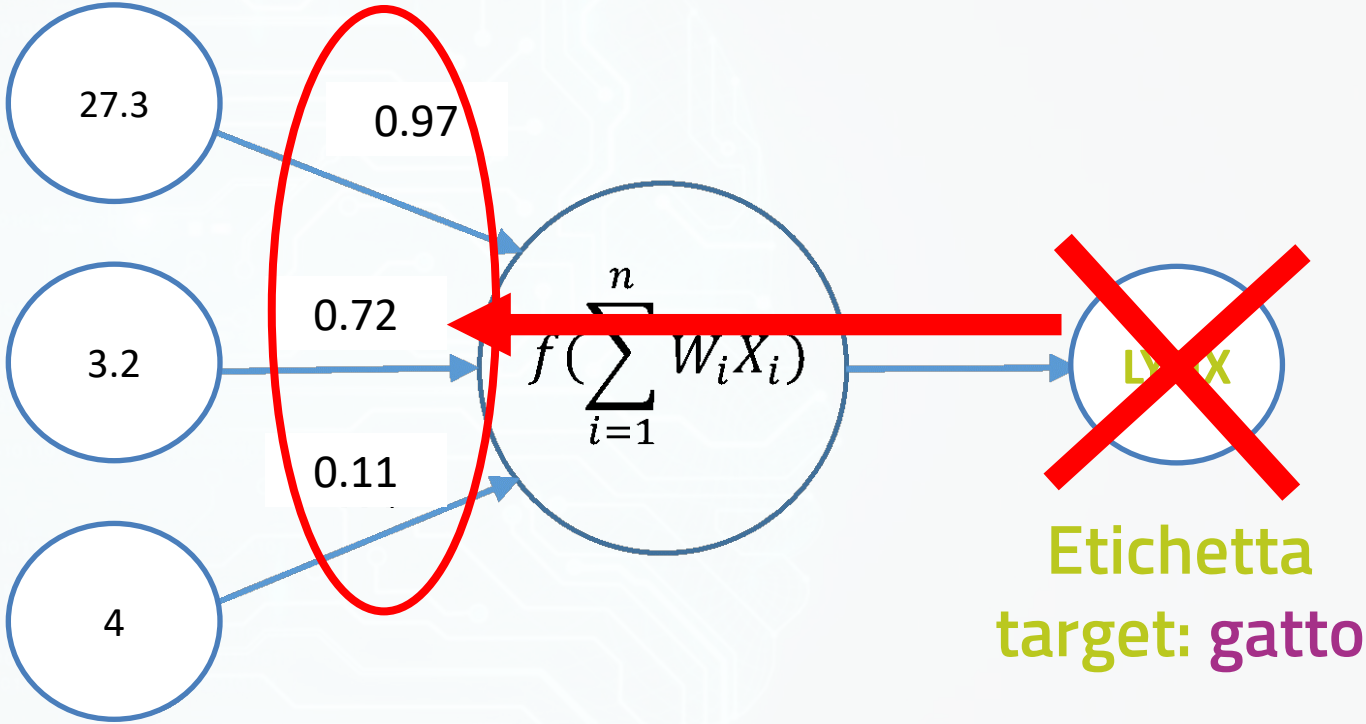
RETI NEURALI – PROCESSO DI TRAINING



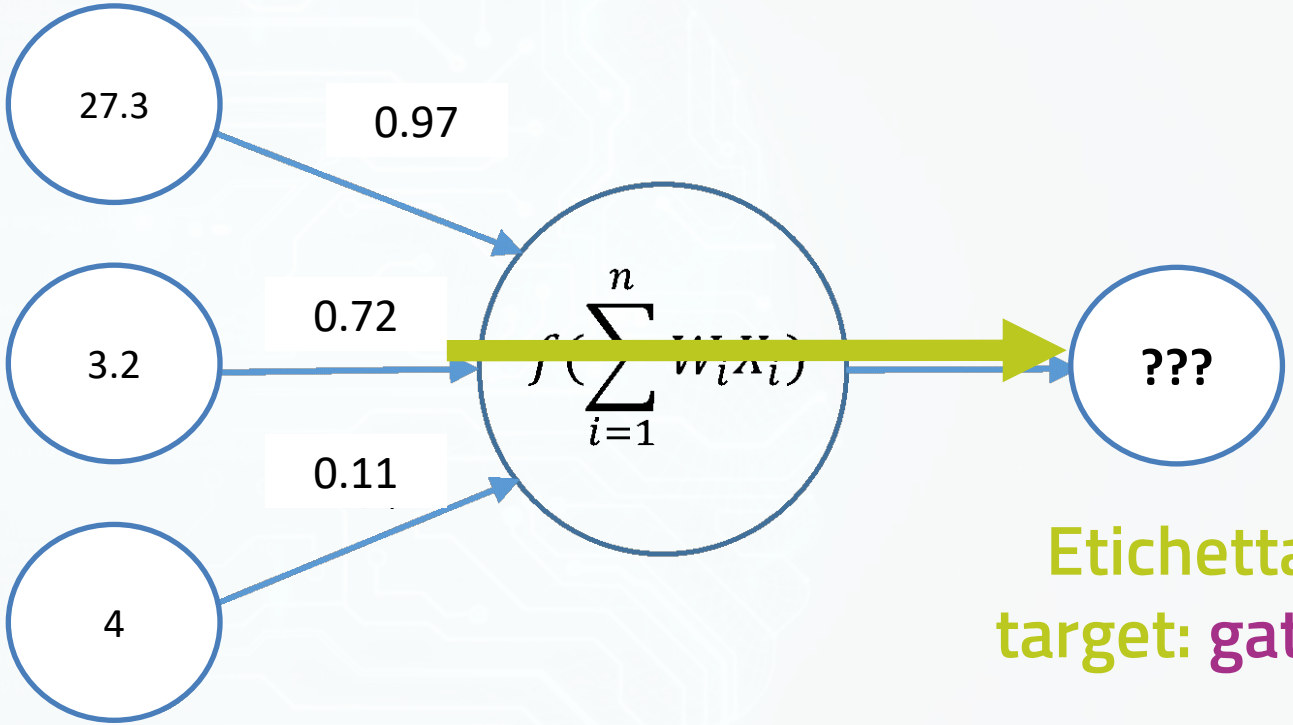
RETI NEURALI – PROCESSO DI TRAINING



RETI NEURALI – PROCESSO DI TRAINING

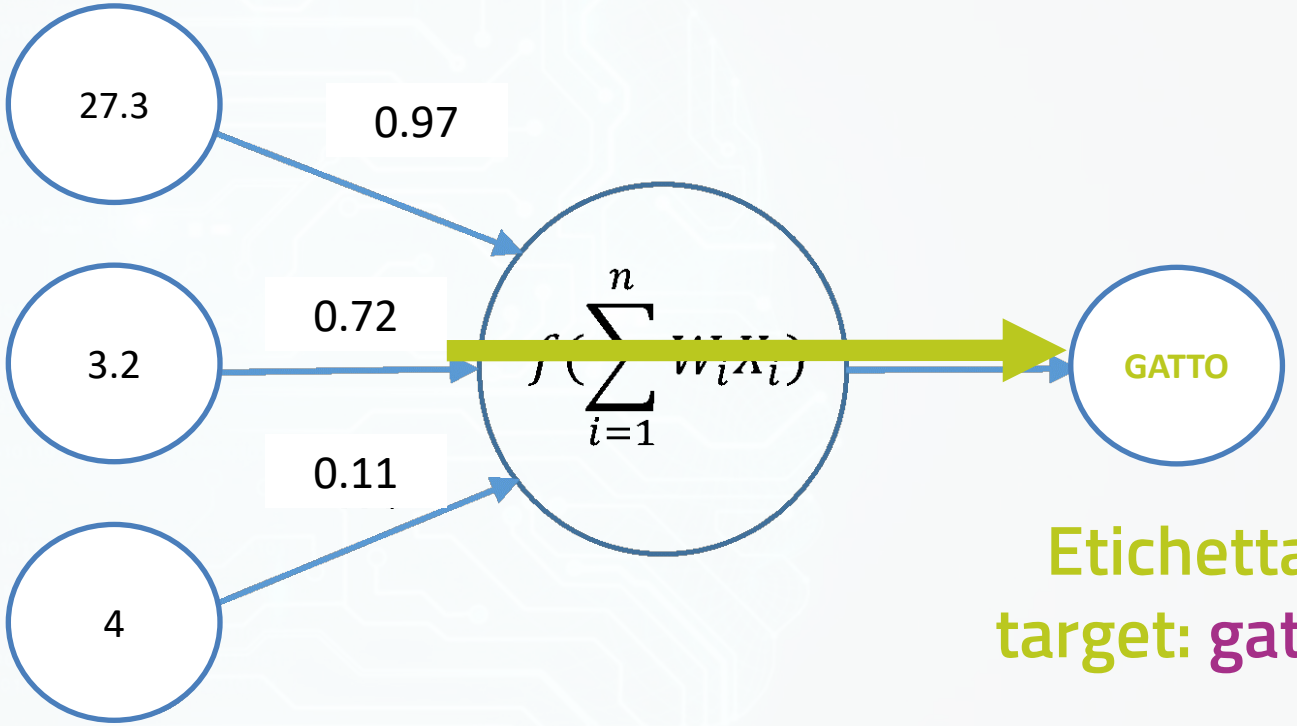


RETI NEURALI – PROCESSO DI TRAINING



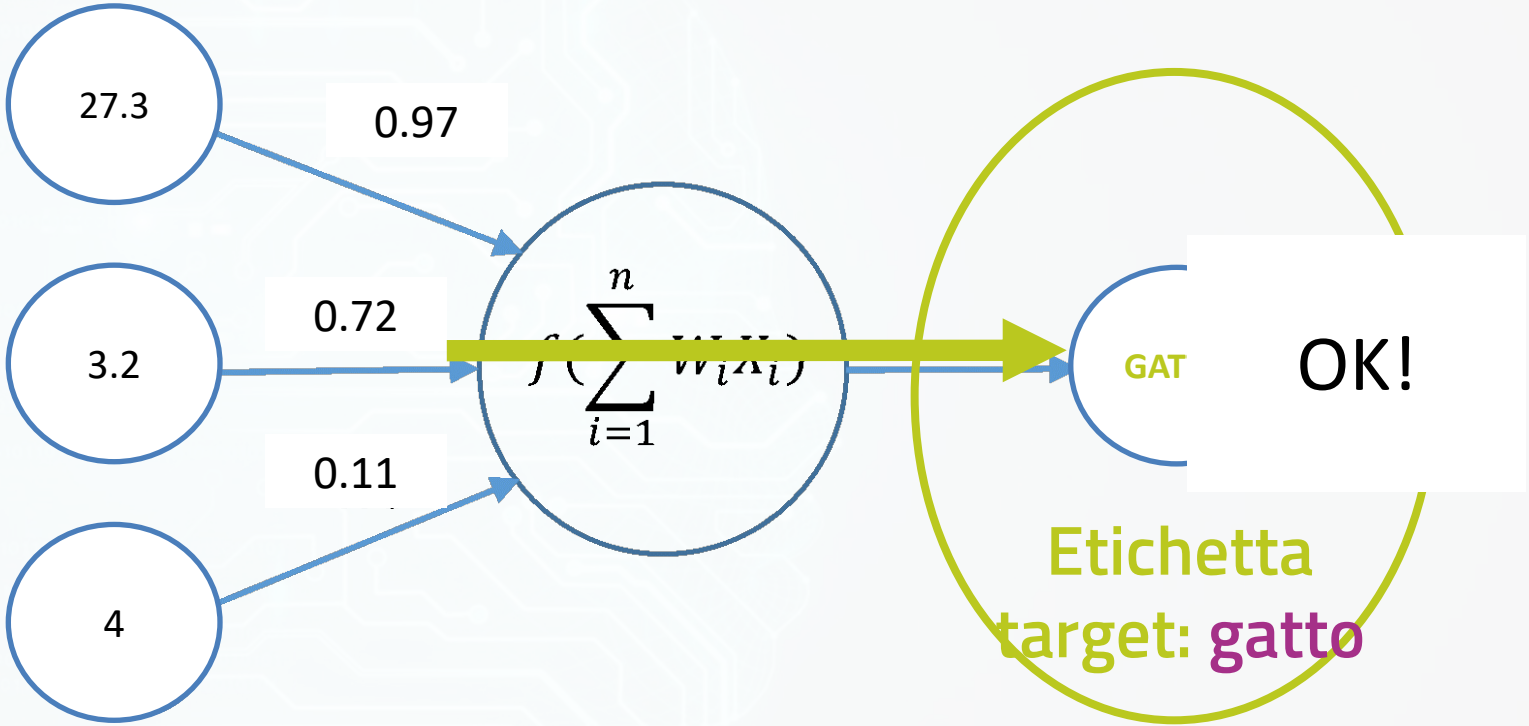
Etichetta target: gatto

RETI NEURALI – PROCESSO DI TRAINING

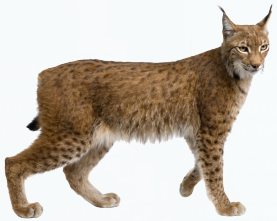


Etichetta target: gatto

RETI NEURALI – PROCESSO DI TRAINING



RETI NEURALI – PROCESSO DI TRAINING



17.7

8.1

4

0.97

0.72

0.11

$$f(\sum_{i=1}^{n} W_i X_i)$$

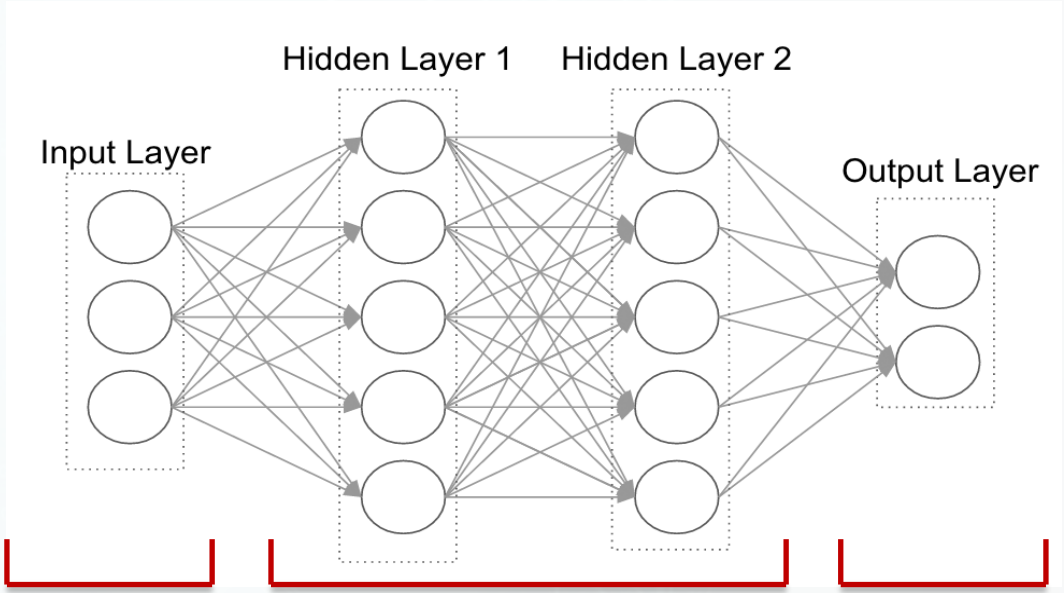
???

Input

Etichetta target: lince

Ancora dall'inizio con un altro esempio

RETI NEURALI – PIÙ LIVELLI



TRIS – APPROCCIO MACHINE LEARNING

Quali sono gli step?

1. costruzione di un **database** di partite da cui apprendere
2. **training** di un modello di Machine Learning
3. **test** delle capacità di Lucy nel gioco del tris

TRIS – APPROCCIO MACHINE LEARNING

Per tutti questi step è fondamentale innanzitutto fissare il **compito della rete**:

TRIS – APPROCCIO MACHINE LEARNING

Per tutti questi step è fondamentale innanzitutto fissare il **compito della rete**:

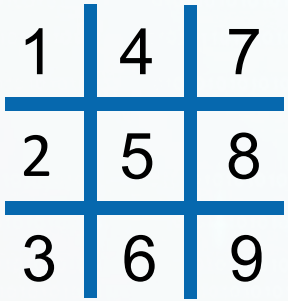
compito: la rete neurale di Lucy deve saper decidere che mossa fare (output) data una certa situazione della scacchiera (input)

1. Database

Costruiamo un **database di mosse**:

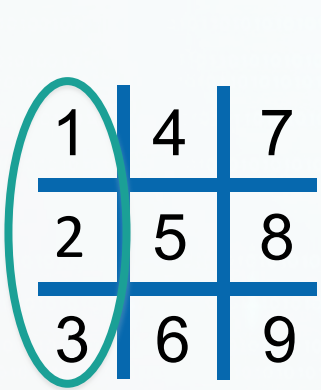
1. Database

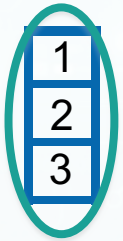
Costruiamo un **database di mosse**:



1. Database

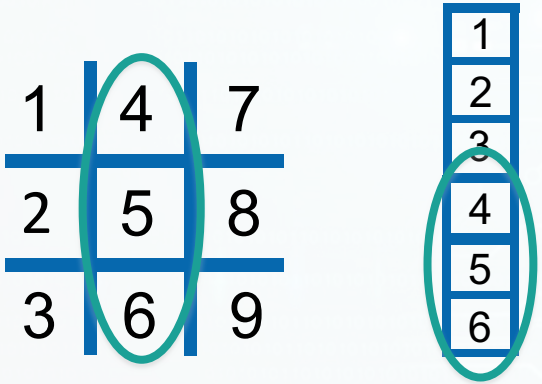
Costruiamo un **database di mosse**:





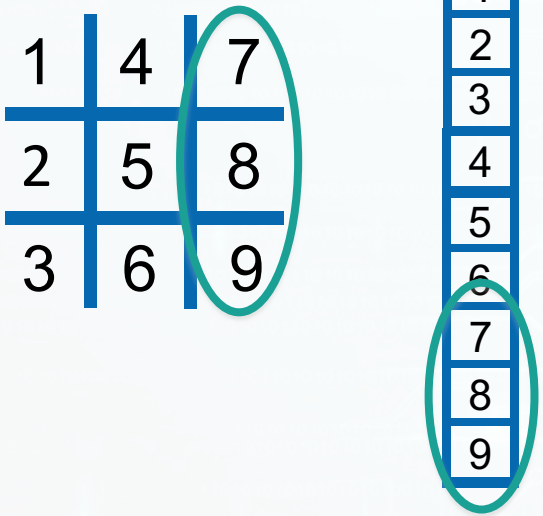
1. Database

Costruiamo un **database di mosse**:



1. Database

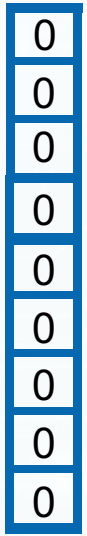
Costruiamo un **database di mosse**:



1. Database

Costruiamo un **database di mosse**:

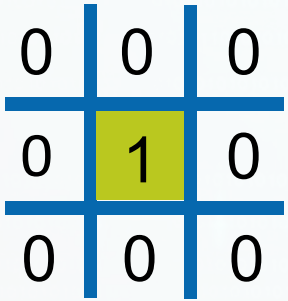
0	0	0
0	0	0
0	0	0

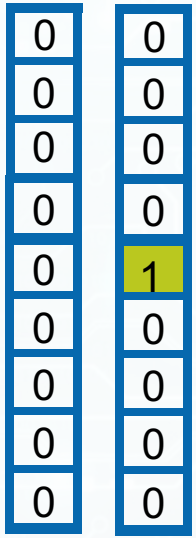


Situazione della scacchiera prima della partita

1. Database

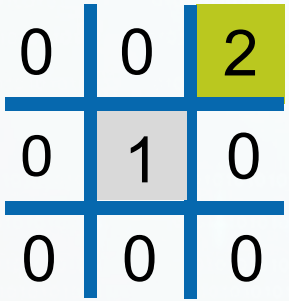
Costruiamo un **database di mosse**:

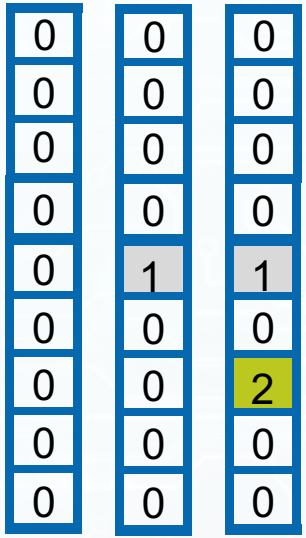




1. Database

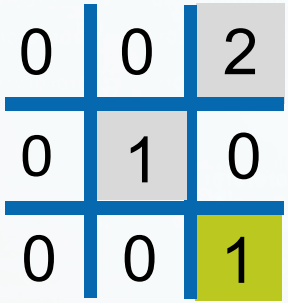
Costruiamo un **database di mosse**:

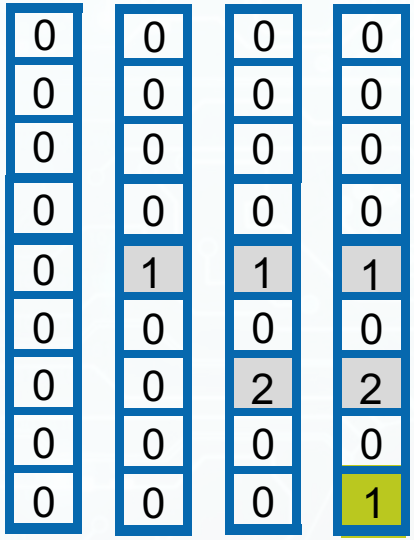




1. Database

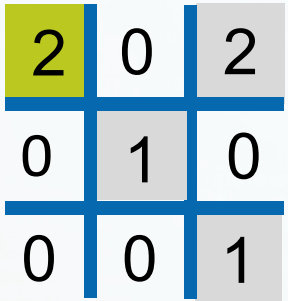
Costruiamo un **database di mosse**:

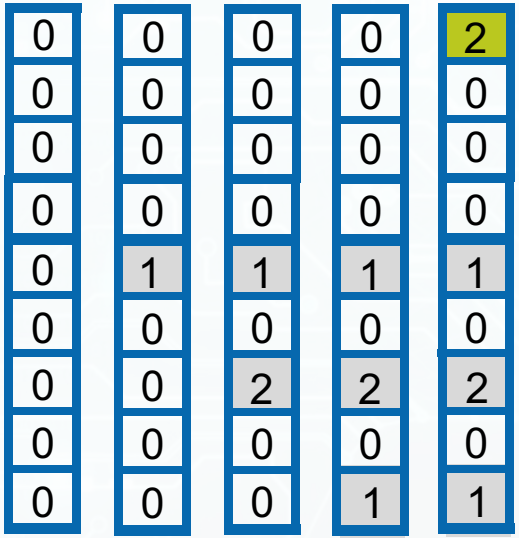




1. Database

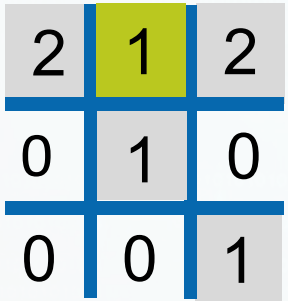
Costruiamo un **database di mosse**:

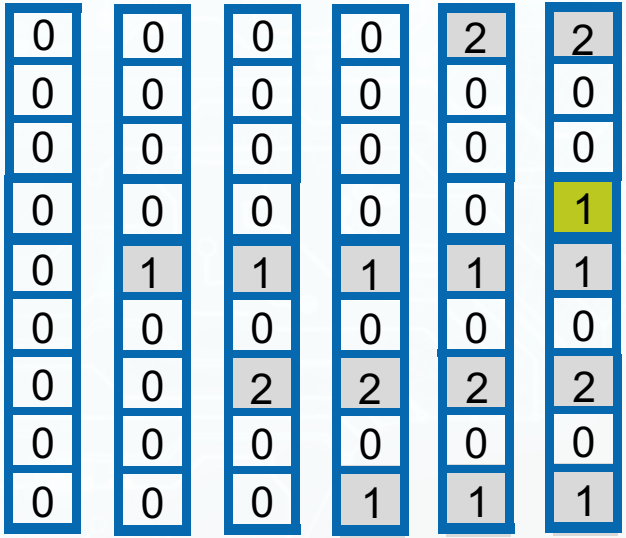




1. Database

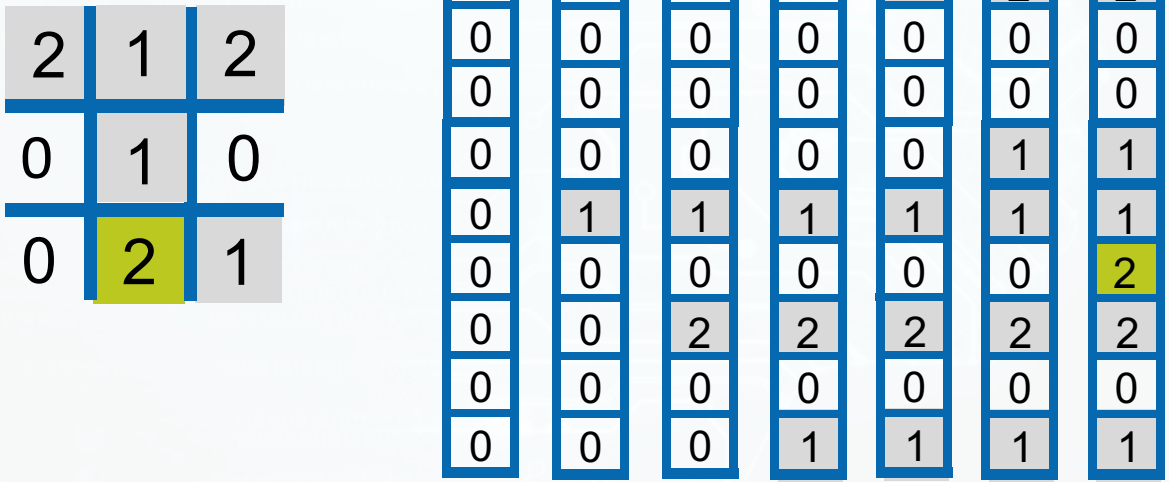
Costruiamo un **database di mosse**:





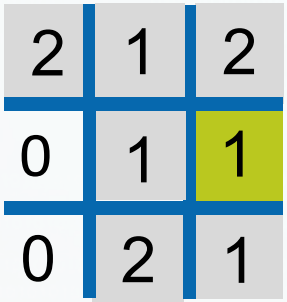
1. Database

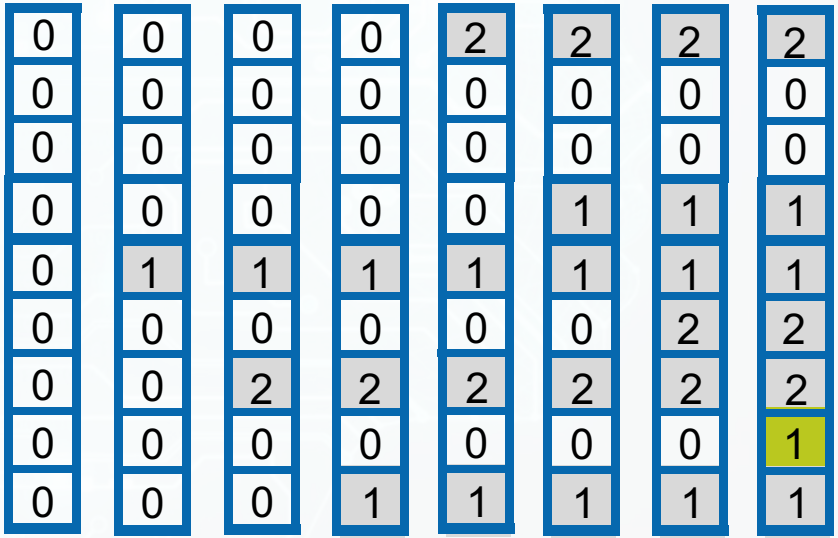
Costruiamo un **database di mosse**:



1. Database

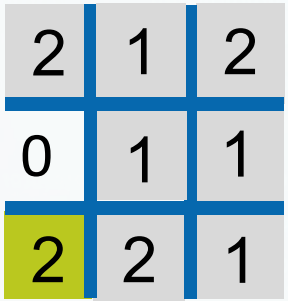
Costruiamo un **database di mosse**:

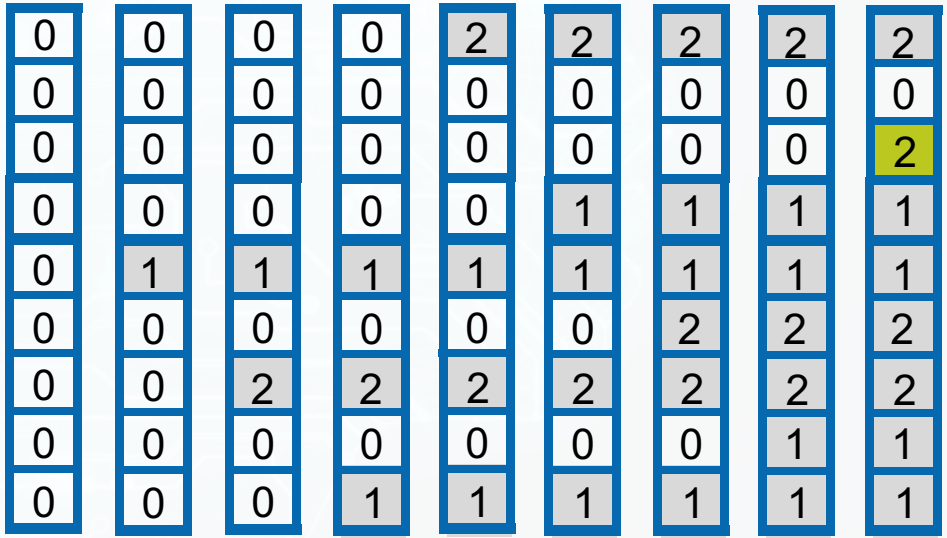




1. Database

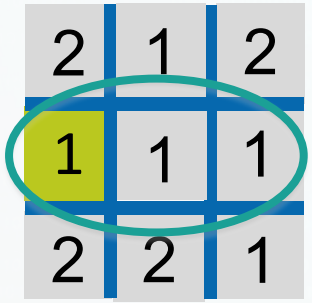
Costruiamo un **database di mosse**:

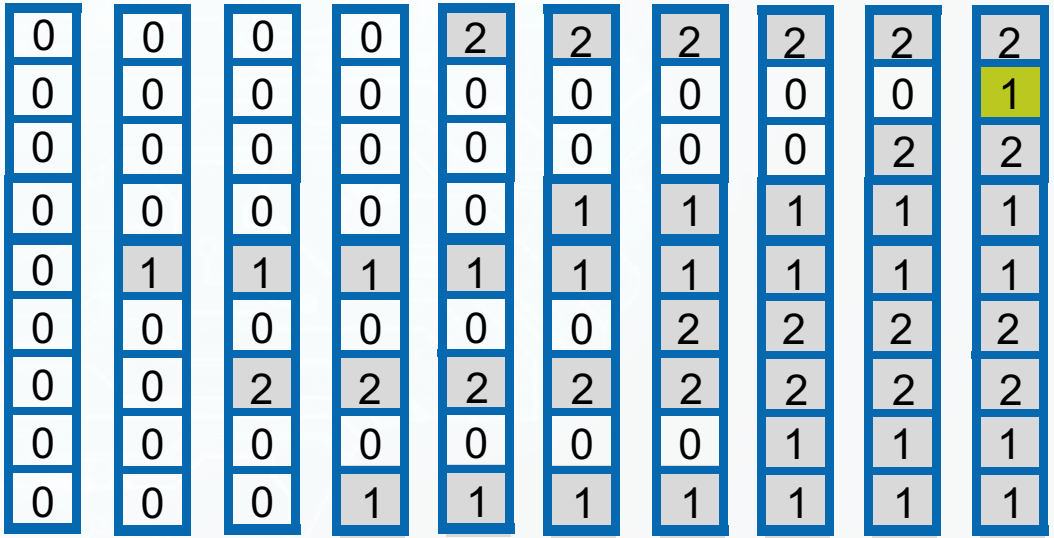




1. Database

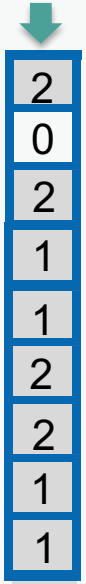
Costruiamo un **database di mosse**:





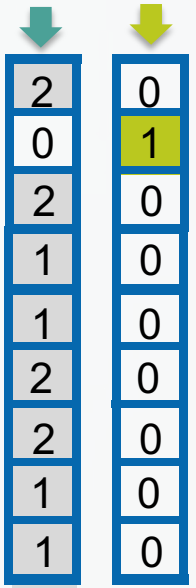
1. Database

Se ha vinto il giocatore 1, data una qualsiasi situazione della scacchiera (examples)



1. Database

Se ha vinto il giocatore 1, data una qualsiasi situazione della scacchiera **(examples),** scelgo le sua mossa come quella da cui imparare **(targets)**

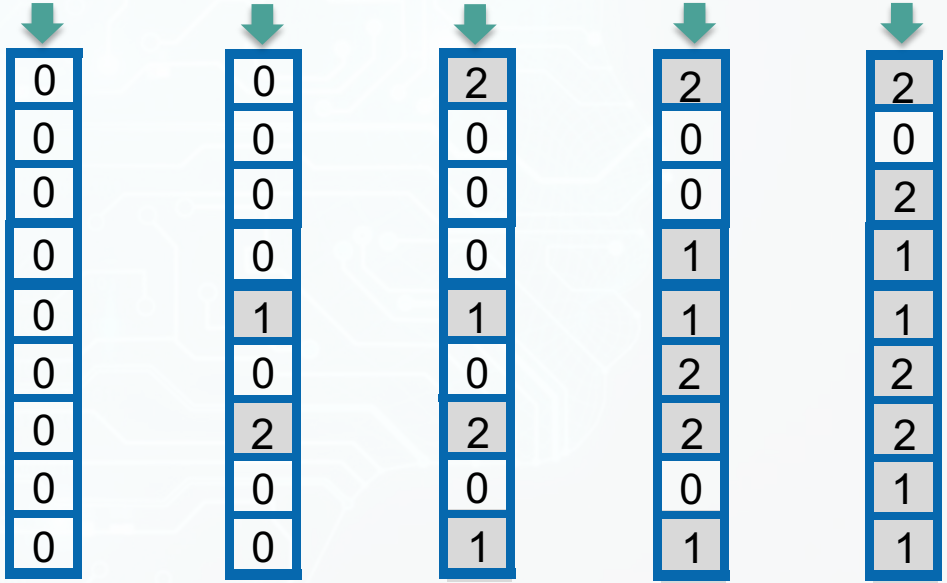


1. Database

Se ha vinto il giocatore 1, data una qualsiasi situazione della scacchiera **(examples),** scelgo le sua mossa come quella da cui imparare **(targets)**

Mosse giocatore perdente

(examples)

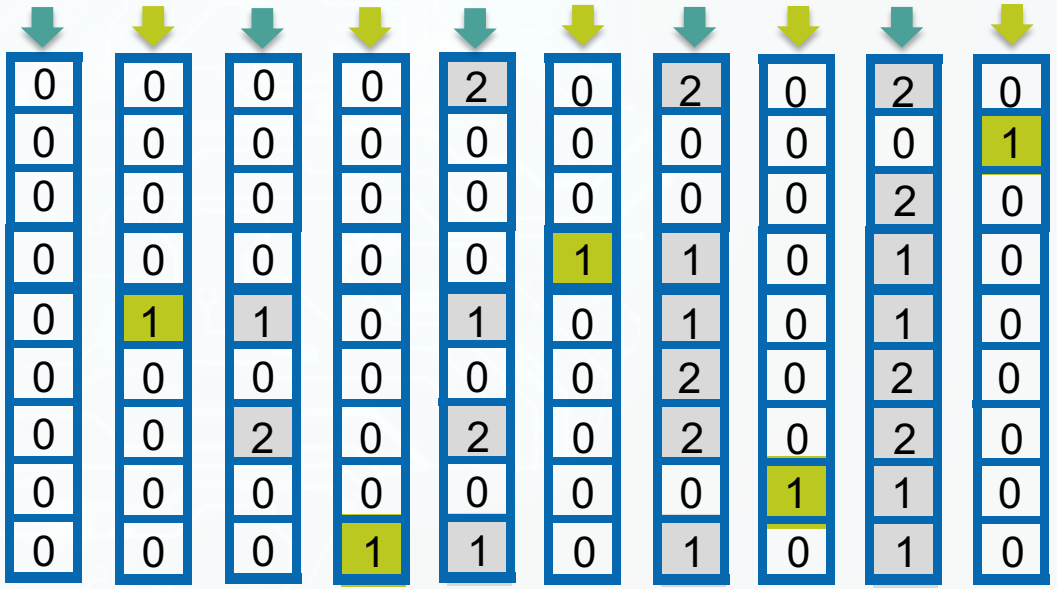


1. Database

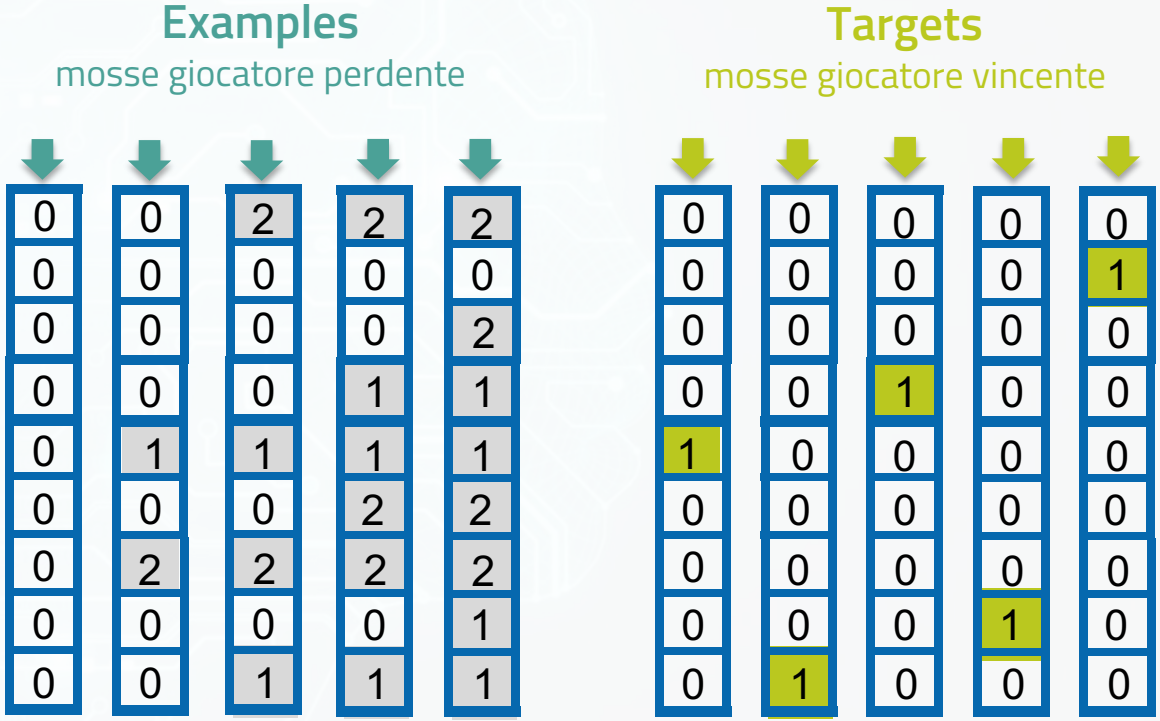
Se ha vinto il giocatore 1, data una qualsiasi situazione della scacchiera (examples), scelgo le sua mossa come quella da cui imparare (targets)

Mosse giocatore vincente

(targets)



1. Database


Examples
mosse giocatore perdente
Targets
mosse giocatore vincente

1. Database

Examples

---

mosse giocatore perdente

0	0	2	2	2
0	0	0	0	0
0	0	0	0	2
0	0	0	1	1
0	1	1	1	1
0	0	0	2	2
0	2	2	2	2
0	0	0	0	1
0	0	1	1	1

Targets

---

mosse giocatore vincente

0	0	2	2	2
0	0	0	0	1
0	0	0	0	2
0	0	1	1	1
1	1	1	1	1
0	0	0	2	2
0	2	2	2	2
0	0	0	1	1
0	1	1	1	1

1. Database

Examples

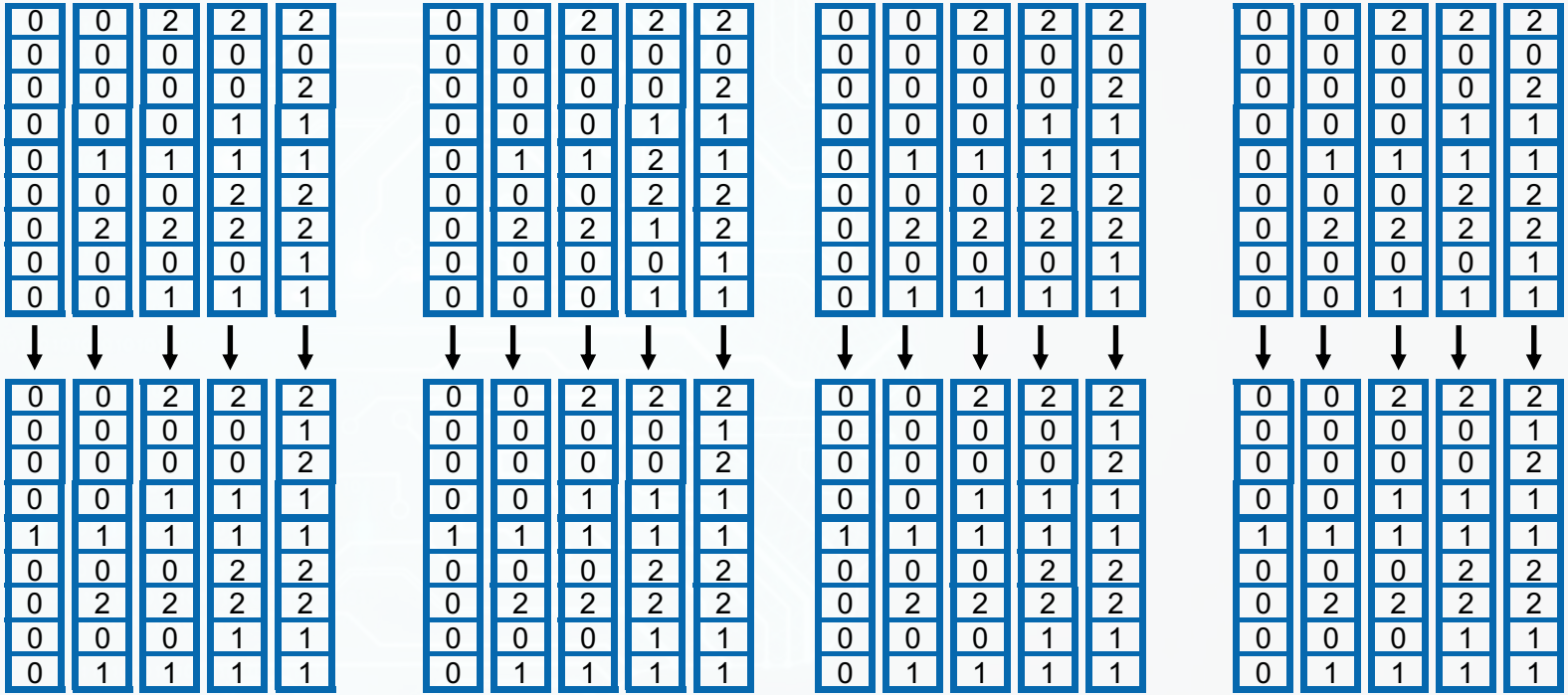
---

mosse giocatore perdente

Targets

---

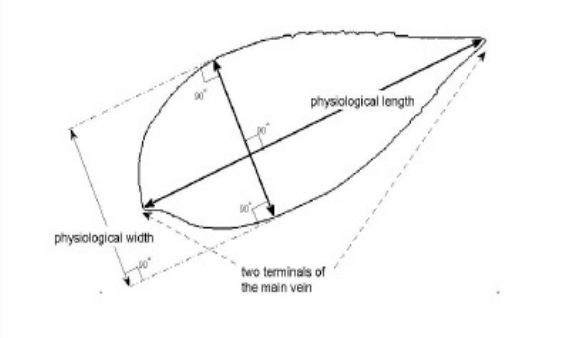
mosse giocatore vincente

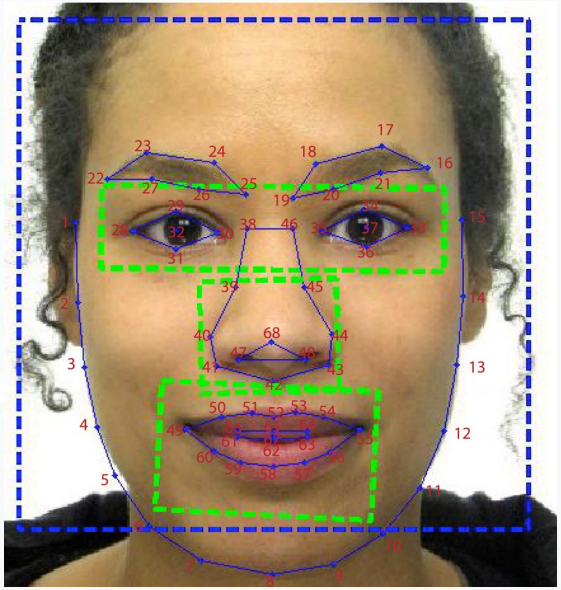


1. Database

Non sarebbe l'unico procedimento, dipende dal problema

→ estrazione di **features**





1. Database

Chi facciamo giocare?

1. Database

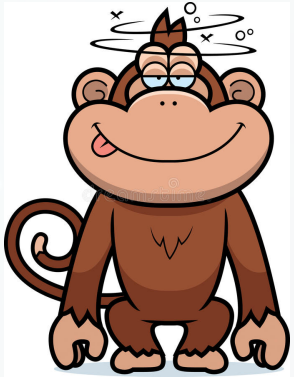
Chi facciamo giocare? Abbiamo a disposizione:

Algoritmo imperativo

1. Database

Chi facciamo giocare? Abbiamo a disposizione:

Algoritmo imperativo



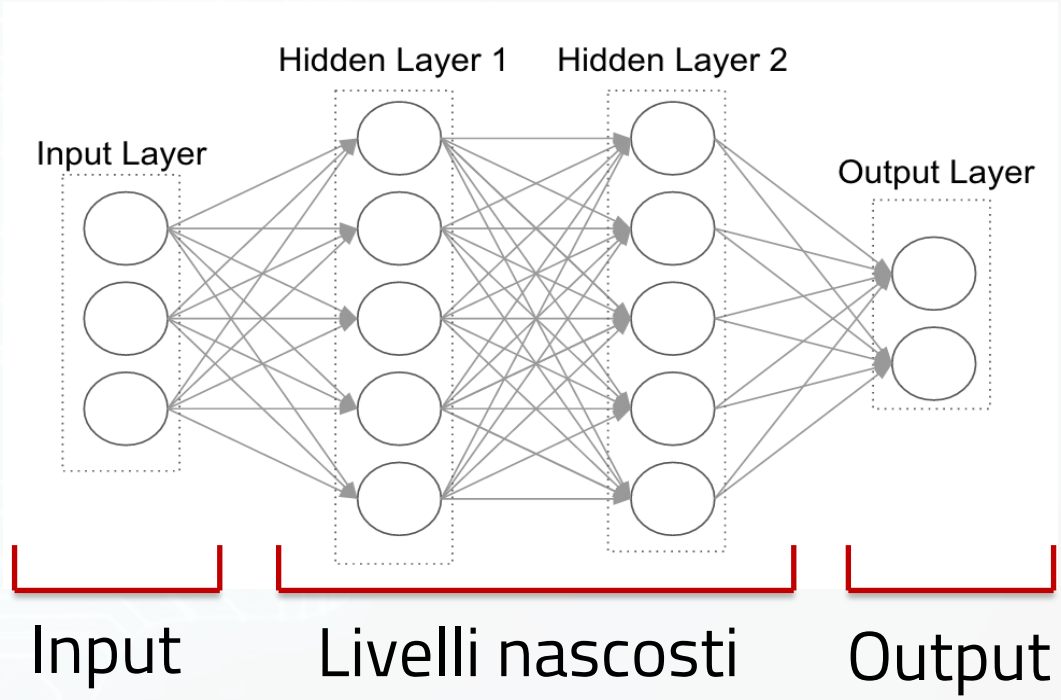
Algoritmo random

1. Database

→ MATLAB

2. Struttura della rete

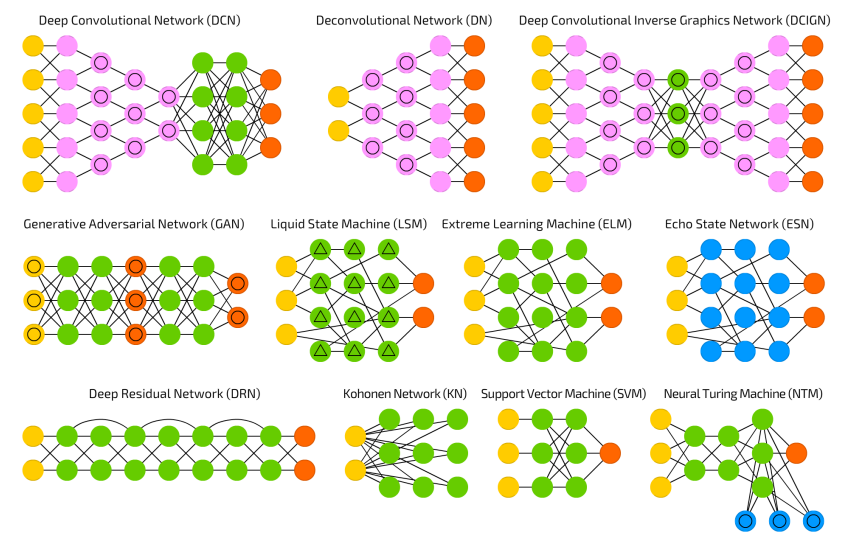
Scegliamo una rete **feed-forward** per il tipo di problema che dobbiamo affrontare.

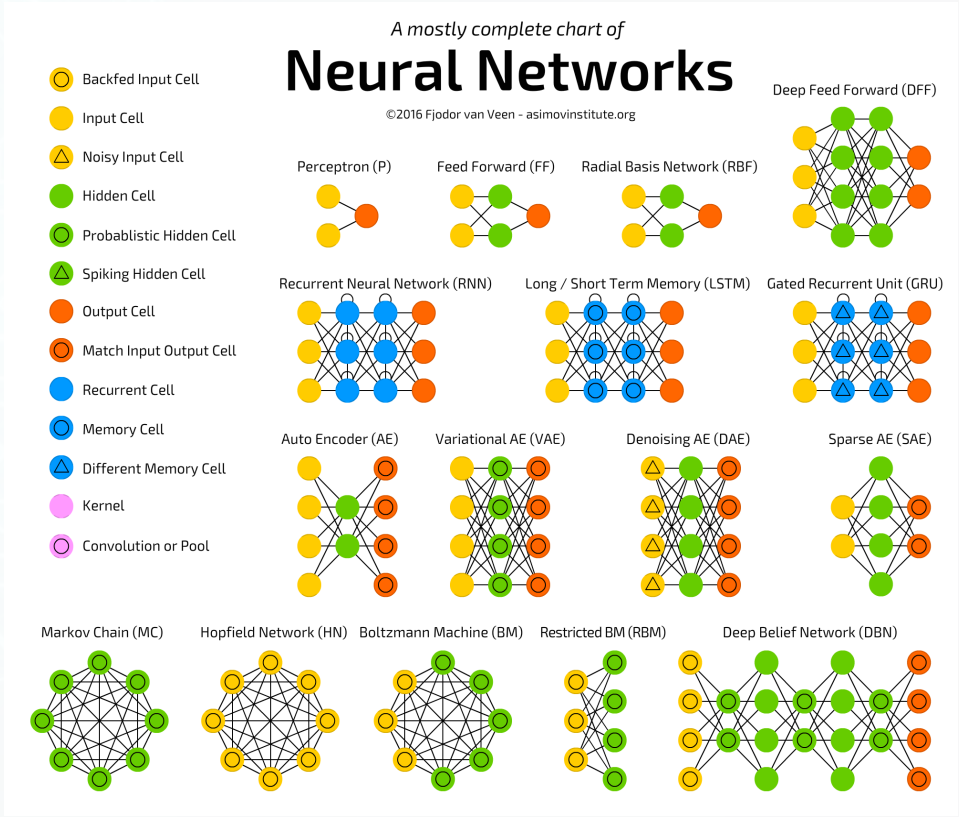


2. Struttura della rete

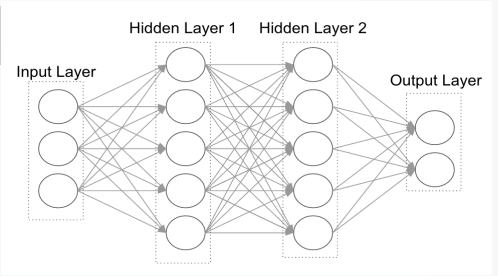
http://www.asimovinstitute.org/neural-network-zoo/

Non sarebbe l'unica scelta...



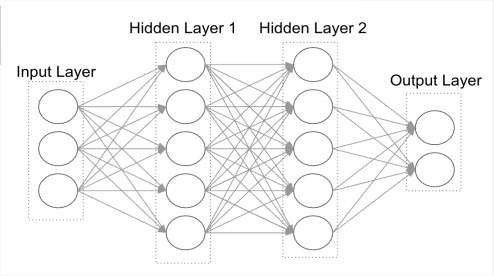


2. Struttura della rete



Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema



2. Struttura della rete

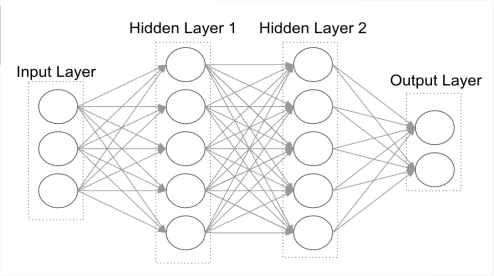
Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema

2	0	2
0	1	0
0	0	1

2
0
0
0
1
0
2
0
1

data una scacchiera qualsiasi **(input)**

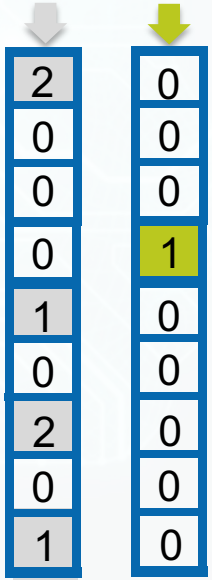


2. Struttura della rete

Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema

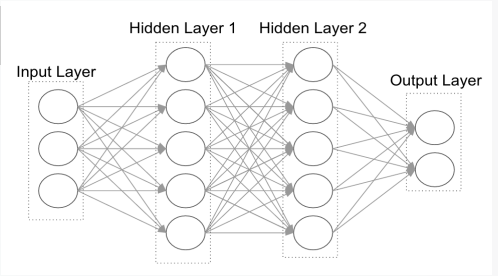
2	1	2
0	1	0
0	0	1



data una scacchiera qualsiasi **(input)**

la rete mi deve dare la mossa successiva **(output)**

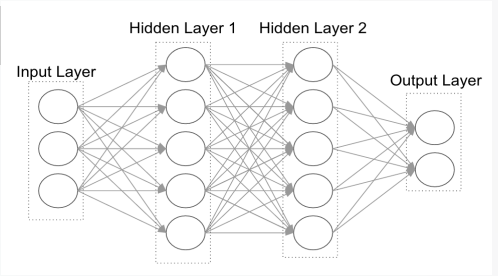
2. Struttura della rete



Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema → **1 input da 9 elementi**
- Quanti OUTPUT? n°output = n° output problema → **1 output da 9 elementi**

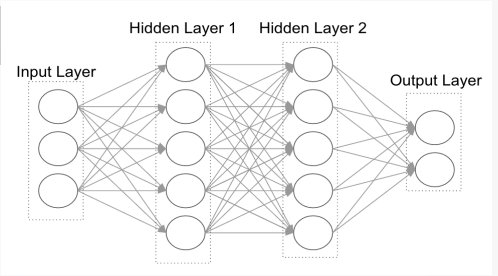
2. Struttura della rete



Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema → **1 input da 9 elementi**
- Quanti OUTPUT? n°output = n° output problema → **1 output da 9 elementi**
- Quanti liv. nascosti? dipende dalla complessità del problema (>2 è già deep learning)

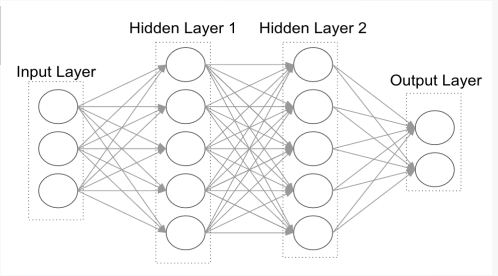
2. Struttura della rete



Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema → **1 input da 9 elementi**
- Quanti OUTPUT? n°output = n° output problema → **1 output da 9 elementi**
- Quanti liv. nascosti? dipende dalla complessità del problema (>2 è già deep learning) → **1 livello nascosto**

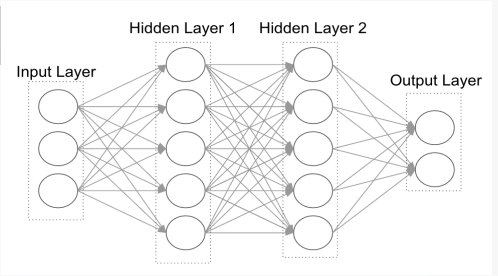
2. Struttura della rete



Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema → **1 input da 9 elementi**
- Quanti OUTPUT? n°output = n° output problema → **1 output da 9 elementi**
- Quanti liv. nascosti? dipende dalla complessità del problema (>2 è già deep learning) → **1 livello nascosto**
- Quanti neuroni? dipende dalla complessità del problema, ma rischio overfitting

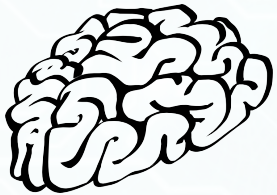
2. Struttura della rete



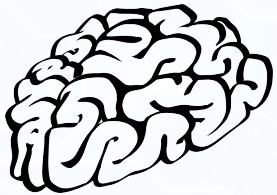
Costruiamo l'**architettura** della rete feed-forward:

- Quanti INPUT? n°input = n° input problema → **1 input da 9 elementi**
- Quanti OUTPUT? n°output = n° output problema → **1 output da 9 elementi**
- Quanti liv. nascosti? dipende dalla complessità del problema (>2 è già deep learning) → **1 livello nascosto**
- Quanti neuroni? dipende dalla complessità del problema, ma rischio overfitting → **30 neuroni**

I.A. – TRAINING E TEST DELLA RETE



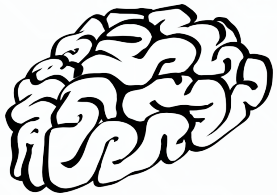
I.A. – TRAINING E TEST DELLA RETE



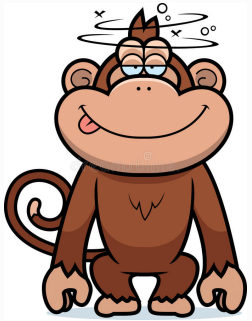
DATABASE

VS

I.A. – TRAINING E TEST DELLA RETE



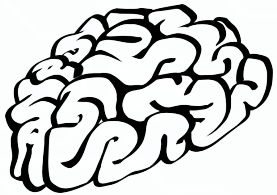
VS



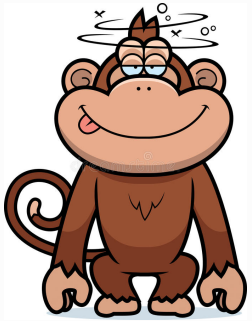
DATABASE

VS

I.A. – TRAINING E TEST DELLA RETE



VS



DATABASE

VS

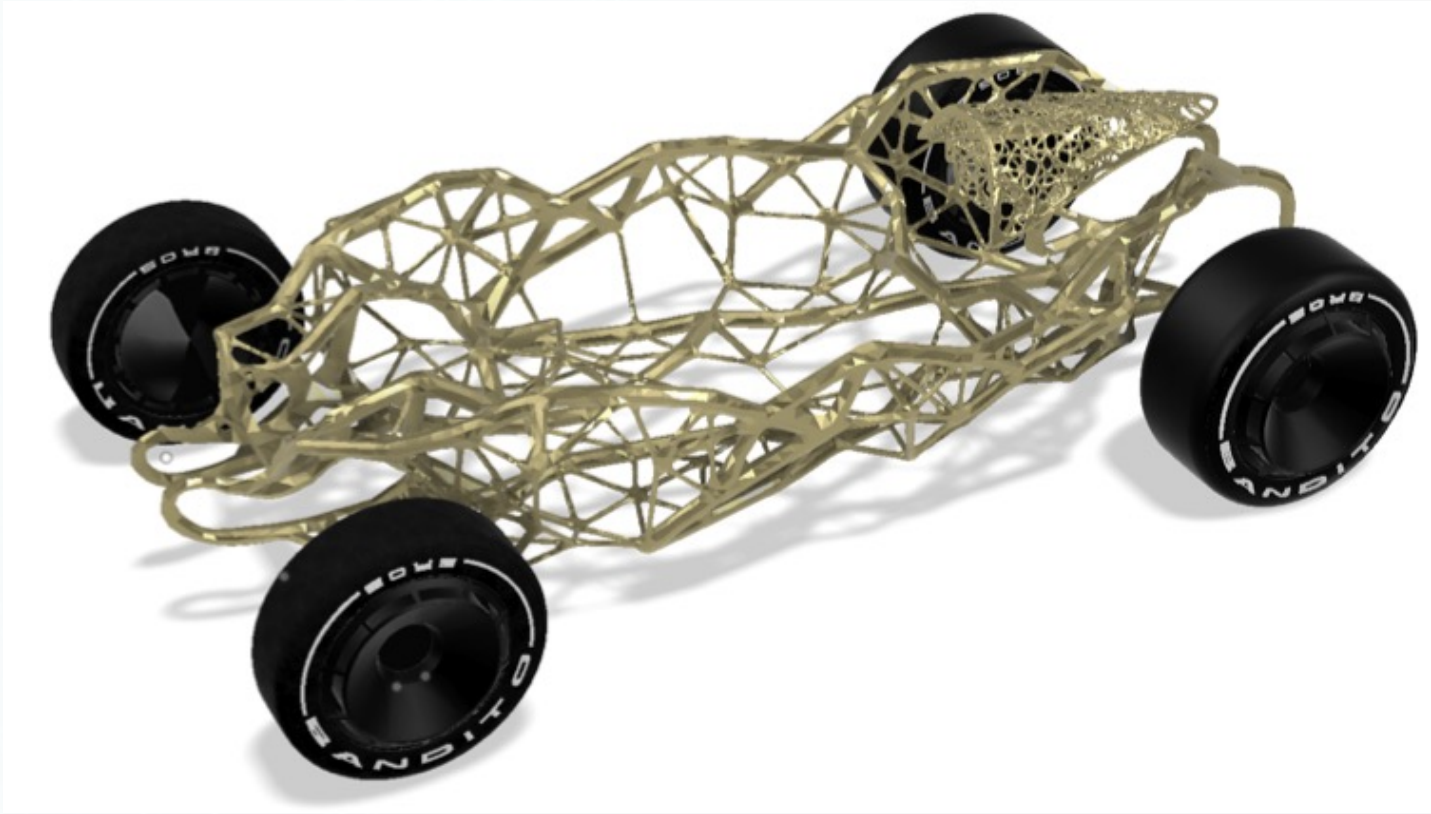
→ MATLAB

Creatività

AlphaGo

Creatività

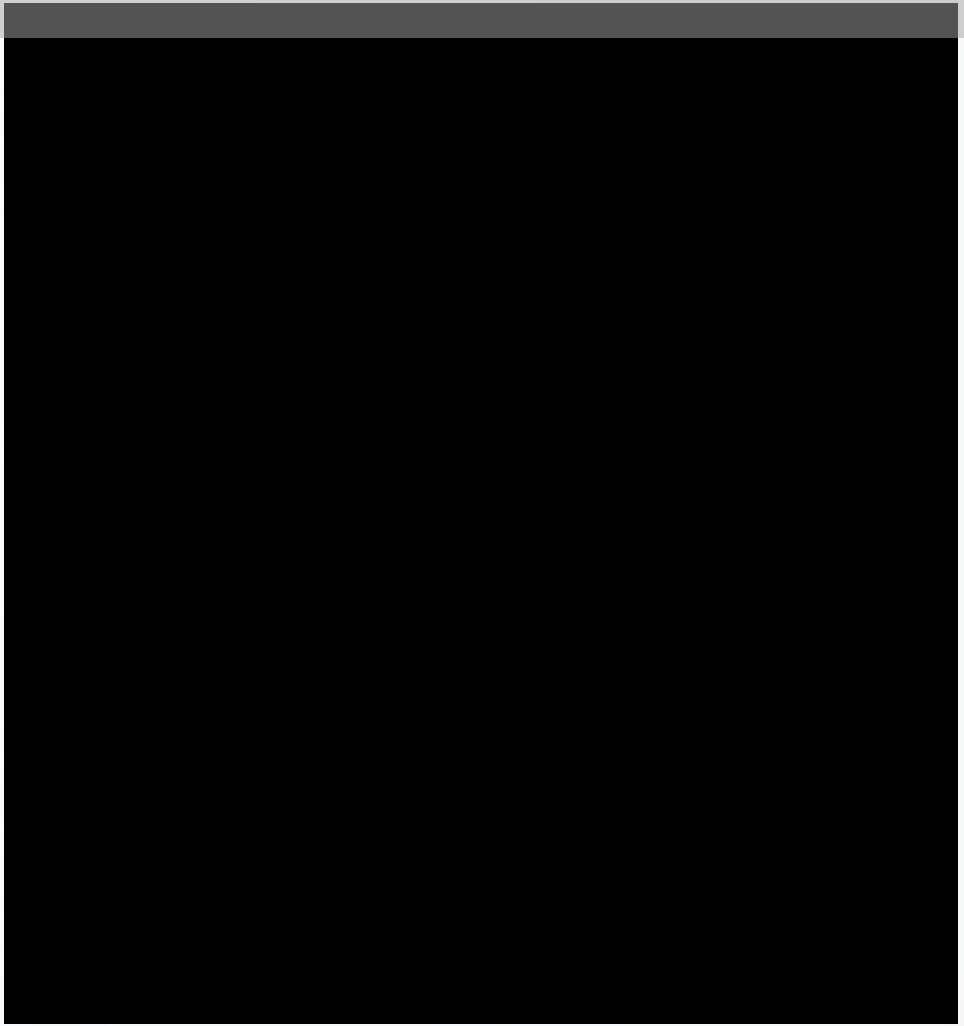
Design



Creatività

- Design

Creatività



Creatività

Creatività

Come ha fatto a trovare queste soluzioni ‘creative’?

Creatività

Come ha fatto a trovare queste soluzioni ‘creative’?

Proviamo ad andare a vedere dentro la rete…

Creatività

Come ha fatto a trovare queste soluzioni ‘creative’?

Proviamo ad andare a vedere dentro la rete…

I pesi dei neuroni del livello nascosto sono:
pesi =

Columns 1 through 8

0.0291	0.1525	0.0701	0.0570	0.0050	0.0125	-0.1629	0.4346
-0.0166	-0.0008	0.2352	0.1899	-0.6553	0.9006	0.0238	-0.6405
0.1245	0.5264	-0.0522	0.0500	0.0079	-0.3409	0.1228	0.0975
-0.0906	0.0095	0.0214	0.0145	0.1118	0.4308	-0.0375	-0.4120
0.1387	0.5143	0.0894	-0.0577	-0.0072	0.3937	0.3187	-0.1469
-0.1931	1.3040	-0.5295	-0.0895	0.2842	-0.6131	0.0421	-0.6583
-0.0427	-0.8549	0.2068	-0.0177	-0.0371	-1.2186	-0.1925	1.2227
-0.0937	-0.1268	0.0279	-0.1022	0.0266	-0.3794	-0.0384	-0.0427
0.1430	0.3354	-0.0442	-0.0642	0.2420	0.3848	-0.0565	-0.0640

Columns 9 through 16

-0.0759	-0.0430	0.5069	0.0617	-0.2819	-0.1783	-0.0339	-0.1018
-0.1113	-0.1593	0.2219	0.0652	-0.2898	-0.7682	0.0966	-0.1292
0.0772	-0.0929	-0.1186	-0.0720	-0.1578	0.8732	-0.0397	-0.0317
0.2212	-0.2835	-0.1149	-0.0118	0.1259	0.7454	-0.0376	0.0020
0.0097	0.4640	-0.0593	0.0850	-0.2828	0.5532	0.1780	-0.4774
-0.1299	-0.2224	-0.1806	-0.2094	-0.0697	0.5152	-0.1985	0.2481
-0.1564	0.0370	-0.2021	0.0219	0.5840	-0.7461	-0.1453	0.0665
0.0551	0.0635	0.0421	-0.1879	-0.0994	-0.6981	0.1123	0.1273
0.1004	0.2116	-0.0521	0.2374	0.5035	-0.3122	0.0753	0.2617

Columns 17 through 24

Creatività

Come ha fatto a trovare queste soluzioni ‘creative’?

Proviamo ad andare a vedere dentro la rete…

I pesi dei neuroni del livello nascosto sono:
pesi =

SUB-SIMBOLICA!

Columns 1 through 8

0.0291	0.1525	0.0701	0.0570	0.0050	0.0125	-0.1629	0.4346
-0.0166	-0.0008	0.2352	0.1899	-0.6553	0.9006	0.0238	-0.6405
0.1245	0.5264	-0.0522	0.0500	0.0079	-0.3409	0.1228	0.0975
-0.0906	0.0095	0.0214	0.0145	0.1118	0.4308	-0.0375	-0.4120
0.1387	0.5143	0.0894	-0.0577	-0.0072	0.3937	0.3187	-0.1469
-0.1931	1.3040	-0.5295	-0.0895	0.2842	-0.6131	0.0421	-0.6583
-0.0427	-0.8549	0.2068	-0.0177	-0.0371	-1.2186	-0.1925	1.2227
-0.0937	-0.1268	0.0279	-0.1022	0.0266	-0.3794	-0.0384	-0.0427
0.1430	0.3354	-0.0442	-0.0642	0.2420	0.3848	-0.0565	-0.0640

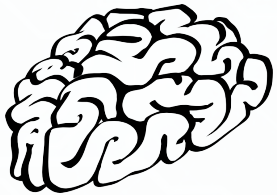
Columns 9 through 16

-0.0759	-0.0430	0.5069	0.0617	-0.2819	-0.1783	-0.0339	-0.1018
-0.1113	-0.1593	0.2219	0.0652	-0.2898	-0.7682	0.0966	-0.1292
0.0772	-0.0929	-0.1186	-0.0720	-0.1578	0.8732	-0.0397	-0.0317
0.2212	-0.2835	-0.1149	-0.0118	0.1259	0.7454	-0.0376	0.0020
0.0097	0.4640	-0.0593	0.0850	-0.2828	0.5532	0.1780	-0.4774
-0.1299	-0.2224	-0.1806	-0.2094	-0.0697	0.5152	-0.1985	0.2481
-0.1564	0.0370	-0.2021	0.0219	0.5840	-0.7461	-0.1453	0.0665
0.0551	0.0635	0.0421	-0.1879	-0.0994	-0.6981	0.1123	0.1273
0.1004	0.2116	-0.0521	0.2374	0.5035	-0.3122	0.0753	0.2617

Columns 17 through 24

-0.0967	-0.2158	-0.4000	-0.2224	-0.0860	-0.0550	-0.0030	-0.1247

I.A. – TRAINING E TEST DELLA RETE



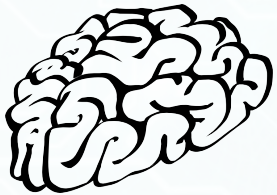
VS

DATABASE

VS

→ MATLAB

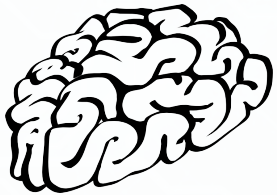
I.A. – TRAINING E TEST DELLA RETE



DATABASE

VS

I.A. – TRAINING E TEST DELLA RETE

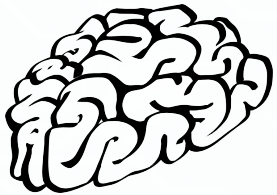


VS

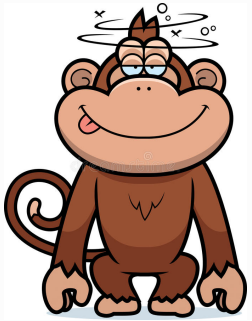
DATABASE

VS

I.A. – TRAINING E TEST DELLA RETE



VS

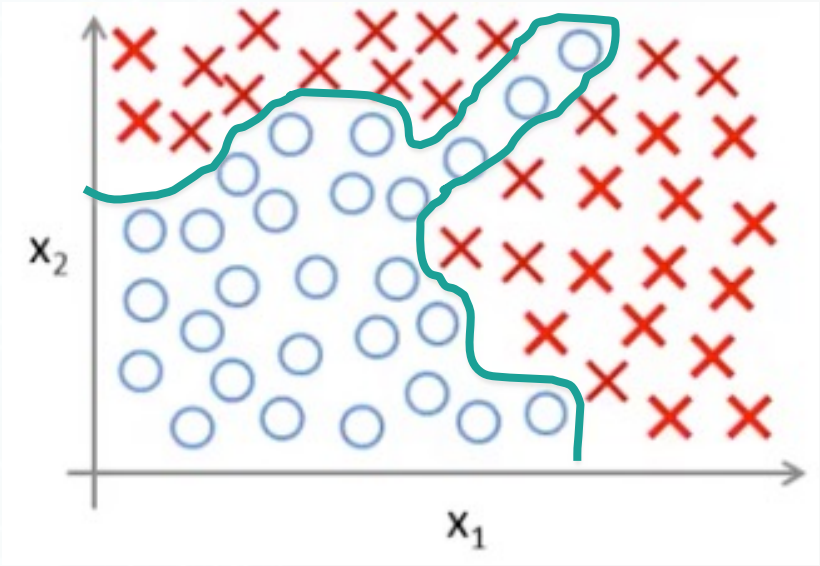


DATABASE

VS

Overfitting

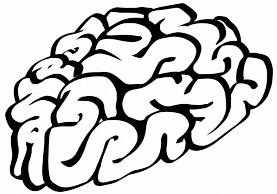
l'insieme dei dati di *training* deve essere **vasto**, ma anche **vario**



Overfitting

l'insieme dei dati di *training* deve essere **vasto**, ma anche **vario**

I.A. – TRAINING E TEST DELLA RETE



VS

DATABASE

VS

3. Allenamento della rete

→ MATLAB

Grazie per l’attenzione.