

Fondazione "I Lincei per la Scuola" "I Lincei per una Nuova Didattica nella Scuola: Una Rete Nazionale"

Matematica 4.0
per le previsioni:
Intelligenza Artificiale,
Machine Learning e
Deep Learning



Prof. Massimiliano Ferrara
Università "Mediterranea" di Reggio Calabria
&
Bocconi ICRIOS

Mercoledì 16 Marzo 2022 - Aula Ciardiello

La **Data Science** è:

- la scienza che si occupa di analizzare i dati e da questi tirare fuori insights e previsioni, utilizzando un approccio “umano”
- l'insieme di regole e processi necessari alla trasformazioni di ipotesi e dati in previsione praticamente utilizzabili



**DATA
SCIENCE**

DATA SCIENTIST

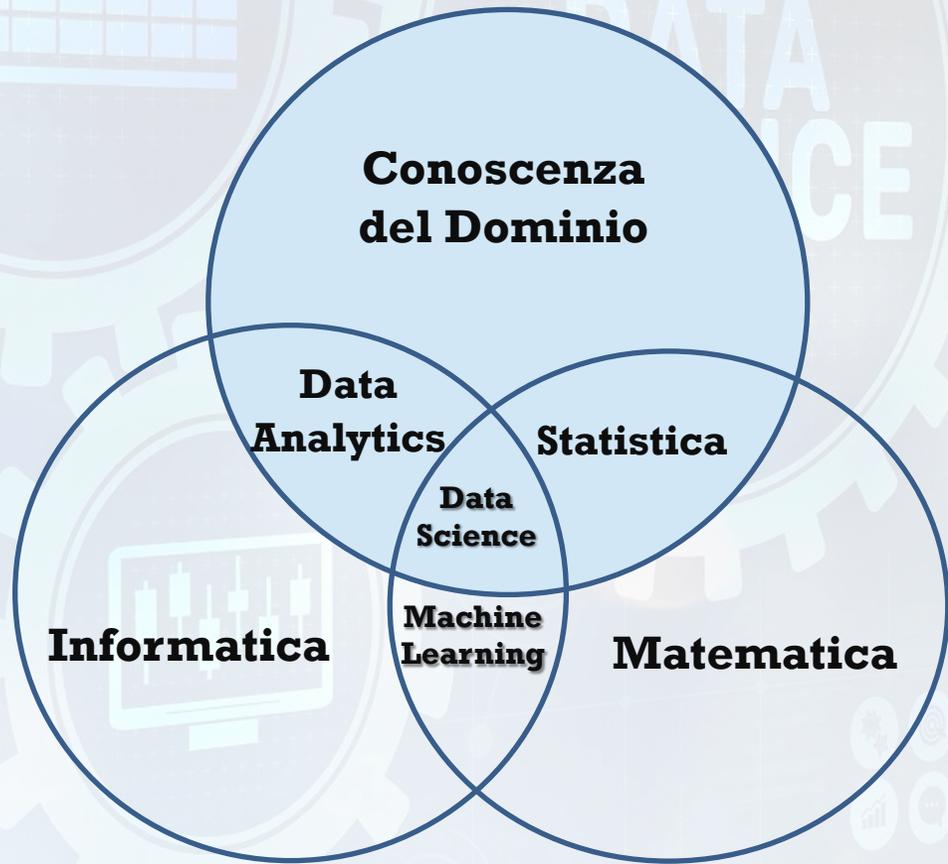
Il Data Scientist utilizza

- tecniche statistiche per produrre analisi, grafici, regressioni, previsioni e tutto quello che la sua abilità riesce a ricavare dai dati a sua disposizione
- algoritmi di Machine Learning e applica i risultati direttamente ai problemi di business che gli vengono proposti
- i dati in modo creativo per generare valore.



DATA SCIENTIST

Il Data Scientist deve avere competenze di statistica, analisi dei dati, conoscenza del dominio del problema che sta affrontando, solide basi informatiche ed un'indispensabile formazione matematica.



LE FASI DELLA DATA SCIENCE

Definizione degli obiettivi

- Definire gli obiettivi
- Creare un progetto

Recuperare i dati

- Dati interni
- Dati Esterni

Preparare i dati

- Pulizia dei dati
- Trasformazione dei dati
- Fusione dei dati

Esplorazione dei dati

- Analisi univariata
- Analisi multivariata
- Correlazioni

Creare il modello

- Feature Engineering
- Modeling
- Validazione

Presentare i risultati

- Presentare i dati
- Autorizzare le analisi

I Sistemi di Supporto alle Decisioni (DSS) sono strumenti informatici interattivi che utilizzano dati e modelli matematici per fornire **supporto ai decision makers** nella risoluzione di problemi complessi.

SUPPORT

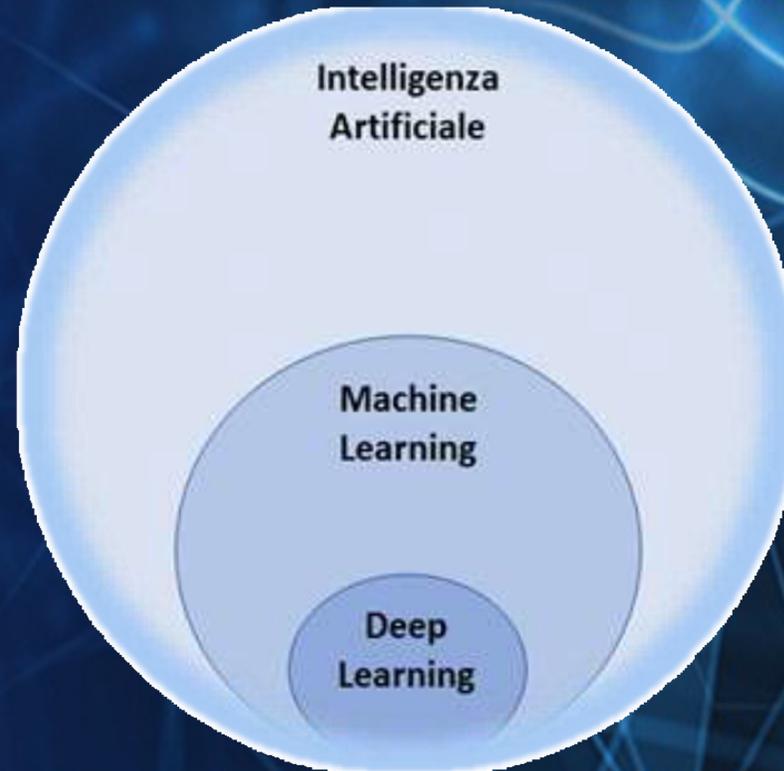


- La funzione principale di un DSS è quella di **estrarre in poco tempo** decisioni da grandi quantità di dati.

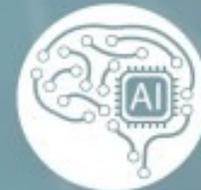
IOT
INTERNET OF THINGS



Il **Machine Learning** è un sottoinsieme **dell'Intelligenza Artificiale** e, insieme al **Deep Learning**, costituisce un approccio moderno che sta riscuotendo un notevole successo



L'Intelligenza Artificiale (IA) (acronimo inglese Artificial Intelligence (AI)) può essere definita come la scienza che si propone di sviluppare macchine intelligenti.



L'Intelligenza Artificiale si occupa della

1. comprensione

2. riproduzione

del comportamento intelligente



L'IA COME SCIENZA EMPIRICA

L'approccio della psicologia cognitiva:

- **Obiettivo:** comprensione dell'intelligenza umana
- **Metodo:** costruzione di modelli computazionali, verifica sperimentale
- **Criterio di successo:** risolvere i problemi con gli stessi processi usati dall'uomo



L'IA come disciplina informatica

- L'approccio “costruttivo”:
- **Obiettivo**: costruzione di entità dotate di razionalità
- **Metodo**: codifica del pensiero razionale; comportamento razionale
- **Criterio di successo**: l'importante è risolvere i problemi che richiedono intelligenza

Definizioni di IA

umanamente

razionalmente

Pensare

“[L’automazione delle] attività che associamo al pensiero umano, come il processo decisionale, la risoluzione di problemi, l’apprendimento ...” [Bellman 1978]

“Lo studio delle facoltà mentali attraverso l’uso di modelli computazionali” [Charniak, McDermott, 1985]

agire

“L’arte di creare macchine che svolgono funzioni che richiedono intelligenza quando svolte da esseri umani” [Kurzweil 1990]

“Il ramo della scienza dei calcolatori che si occupa dell’automazione del comportamento intelligente” [Luger-Stubblefield 1993]

“L’impresa di costruire artefatti intelligenti” [Ginsberg 1993]

DA “STRATEGIC DIRECTIONS ... [1996]”

Il settore dell'IA consiste nell'indagine tecnologica e intellettuale, a lungo termine, che mira al raggiungimento dei seguenti obiettivi scientifici e pratici:

- costruzione di macchine intelligenti, sia che operino come l'uomo che diversamente;**
- formalizzazione della conoscenza e meccanizzazione del ragionamento, in tutti i settori di azione dell' uomo;**

DA “STRATEGIC DIRECTIONS ... [1996]”

- comprensione mediante modelli computazionali della psicologia e comportamento di uomini, animali e agenti artificiali;
- rendere il lavoro con il calcolatore altrettanto facile e utile che del lavoro con persone, capaci, cooperative e possibilmente esperte.

[da Strategic Directions in Artificial Intelligence, ACM 1996]

ECONOMIA

INFORMATICA

MATEMATICA

FILOSOFIA

NEUROSCIENZE

LINGUISTICA

PSICOLOGIA

GIURISPRUDENZA

**TEORIA DEL CONTROLLO E
CIBERNETICA**

NLP: è un sottocampo della linguistica, dell'informatica e dell'intelligenza artificiale che si occupa delle interazioni tra computer e linguaggio umano

QUESTION TIME

**Ma cosa significa
“intelligente”?**

**Provate a dare la
vostra definizione**



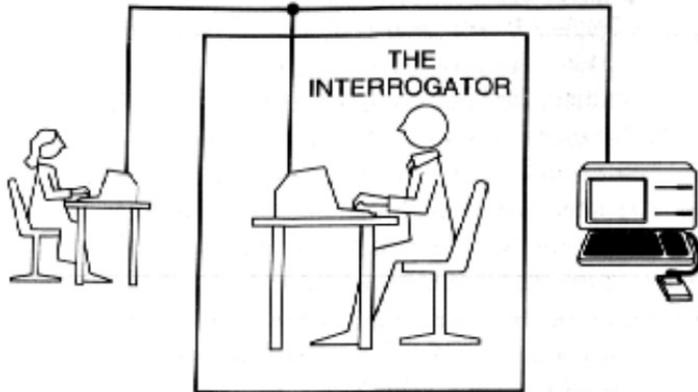
CHE TIPO DI CAPACITÀ?

- Capacità di simulare il comportamento umano?
- Capacità di ragionamento logico/matematico?
- Intelligenza come competenza “da esperto”?
- Intelligenza come “buon senso” (senso comune)?
- Capacità di interagire con un ambiente?
- Capacità sociali, di comunicazione e coordinamento?
- Capacità di comprendere e provare emozioni?
- Capacità di “immagazzinare” esperienza?



CAPACITÀ DI IMITAZIONE?

Test di Turing (1950): un tentativo di definizione operativa di intelligenza



Ogni aspetto dell'apprendimento o ogni altra caratteristica dell'intelligenza può essere descritta in modo talmente preciso da consentire la costruzione di una macchina in grado di simularla".



PREVISIONI

«Credo che tra circa 50 anni sarà possibile programmare computer con una memoria di un miliardo di byte in maniera tale che essi giochino il gioco dell'imitazione tanto bene che una persona comune non avrà più del 70% di probabilità di identificarli dopo 5 minuti di interrogazione»

Computing machinery and intelligence, Turing 1950

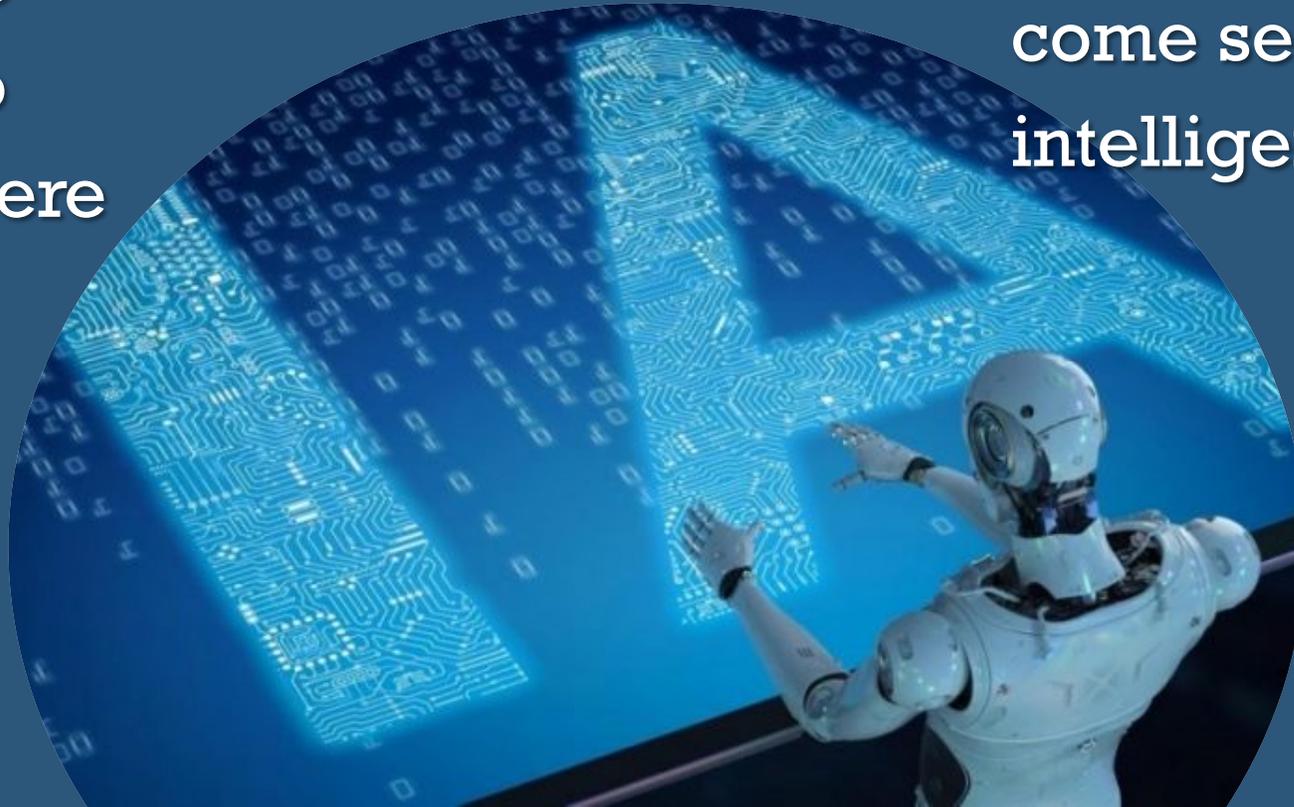


INTELLIGENZA ARTIFICIALE FORTE E DEBOLE

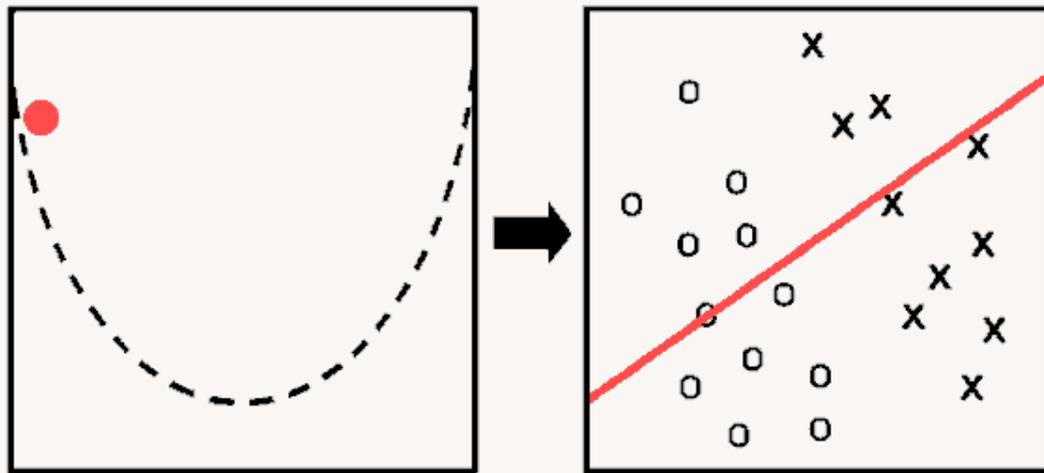
La **teoria dell' IA forte** si basa sulla convinzione che le macchine possano effettivamente essere intelligenti.

La **teoria dell' IA debole**, invece, in modo più realistico e pragmatico.

Pensa che le macchine possano comportarsi come se fossero intelligenti.



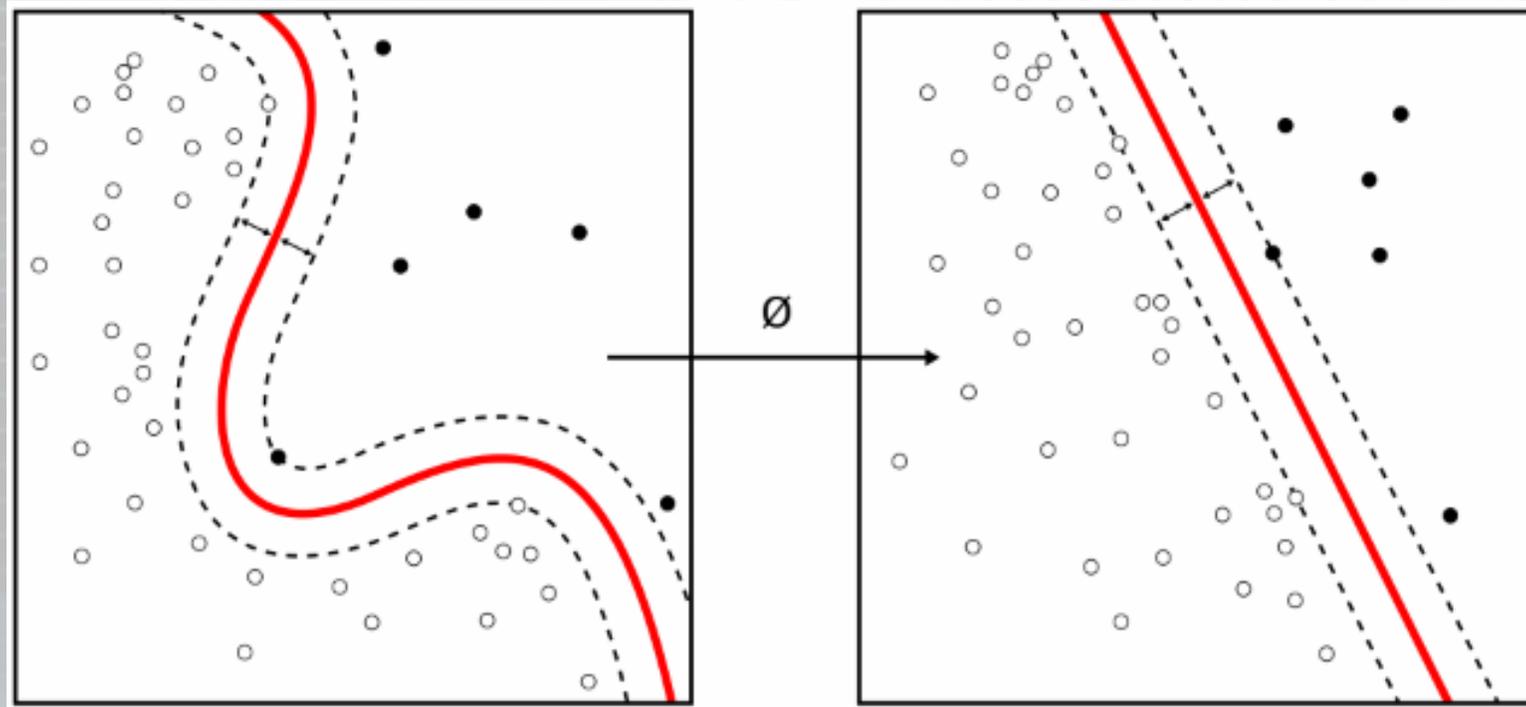
Metodi di ottimizzazione



Il training e la valutazione trasformano gli algoritmi di apprendimento supervisionato in **modelli** ottimizzando i pesi dei parametri per trovare l'insieme di valori che si adattano meglio alla realtà fondamentale espressa dai dati.

Per l'ottimizzazione gli algoritmi spesso si basano su metodi di analisi della “discesa più ripida”, ad esempio la **discesa stocastica del gradiente (SGD)**, che è essenzialmente la ricerca della discesa più ripida eseguita più volte da punti di partenza casuali.

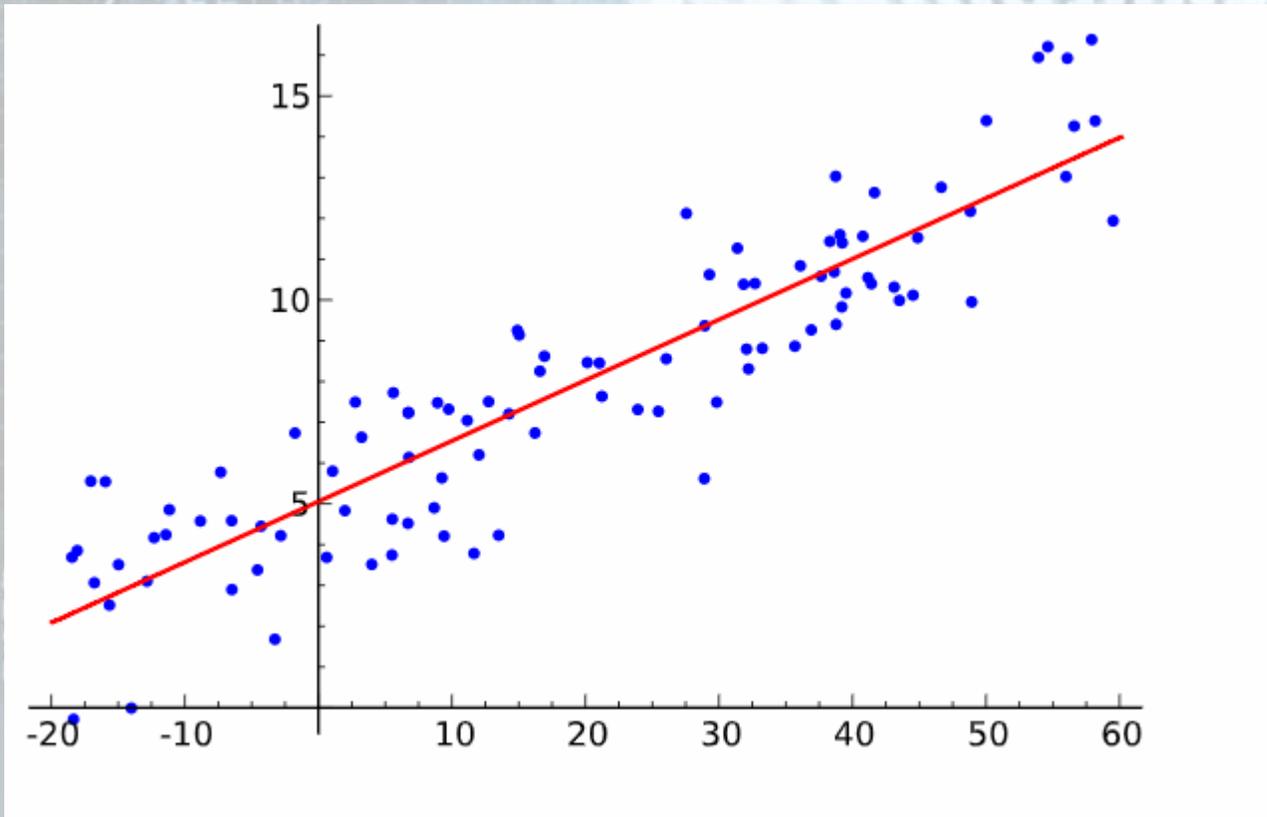
Algoritmi di classificazione



Un problema di classificazione è un problema di apprendimento supervisionato che richiede di fare una scelta tra due o più classi da attribuire ai dati, in genere fornendo una probabilità per ogni classe.

Un algoritmo di classificazione individua la funzione che meglio descrive le zone che separano i dati che presentano caratteristiche differenti tra loro. (Immagine di Alisneaky, licenza [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/))

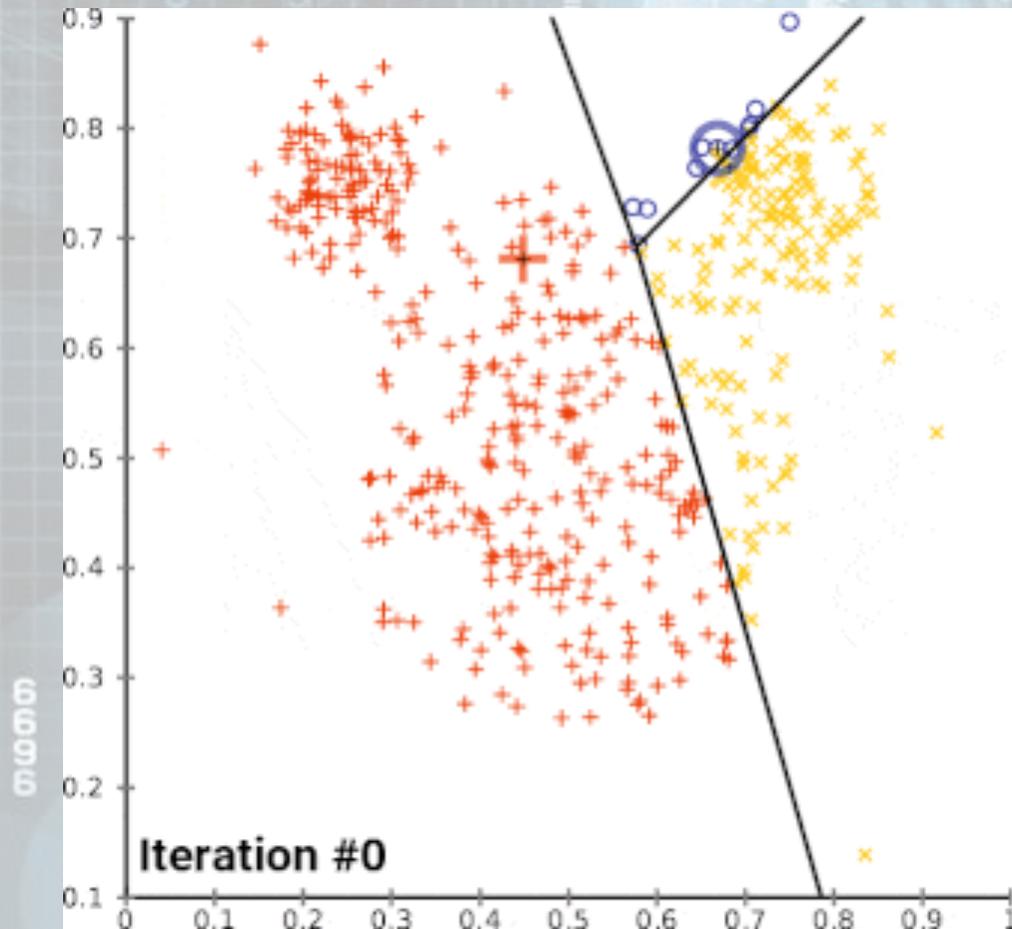
Algoritmi di regressione



La regressione individua una funzione che traccia una linea che minimizzi la distanza media tra i punti.

Un problema di regressione è un problema di apprendimento supervisionato che richiede al modello di prevedere un numero. L'algoritmo più semplice e veloce è la **regressione lineare** (metodo dei minimi quadrati), ma bisognerebbe fermarsi qui, perché spesso offre un risultato mediocre.

Algoritmi di clustering



Un problema di **clustering** è un problema di apprendimento senza supervisione che richiede al modello di trovare gruppi di punti dati che presentino delle similarità tra loro.

L'algoritmo K-Means affina il raggruppamento dei dati (Immagine di Chire, licenza [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/))

MACHINE LEARNING E DEEP LEARNING

1

MACHINE LEARNING

Si occupa principalmente di fare delle previsioni, «imparando» dai dati



2

DEEP LEARNING

E' un sottoinsieme del Machine Learning ed entra in gioco quando il Machine Learning non riesce ad ottenere i risultati desiderati

ALGORITMI DI MACHINE LEARNING

APPRENDIMENTO SUPERVISIONATO



Questi algoritmi utilizzano dataset in cui è già presente la «risposta giusta».

Da questi apprendono ed imparano a prevedere la risposta per un nuovo insieme di dati, che non è presente nell'insieme utilizzato per l'addestramento.

Questi algoritmi utilizzano dataset che non hanno delle risposte disponibili.

E' l'algoritmo stesso che, analizzando i dati, produce delle relazioni tra gli stessi e genera quelle che possono essere le possibili risposte, aiutando a scoprire nuove informazioni presenti nei dati.



APPRENDIMENTO NON SUPERVISIONATO

ALGORITMI DI MACHINE LEARNING

APPRENDIMENTO SEMI SUPERVISIONATO



Questi algoritmi utilizzano dataset in cui la risposta è disponibile solo per una piccola parte degli esempi.

Si proverà, quindi, a sfruttare al meglio i pochi dati con risposta disponibile per ottenere una regola generale da sfruttare per i dati senza risposta

Questi algoritmi non prevedono un dataset di addestramento, per quest non hanno un'esperienza da utilizzare per imparare.

Pertanto da una situazione iniziale qualsiasi e, attraverso regole di premiazione o penalizzazione, cercano di arrivare al risultato ottimale.



APPRENDIMENTO PER RINFORZO

FASI DEL MACHINE LEARNING

Un programma apprende se dopo aver eseguito un compito migliorano le prestazioni





Tech Help Desk Future of Transportation Innovations

Innovations

The battle to prevent another Jan. 6 features a new weapon: The algorithm

A year after the attack on the Capitol, data scientists say artificial intelligence can help forecast insurrection — with some big concerns



DEEP LEARNING

- Il *Deep Learning* (o apprendimento profondo, talvolta tradotto con apprendimento approfondito) è, invece, una **branca del *Machine Learning***, supervisionato o meno, che caratterizza i processi di **reti neurali artificiali** dotate di due o più strati capaci di processare informazioni in modo **non lineare**.
- Il *Deep Learning* combina computer sempre più potenti a speciali tipi di sistemi neuronali **per apprendere i complicati schemi dei grandi volumi di dati**.

Il *Deep Learning* può essere definito come un sistema che sfrutta una classe di algoritmi di apprendimento automatico che:

- usano **vari livelli di unità non lineari a cascata per svolgere compiti di estrazione di caratteristiche e di trasformazione**. Ciascun livello successivo utilizza l'uscita del livello precedente come input. Gli algoritmi possono essere sia di tipo supervisionato sia non supervisionato e le applicazioni includono l'analisi di pattern (apprendimento non supervisionato) e classificazione (apprendimento supervisionato);
- sono basati sull'**apprendimento non supervisionato di livelli gerarchici multipli di caratteristiche (e di rappresentazioni) dei dati**. Le caratteristiche di più alto livello vengono derivate da quelle di livello più basso per creare una rappresentazione gerarchica;
- apprendono multipli livelli di rappresentazione che corrispondono a differenti livelli di astrazione; questi livelli formano una **gerarchia di concetti**

Seppur la richiesta di capacità computazionali enormi possa rappresentare un limite, **la scalabilità del *Deep Learning* grazie all'aumento dei dati disponibili e degli algoritmi è ciò che lo differenzia dal *Machine Learning*:**

- i sistemi di *Deep Learning*, infatti, migliorano le proprie prestazioni all'aumentare dei dati mentre le applicazioni di *Machine Learning* (o meglio, i cosiddetti 'sistemi di apprendimento superficiale') una volta raggiunto un certo livello di performance non sono più scalabili nemmeno aggiungendo esempi e dati di training alla rete neurale.

Grazie a modelli **di reti neurali** particolarmente complessi e gerarchicamente organizzati, riesce ad elaborare grosse moli di dati e ad ottenere risultati più precisi, ovviamente a costo di una potenza di calcolo ed un tempo di elaborazione maggiore.



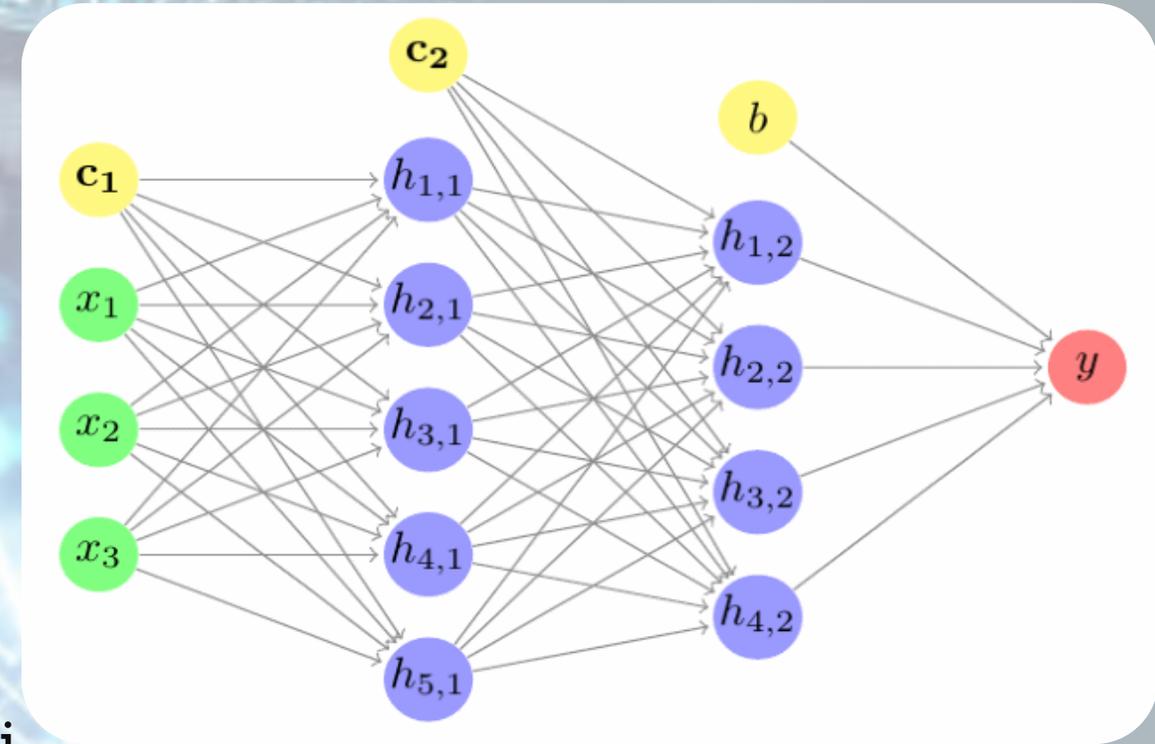
LE RETI NEURALI

- ✓ Le reti neurali sono basate sull'idea di riprodurre l'intelligenza ed in particolare l'apprendimento, simulando la struttura neurale del cervello
- ✓ L'idea di costruire una macchina intelligente a partire da neuroni artificiali si può far risalire alla nascita dell'*Intelligenza Artificiale*, e già alcuni risultati furono ottenuti da McCulloch e Pitts nel 1943 quando nacque il primo modello neurale.
- ✓ Nel 1962 Rosenblatt propose un nuovo modello di neurone, il «perceptrone», capace di apprendere mediante esempi.

La struttura di una rete neurale

Una rete neurale è una rete divisa in tre livelli:

- **Input:** in questo livello vengono trattati i dati attraverso una prima funzione (matematica) di attivazione che può introdurre la non linearità (tangente, coseno, sigmoide, logaritmo...)
- **Hidden:** eseguono una combinazione lineare (somma pesata) dell'output ottenuto dall'Input Layer; l'output di questi neuroni è una funzione di attivazione della combinazione; in una rete neurale possono esser presenti più hidden layer
- **Output:** stesso funzionamento dell'hidden layer; sono il livello finale della rete e l'output di questi neuroni è il risultato dell'intera rete





Con il **Deep Learning** vengono simulati i processi di apprendimento del cervello biologico attraverso le reti neurali artificiali per insegnare alle macchine non solo ad apprendere autonomamente ma a farlo in modo più “profondo” come sa fare il cervello umano dove profondo significa “su più livelli” (vale a dire sul numero di **hidden layer**: quelle “tradizionali” contengono 2-3 layer, mentre le reti neurali profonde possono contenerne oltre 150).



**INTELLIGENZA ARTIFICIALE E TEORIA
DELLE DECISIONI:**

LA MACCHINA PUÒ SOSTITUIRE L'UOMO?

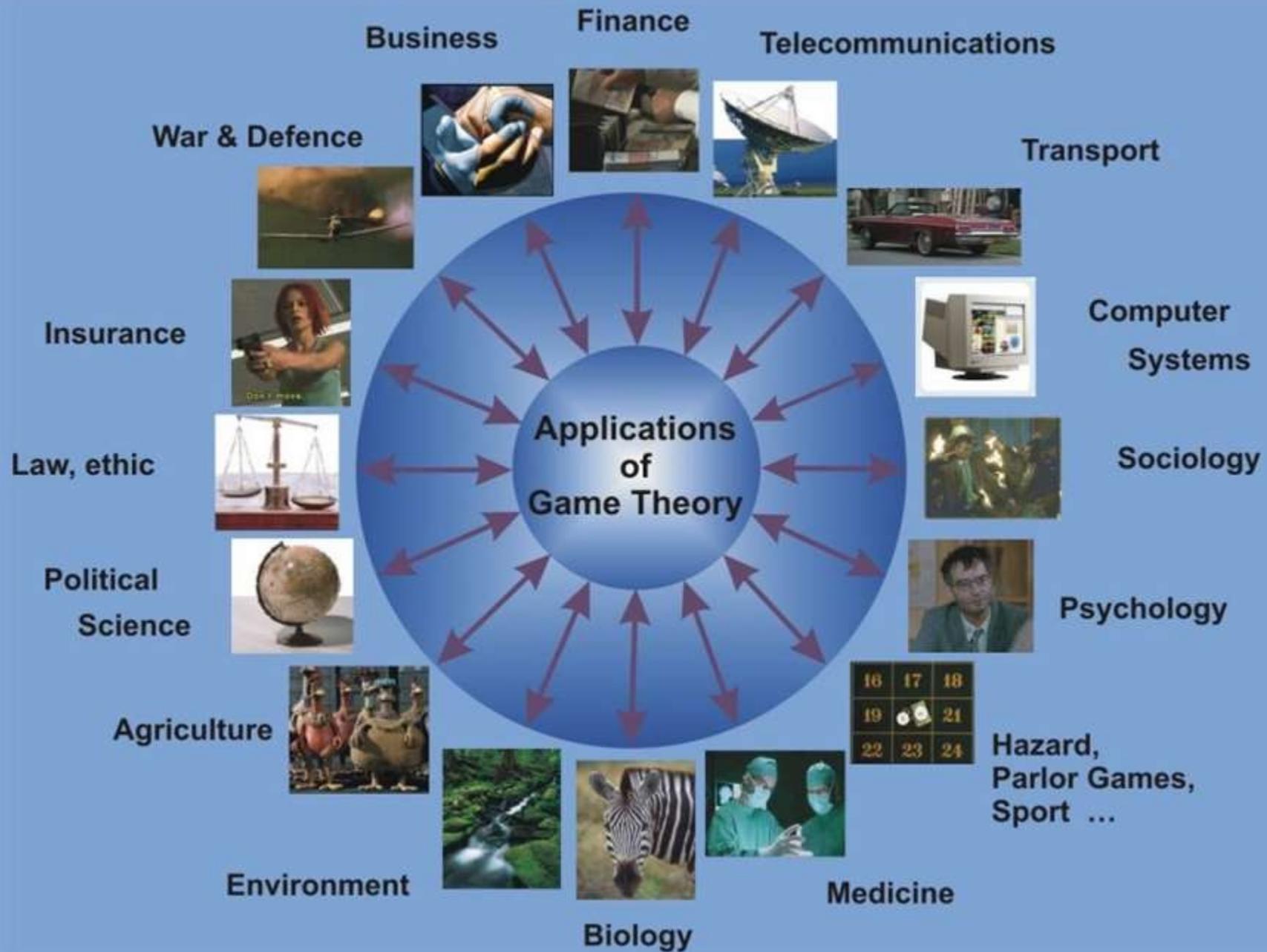
La Teoria dei Giochi

La **teoria dei giochi** è una branca della matematica utilizzata per modellare l'interazione strategica tra diversi giocatori in un contesto con regole e risultati predefiniti.

La teoria dei giochi può essere applicata in diversi ambiti dell'intelligenza artificiale:

- *sistemi di intelligenza artificiale multi-agente.*
- *apprendimento dell'imitazione e del rinforzo.*
- *addestramento sugli avversari nelle reti generative degli avversari (GAN).*

La teoria dei giochi può anche essere usata per descrivere molte situazioni nella nostra vita quotidiana e nei modelli di apprendimento automatico.

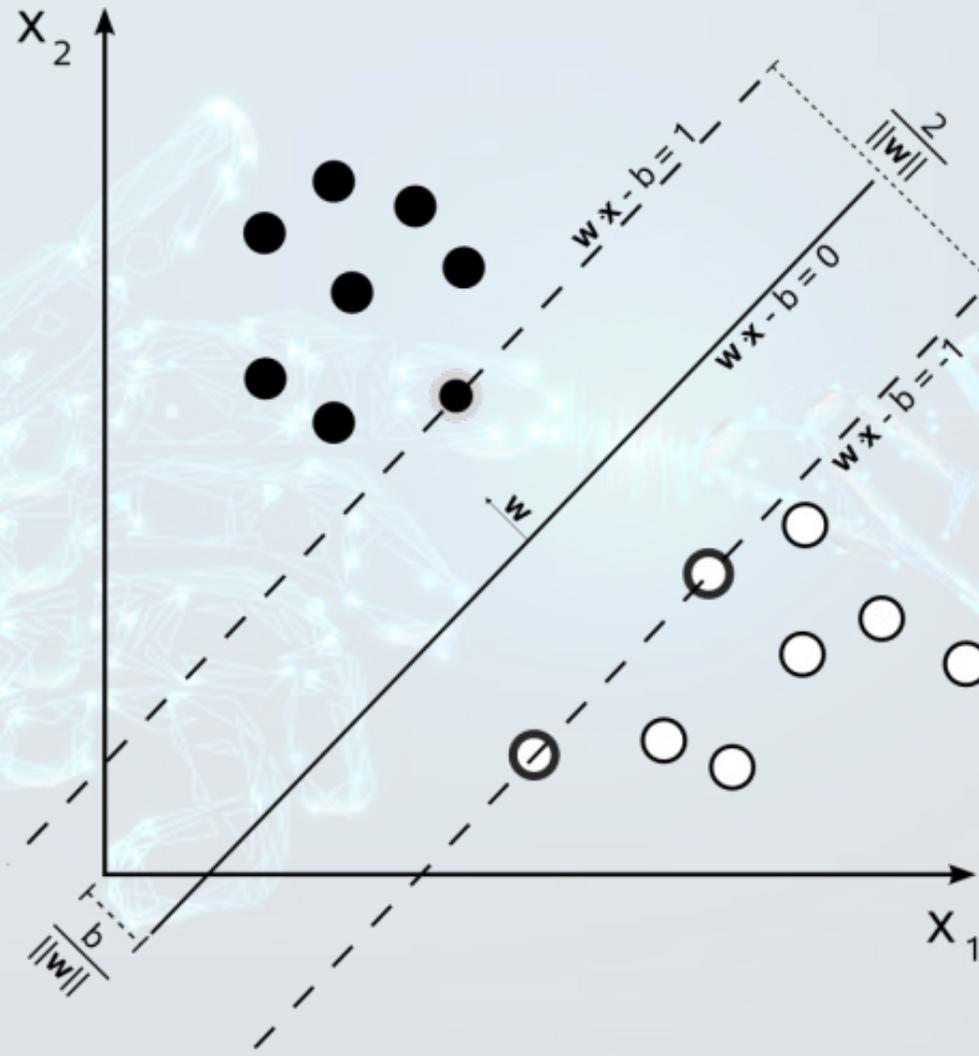


S.V.M. (Support Vector Machines)

Ad esempio, un algoritmo di classificazione come **SVM (Support Vector Machines)** può essere spiegato come se fosse un gioco a due attori in cui un giocatore sta sfidando l'altro a trovare il miglior iperpiano che gli dia i punti più difficili da classificare.

Il gioco converge in una soluzione che sarà un compromesso tra le abilità strategiche dei due giocatori (tanto più il primo giocatore sarà bravo a sfidare il secondo attore nel classificare punti dati difficili, tanto più il secondo giocatore dovrà essere abile nell'identificare i limiti della decisione migliore).

S.V.M. (Support Vector Machines)



Cos'è la teoria dei giochi

La teoria dei giochi può essere divisa in 5 tipi principali di giochi:

1) Giochi cooperativi contro giochi non cooperativi: nei giochi cooperativi i partecipanti possono stabilire alleanze al fine di massimizzare le loro possibilità di vincere la partita (ricorrendo ad esempio a delle trattative). Nei giochi non cooperativi i partecipanti non possono invece formare alleanze (come ad esempio nelle guerre).

2) Giochi simmetrici contro giochi asimmetrici: in una partita simmetrica tutti i partecipanti hanno gli stessi obiettivi e solo le loro strategie messe in atto per raggiungerli determineranno chi vince la partita (ne è un esempio classico il gioco degli Scacchi). Nei giochi asimmetrici invece, i partecipanti hanno obiettivi diversi o contrastanti.

3) Giochi di informazione perfetta contro giochi di informazione imperfetta: nei giochi di informazione perfetta tutti i giocatori possono vedere le mosse degli altri giocatori (come negli Scacchi). Invece, nei giochi con informazioni imperfette, le mosse degli altri giocatori sono nascoste (come ad esempio nei principali giochi di carte).

4) Giochi simultanei e giochi sequenziali: nei giochi simultanei i diversi giocatori possono intraprendere azioni contemporaneamente. Nei giochi sequenziali, al contrario, ogni giocatore è a conoscenza delle precedenti azioni degli altri giocatori (cosa che avviene per esempio nei comuni giochi da tavolo).

5) **Giochi a somma zero contro giochi a somma non zero:** nei giochi a somma zero un giocatore che guadagna qualcosa provoca una perdita agli altri giocatori. Nei giochi con somma diversa da zero, invece, più giocatori possono trarre vantaggio dai guadagni di un altro giocatore.

Diversi aspetti della teoria dei giochi sono comunemente usati nell'**Intelligenza Artificiale**, come ad esempio *Nash Equilibrium*, *Inverse Game Theory*. Nei prossimi paragrafi spiego il perché, fornendo anche qualche caso pratico.

I Giochi cooperativi

Nei Giochi Cooperativi (G.C.) si studia il formarsi di coalizioni con accordi sottoscritti e vincolanti che possono essere di vantaggio ai singoli componenti. Esiste una comunanza di interessi tra i giocatori, i quali perseguono un fine comune. In questo caso, i vari agenti economici coinvolti possono giungere ad un accordo coalizzante. L'obiettivo di fondo diventa il conseguimento del miglior risultato possibile per tutti i partecipanti all'accordo considerati.

Un gioco cooperativo a n persone è una coppia $G = \langle N, v \rangle$, dove $N = \{1, 2, \dots, n\}$ è un insieme finito con n elementi e $v : P(N) \rightarrow R$ è una funzione a valori reali definita su tutti i sottoinsiemi di N e tale che $v(\emptyset) = 0$.

Gli elementi di N rappresentano gli n giocatori.

La funzione caratteristica

Sia $\{N=1, \dots, n\}$ l'insieme dei giocatori di un gioco cooperativo. Si chiamano *coalizioni* (fra membri di N) tutti i 2^N sottoinsiemi di N . Si dice che il gioco è espresso in forma caratteristica v se ad ogni coalizione S è associato un numero reale $v(S)$ (la vincita), con la convinzione che alla coalizione vuota \emptyset è associata una vincita nulla: $v(\emptyset) = 0$.

Se in un gioco a n persone si costituiscono due coalizioni, il gioco diviene un gioco a due giocatori, in cui le strategie del primo sono l'unione delle strategie dei suoi componenti, analogamente per il secondo; le vincite del primo sono la somma delle vincite dei suoi componenti, analogamente per il secondo.

Potrebbe anche costituirsi una sola coalizione, in tal caso quella avversaria sarebbe la coalizione vuota.

La funzione caratteristica

Pertanto, si dice *funzione caratteristica* di un gioco ad n giocatori una funzione indicata con v (se il gioco è senza pagamenti laterali si usa V ed è più complessa) per cui si ha:

$$v : \wp(N) \rightarrow R \text{ con } v(\emptyset) = 0$$

- Se per ogni coppia di coalizioni disgiunte S e T si ha $v(S \cup T) = v(S) + v(T)$ la funzione v è detta *additiva*;
- Se si ha $v(S \cup T) \geq v(S) + v(T)$ la funzione v è detta *superadditiva*;
- Se si ha $v(S \cup T) \leq v(S) + v(T)$ la funzione v è detta *subadditiva*.

In altre parole v assegna ad S la massima vincita possibile indipendentemente dal comportamento degli altri giochi

La funzione caratteristica

In generale la funzione caratteristica è sufficiente a descrivere il gioco, per cui possono essere identificati. Un gioco descritto tramite la funzione caratteristica è detto *in forma caratteristica* o *coalizionale*. Se la funzione caratteristica è additiva o superadditiva o subadditiva anche il gioco è detto *additivo* o *superadditivo* o *subadditivo*. Se per ogni coalizione S si ha $v(S) + v(N \setminus S) = v(N)$ il gioco è detto *a somma costante*.

Nucleo

L'insieme di tutte le imputazioni dominate si chiama **nucleo** («core»).

Il nucleo è una generalizzazione per giochi n-persone. Esso ha certe caratteristiche di stabilità.

Il concetto di *Insiemi stabili* è costituito da tutte e sole le imputazioni non dominate tali che per ogni imputazione non appartenente allo stable set, ne esista almeno una dello stable set che la domina.

Caso Pratico

Si consideri una generica imputazione $(0.5, 1, 1.5)$.

Osserviamo che: $v(1, 2) = 2$, mentre $x_1 + x_2 = 1.5$.

Pertanto, la coalizione $(1,2)$ potrebbe assicurarsi una vincita superiore di 0.5 a quella che le viene assegnata da quella imputazione.

Il rimpianto $c(1, 2)$ della coalizione $(1, 2)$ rispetto all'imputazione $(0.5, 1, 1.5)$ è 0.5 . rispetto alla stessa imputazione, il rimpianto della coalizione $(1,3)$ è $c(1, 3) = v(1, 3) - x_1 - x_3 = -1$: in questo caso, non si tratta allora di rimpianto, ma di soddisfazione.

Caso Pratico

$$\begin{aligned}c(1, 2) &= 0.5 & c(1) &= c(2, 3) = -0.5 \\c(2) &= c(1, 3) = -1 & c(3) &= -1.5 .\end{aligned}$$

Pertanto, la coalizione che è peggio trattata da questa imputazione è la $(1, 2)$, con un rimpianto di 0.5. Scriviamo allora che $c_M(0.5, 1, 1.5) = 0.5$. Per simmetria, anche $c_M(1.5, 1, 0.5) = 0.5$. Se ripetiamo gli stessi calcoli rispetto all'imputazione $(3, 0, 0)$ otteniamo che il massimo rimpianto compete questa volta alla coalizione $(2, 3)$ ed è $c_M(3, 0, 0) = 2$. Quindi quest'ultima imputazione è meno preferibile della $(0.5, 1, 1.5)$, in quanto produce un danno maggiore ($c_M(3, 0, 0) > c_M(0.5, 1, 1.5)$).

Caso Pratico

È possibile verificare che, di tutte le possibili imputazioni, quella per la quale il massimo rimpianto c_M è minimo, è l'imputazione $(\frac{2}{3}, \frac{5}{3}, \frac{2}{3})$. Per tale imputazione c_M vale $-1/3$.

Il nucleolo è l'insieme delle imputazioni che minimizzano il massimo rimpianto.

Nel nostro esempio, tale insieme è composto da un unico elemento, ma ciò non vale in generale. È possibile dimostrare che se l'insieme dei vettori dei pagamenti è non vuoto, compatto e convesso (cioè nei casi più usuali), il nucleolo è non vuoto e consiste in un'unica imputazione.

Caso Pratico

Sia $N = \{1, \dots, n\}$ l'insieme dei giocatori di un gioco cooperativo fra n persone e sia $\mathcal{P}(N)$ l'insieme delle parti di N (*coalizioni*). Come è noto, tale insieme si compone di 2^n elementi, considerando anche la coalizione vuota \emptyset .

Supponiamo che sia possibile assegnare ad ogni coalizione $S \in \mathcal{P}(N)$ una vincita $v(S)$, con la condizione $v(\emptyset) = 0$. Si dirà allora che il gioco è espresso in *forma caratteristica* v . In particolare se avviene che:

$$v(S) + v(R) \leq v(S \cup R)$$

per tutte le $S, R \in \mathcal{P}(N)$ tali che $S \cap R = \emptyset$, si dirà che la funzione caratteristica del gioco è *superadditiva*.

Giochi con v -subadditiva

Esistono anche giochi con v *subadditiva*, ad esempio per la ripartizione di costi, e giochi con v qualsiasi, quando alcune collaborazioni non sono convenienti.

I giochi in cui tutte le disuguaglianze sono sostituite da uguaglianze si chiamano *inessenziali*; altrimenti, si chiamano *essenziali*.

Per tutte le funzioni superadditive è facile verificare che:

$$\sum_{i \in S} v(i) \leq v(S) \quad \text{per tutte le } S \in \mathcal{P}(N).$$

Un gioco in forma caratteristica si dice *a somma costante*

$$v(S) + v(N - S) = v(N) \quad \text{per tutte le } S \in \mathcal{P}(N).$$

Un'*imputazione* è un vettore (x_1, \dots, x_n) tale che

$$x_i \geq v(i) \quad \text{per tutti gli } i \text{ da } 1 \text{ a } n \text{ (razionalità individuale)}$$

$$\sum_{i=1}^n x_i = v(N) \quad \text{(razionalità di gruppo)}.$$

Si dice che l'imputazione x *domina* l'imputazione y tramite S se

$$x_i > y_i \quad \text{per tutti gli } i \in S$$

e

$$v(S) \geq \sum_{i \in S} x_i .$$

Si dice che x *domina* y se esiste almeno una coalizione S per cui x domina y tramite S .

Per le definizioni di *insiemi stabili*, *nucleo* e *nucleolo* rimandiamo alle pagine precedenti. In particolare, ricordiamo che ognuna di tali soluzioni può essere vuota, oppure può essere composta da più di un elemento.

Un importantissimo teorema che consente di costruire il core è il seguente.

L'imputazione $x = (x_1, \dots, x_n)$ appartiene al core se e solo se

$$\sum_{i \in S} x_i \geq v(S) \quad \text{per tutte le } S \in \mathcal{P}(N).$$

Caso Pratico: il core

Nel caso dell'esempio utilizzato in questo paragrafo, il core è dunque l'insieme delle imputazioni per cui vale:

$$\left\{ \begin{array}{l} x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \\ x_1 + x_2 \geq 2 \\ x_1 + x_3 \geq 1 \\ x_2 + x_3 \geq 2 \\ x_1 + x_2 + x_3 = 3 . \end{array} \right.$$

Caso Pratico: il Core

Ad esempio è facile vedere che il vettore $(0, 0, 3)$ è un'imputazione che non appartiene al core. Chi ha un po' di dimestichezza con la geometria dello spazio potrà verificare che tale insieme è costituito dall'involucro convesso dei punti $(0, 2, 1)$, $(1, 2, 0)$ e $(1, 1, 1)$, cioè dall'insieme delle imputazioni $(1 - a, 1 + a + b, 1 - b)$ che si ottengono facendo variare in tutti i modi possibili i parametri reali a e b nell'intervallo $[0, 1]$.

Vi sono giochi a nucleo vuoto. Ad esempio, è facile verificare che nel gioco a tre persone in cui le vincite dei singoli giocatori sono nulle, mentre tutte le altre valgono 1, il core è vuoto. Osserviamo, infatti, che deve valere:

Caso Pratico: il Core

$$\begin{cases} x_i \geq 0 & (i = 1, 2, 3) \\ x_1 + x_2 \geq 1 \\ x_1 + x_3 \geq 1 \\ x_2 + x_3 \geq 1 \\ x_1 + x_2 + x_3 = 1 . \end{cases}$$

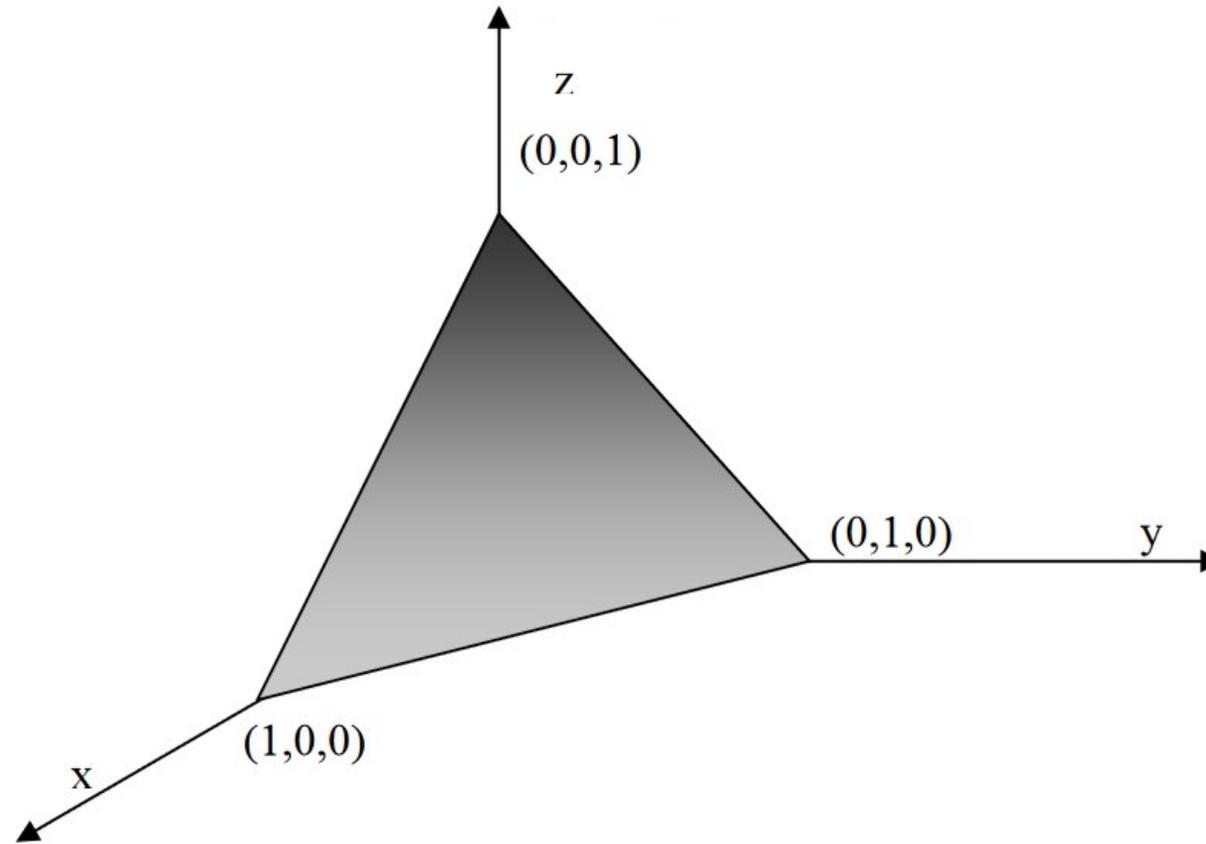
Sommando le tre disuguaglianze centrali, si ottiene:

$$2x_1 + 2x_2 + 2x_3 \geq 3$$

e questo è in contraddizione con l'ultima uguaglianza.

In generale, si può provare che ogni gioco essenziale a somma costante ha core vuoto.

Aspetto Qualitativo



Nash Equilibrium: l'equilibrio di Nash applicato all'Intelligenza Artificiale

L'equilibrio di Nash è una condizione in cui tutti i giocatori coinvolti nel gioco concordano sul fatto che non c'è soluzione migliore per il gioco se non quella della situazione reale in cui si trovano in quel dato punto/momento. In altre parole, nessuno dei giocatori avrebbe un vantaggio nel cambiare la propria strategia attuale (in base alle decisioni prese dagli altri giocatori).

Tornando all'esempio di prima (*l'algoritmo di classificazione SVM*), l'equilibrio di Nash si verifica quando il classificatore SVM concorda su quale iperpiano utilizzare per classificare i dati.

Equilibrio di Nash: il dilemma del prigioniero

Uno degli esempi più comuni usati per spiegare l'equilibrio di Nash è il *dilemma del prigioniero*.

		PLAYER B	
		COOPERATE	DEFECT
PLAYER A	COOPERATE	A: 1 year jail B: 1 year jail	A: 10 years jail B: 0 years jail
	DEFECT	A: 0 years jail B: 10 years jail	A: 5 years jail B: 5 years jail

Immaginiamo che due criminali vengano arrestati e tenuti in isolamento senza avere alcuna possibilità di comunicare tra loro:

- se **uno dei due prigionieri confesserà** che l'altro ha commesso un crimine, il primo verrà liberato mentre l'altro passerà 10 anni in prigione;
- se **nessuno dei due confessa**, trascorrono solo un anno in prigione per ciascuno;
- se **entrambi confessano**, invece trascorrono entrambi 5 anni in prigione.

In questo caso, *l'equilibrio di Nash* viene raggiunto quando entrambi i criminali si tradiscono.

Un modo semplice per scoprire se una partita ha raggiunto un equilibrio di Nash può essere quello di rivelare la tua strategia ai tuoi avversari.

		PLAYER B	
		COOPERATE	DEFECT
PLAYER A	COOPERATE	A: 1 year jail B: 1 year jail	A: 10 years jail B: 0 years jail
	DEFECT	A: 0 years jail B: 10 years jail	A: 5 years jail B: 5 years jail

Se dopo la tua rivelazione nessuno di loro cambia la propria strategia, l'equilibrio di Nash è dimostrato.

Sfortunatamente, un equilibrio di Nash è più facile da ottenere nei *giochi simmetrici* che asimmetrici. Un limite per poterlo applicare all'A.I.: i giochi asimmetrici sono infatti i più comuni nelle applicazioni del mondo reale e nell'intelligenza artificiale.

Inverse game Theory: la teoria dei giochi “inversa” importante per l’A.I.

La teoria dei giochi mira a comprendere le dinamiche di un gioco per ottimizzare il possibile risultato dei suoi giocatori.

L'***inverse game theory***, la cosiddetta teoria dei giochi “inversa”, mira invece a progettare un gioco basato sulle ***strategie*** e gli ***obiettivi*** dei giocatori.

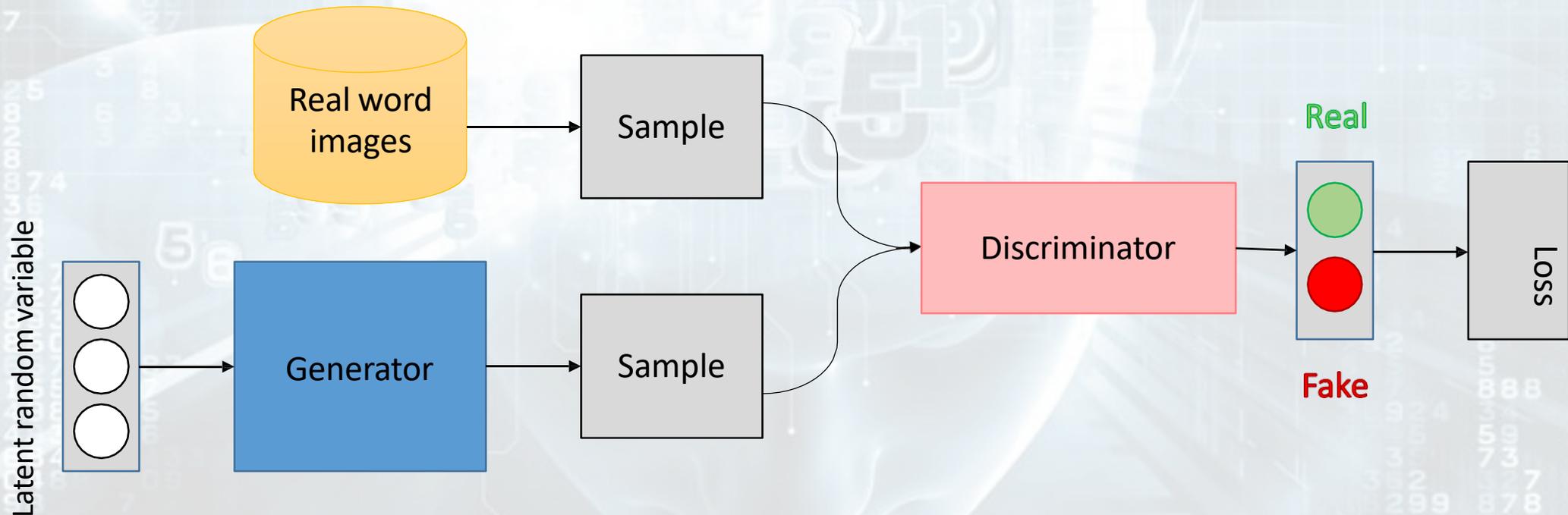
La teoria dei giochi inversi svolge un ruolo importante nella progettazione di ambienti degli agenti di Intelligenza Artificiale.



ESEMPI PRATICI DI TEORIA DEI GIOCHI
APPLICATA AD ALCUNI AMBITI
DELL'INTELLIGENZA ARTIFICIALE

Addestramento dell'avversario nella G.A.N. Generative Adversarial Network (rete generativa avversaria)

Una rete generativa avversaria è composta da due componenti: un **modello generativo**, o *generatore*, un **modello discriminativo**, o *discriminatore*, entrambi realizzati tramite reti neurali.



Modelli generativi e modelli discriminatori

I **modelli generativi** prendono come input alcune funzionalità, esaminano le loro distribuzioni e cercano di capire come sono state prodotte.

Alcuni esempi di modelli generativi sono Hidden Markov Models (HMMs) e Restricted Boltzmann Machines (RBMs).

I **modelli discriminatori** utilizzano invece le funzionalità di input per prevedere a quale classe potrebbe appartenere un campione. Support Vector Machines (SVM) è un esempio di modello discriminante.

Modello generativo nelle G.A.N.

Nelle **G.A.N.**, il modello generativo utilizza le funzionalità di input per creare nuovi campioni che mirano ad assomigliare abbastanza da vicino alle caratteristiche principali dei campioni originali.

I campioni generati vengono quindi passati con quelli originali al modello discriminatorio che deve riconoscere quali campioni sono autentici e quali sono falsi.

Un'applicazione di esempio di GAN può essere quella di **generare immagini** e quindi distinguere tra quelle *reali* e *false*.



Immagini generate da Nvidia GAN

Questo processo ricorda abbastanza da vicino le dinamiche di un gioco. In questo gioco, i nostri **giocatori** (i **due modelli**) si sfidano a vicenda.

Il primo crea campioni falsi per confondere l'altro, mentre il secondo giocatore cerca sempre meglio di identificare i campioni giusti.

Questo gioco viene quindi ripetuto in modo iterativo e in ogni iterazione, i parametri di apprendimento vengono aggiornati al fine di ridurre la perdita complessiva.

Questo processo continuerà fino al raggiungimento dell'**equilibrio di Nash** (i due modelli diventano competenti nell'esecuzione dei loro compiti e non sono più in grado di migliorare).

Multi-Agents Reinforcement Learning (MARL)

Il cosiddetto **Reinforcement Learning** (RL) mira a far apprendere un agente (il nostro “modello”) attraverso l’interazione con un *ambiente* (può essere virtuale o reale).

Il Reinforcement Learning è stato inizialmente sviluppato per aderire ai ***processi decisionali di Markov*** (framework matematico per la modellizzazione del processo decisionale in situazioni in cui i risultati sono in parte casuale e in parte sotto il controllo decisionale). In questo ambito, un agente viene posto in un ambiente stocastico stazionario e cerca di apprendere una politica attraverso un meccanismo di ricompensa/punizione. In questo scenario, è dimostrato che l’agente converge in una politica soddisfacente.

Tuttavia, se più agenti vengono inseriti nello stesso ambiente, questa condizione non è più vera. In effetti, nel primo scenario l'apprendimento dell'agente dipende solo dall'interazione tra l'agente e l'ambiente, nel secondo scenario (dove ci sono più agenti) l'apprendimento dipende anche dall'interazione tra agenti.

Immaginiamo che stiamo cercando di migliorare il **flusso del traffico** in una città usando un gruppo di auto a guida autonoma alimentate *dall'Intelligenza Artificiale*. Da sola, ciascuna delle auto può interagire perfettamente con l'ambiente esterno, ma le cose possono diventare più complicate se vogliamo *“far pensare”* le auto in gruppo. Ad esempio, un'auto potrebbe entrare in conflitto con un'altra perché per entrambi è più conveniente seguire un determinato percorso.

Questa situazione può essere facilmente modellata usando la teoria dei giochi. In questo caso, le nostre auto rappresentano i diversi giocatori e il punto di equilibrio tra la collaborazione delle diverse auto costituiscono l'equilibrio di Nash.

Modelli di apprendimento di rinforzo multi- agente con scenari di campo medio (MFS)

La modellazione di sistemi con un gran numero di agenti può diventare un compito davvero difficile. Questo perché, aumentando il numero di agenti, aumenta esponenzialmente il numero di possibili modi in cui i diversi agenti interagiscono tra loro.

In questi casi, la modellazione di ***modelli di apprendimento di rinforzo multi-agente con scenari di campo medio (MFS)*** potrebbe essere la soluzione migliore. Gli scenari sul campo medio possono, infatti, ridurre la complessità dei modelli MARL rendendo a priori il presupposto che tutti gli agenti abbiano funzioni di ricompensa simili.



Massimiliano Ferrara



0965 -1695446



massimiliano.ferrara@unirc.it



www.decisionslab.unirc.it

Grazie per l'Attenzione