

Introduzione alle tecniche di AI

Prof. Ing. Loris Penserini, PhD

<https://orcid.org/0009-0008-6157-0396>

Artificial Intelligence: cosa è...

L'intelligenza artificiale (AI) è la tecnologia informatica, formata da modelli logici e tecniche computazionali complesse, che consente di simulare i processi dell'intelligenza umana attraverso l'applicazione di algoritmi.

Uno degli obiettivi dell'AI è quello di creare computer in grado di pensare e agire come gli esseri umani.

Per realizzare questo obiettivo sono necessari tre componenti chiave:

- ▶ **Sistemi di calcolo**
- ▶ **Sistemi per la gestione dei dati**
- ▶ **Algoritmi e strutture dati avanzati (software)**

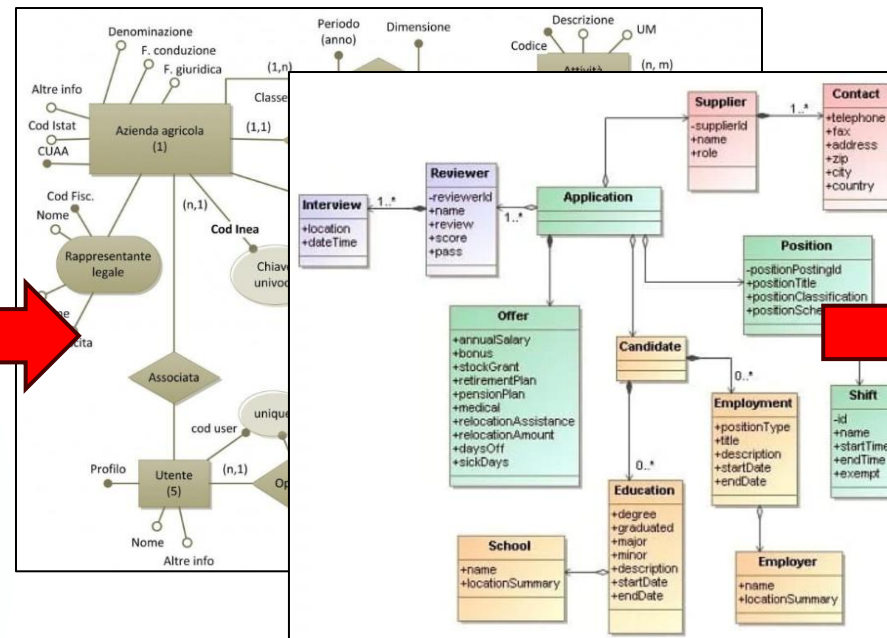
Artificial Intelligence: perché...

Per poter pensare e agire come gli esseri umani, dobbiamo creare **sistemi automatici per generare conoscenza** e...



Dati

Dati non
strutturati



Informazioni

Modelli per organizzare e
strutturare i Dati in informazioni

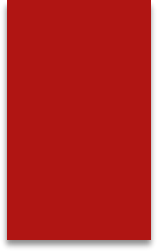


Conoscenza

Tecniche di IA per
correlare e visualizzare

La piramide di DIKW

[Ridi, AIB studi 2020]



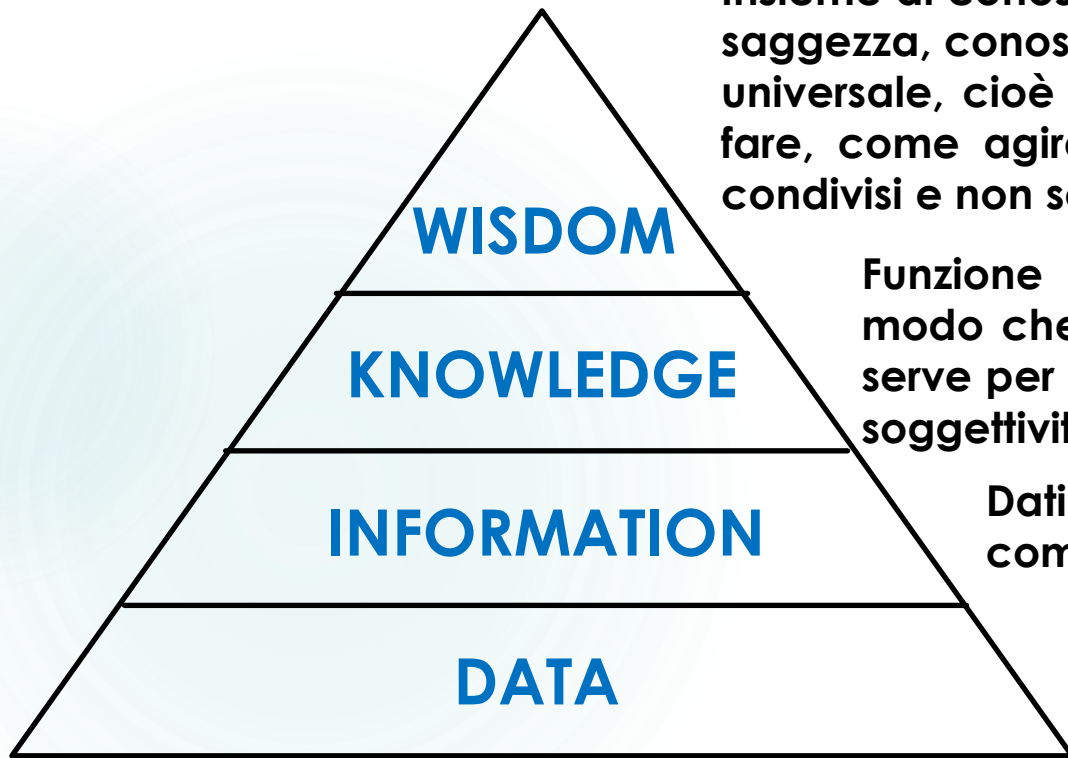
Per poter pensare e agire come gli esseri umani, dobbiamo creare sistemi automatici per generare conoscenza e... saggezza!

Insieme di conoscenze. Comprensione valutata, strato discreto tra conos. e saggezza, conos. integrata, tutto per conoscere il giusto: nel senso di scopo universale, cioè decisioni condivise. «Sapere il perché», «conoscere cosa fare, come agire» → si procede verso la capacità di esprimere giudizi condivisi e non soggettivi

Funzione dell'informazione elaborata, organizzata e strutturata, in modo che siano applicate come esperienze strutturate, intuizioni. E serve per valutare, incorporare nuove esperienze e informazioni (c'è soggettività)

Dati contestualizzati con significato e scopo, cioè utili per compiere azioni

Dati come simboli e segni che rappresentano stimoli e segnali, che non sono utili se non vengono contestualizzati. Per cui sono fatti e osservazioni oggettive disgiunte (fatto: tipico della logica proposizionale 0/1)



Informazione e Conoscenza

Nelle diverse presentazioni della piramide DIKW, la transizione da informazione a conoscenza avviene sempre quando si aggiunge uno di questi «ingredienti» o condizioni: *l'applicabilità, l'apprendimento accumulato, la comprensione, l'esperienza, l'inferenza, l'interiorizzazione, l'interpretazione, la riflessione, la rilevanza, la sintesi, la strutturazione cognitiva, la veridicità, ecc.*



al fine di

incrementare la qualità e l'utilità dell'informazione, rendendola solo adesso, appunto, vera e propria conoscenza

Conoscenza e Saggezza

Molti ricercatori nel presentare la piramide di DIKW evitano di parlare di «saggezza»... forse perché ha più a che fare con l'intuizione, la comprensione, l'interpretazione e altre azioni umane che con i sistemi informativi?

Saggezza come l'insieme di:

- ▶ *conoscenza-integrata, cioè superiore alla conoscenza circoscritta,*
- ▶ *informazioni e conoscenze estremamente utili e condivise, per saper individuare e applicare in modo oggettivo il concetto del giusto*

Artificial Intelligence

Nell'epoca moderna, il termine intelligenza artificiale è stato coniato nel 1955 da **John McCarthy**. Nel 1956, John McCarthy e altri scienziati organizzarono la conferenza "Dartmouth Summer Research Project on Artificial Intelligence".

Questo evento portò alla creazione del **machine learning**, del **deep learning**, dell'**analisi predittiva** e, ultimamente, dell'**analisi prescrittiva**. Ha inoltre dato origine a un campo di studio completamente nuovo, ovvero la scienza dei dati.

Considerazioni

Alcuni dei fattori di forza che hanno aumentato l'interesse ad utilizzare questo tipo di analisi dei dati:

- ▶ Big data in termini di volumi e tipologie di dati, e conseguente interesse delle aziende nel loro utilizzo per ottenere conoscenza di valore.
- ▶ Dispositivi elettronici sempre più veloci e *mobile*.
- ▶ Applicazioni alla portata di tutti.
- ▶ Un mercato del lavoro più complesso, dove il problema maggiore è riuscire a differenziarsi dalla concorrenza.

Quindi, analisti e manager aziendali stanno sfruttando questi strumenti di analisi.

Analisi Descrittiva

Le analisi descrittive sintetizzano o descrivono i dati grezzi e creano qualcosa che è interpretabile dagli esseri umani.

Nello specifico vengono analizzati gli eventi passati, dove per eventi passati ci si riferisce a qualsiasi punto del tempo in cui si è verificato un evento, che sia un minuto fa o un mese fa.

Le statistiche descrittive sono utili per mostrare per esempio, il totale dei prodotti presenti in magazzino o la spesa media per cliente. Le organizzazioni devono quindi utilizzare le analisi descrittive quando vogliono capire, a livello aggregato, cosa sta succedendo nella loro società.

Analisi Predittiva

Riguardano la comprensione del futuro. Questo tipo di analisi cercano di fornire alle organizzazioni informazioni basate su dati e stime circa la probabilità di un futuro risultato.

Tecniche di analisi predittiva esistono ormai da decenni, ma ora non sono più di esclusivo dominio della matematica e della statistica...

In ambito aziendale si è generata una forte consapevolezza che la mole di dati prodotta dall'utilizzo massivo della rete Internet rende evidente il notevole **vantaggio competitivo per le organizzazioni che stanno ricorrendo all'analisi predittiva per aumentare i loro profitti.**

Analisi Prescrittiva

In ambito aziendale sono quelle applicazioni che tentano di quantificare l'effetto delle future decisioni per consigliare i possibili risultati prima che esse siano effettivamente adottate.

L'analisi prescrittiva non solo prevede che cosa accadrà, ma spiega anche perché accadrà e fornisce le raccomandazioni in merito alle azioni che sfruttano queste previsioni.

Questo tipo di analisi sono complesse da amministrare e la maggior parte delle aziende non le utilizza ancora.



Costruire Macchine Intelligenti con Machine Learning

Creare Conoscenza in modo automatico

Nell'era dei **Big-Data**, si è evoluto l'apprendimento automatico, **Machine Learning**, come un sottocampo dell'Intelligenza Artificiale.

Per cui, sostituendo gli esseri umani nel difficile compito di definire regole e costruire modelli analizzando grandi quantità di dati, **il ML offre un'alternativa efficiente per estrarre conoscenza dai dati e dalle informazioni**; con il fine ultimo di migliorare gradualmente le prestazioni dei modelli predittivi e prendere decisioni «**utili e sagge**» per raggiungere obiettivi strategici (es. aziendali).

ML nella vita quotidiana

Il ML ricopre un ruolo sempre più importante della *Computer Science* grazie al suo valido impiego crescente nella vita quotidiana:

- ▶ Riconoscimento di voce e testo
- ▶ Efficaci filtri anti-spam nelle email
- ▶ Motori affidabili di ricerca Web
- ▶ Ottimi giocatori nei videogame e nei giochi da scacchiera
- ▶ Ecc.

ML nella vita quotidiana prossima...

Il ML è attualmente sperimentato anche in ambiti tanto più complessi quanto più strategici:

- ▶ Guida e volo autonomi
- ▶ Nel campo medico: modelli di *deep learning* per individuare tumori della pelle con accuratezza umana [Esteva et al., nature 2017]
- ▶ Modelli di *deep learning* per predire la struttura 3D delle proteine [DeepMind]
- ▶ Ecc.

Tipologie di Machine Learning

▶ Supervised Learning

- ▶ **Dati etichettati**
- ▶ **Feedback diretto**
- ▶ **Predizione del risultato/del futuro**

▶ Unsupervised Learning

- ▶ **Nessuna etichetta**
- ▶ **Nessun feedback**
- ▶ **Ricerca di strutture nascoste nei dati**

▶ Reinforcement Learning

- ▶ **Processo decisionale**
- ▶ **Sistema a ricompense (rewards)**
- ▶ **Apprendimento di sequenze di azioni**



La Storia delle RETI NEURALI (ANN)

Origini: Anni '40 - '50

Il modello di McCulloch e Pitts (1943): Warren McCulloch e Walter Pitts svilupparono uno dei primi modelli matematici di un neurone artificiale, basato sull'idea che il cervello umano potesse essere descritto in termini di reti di neuroni. Il loro modello, chiamato McCulloch-Pitts neuron, era una rappresentazione molto semplificata del funzionamento neurale, in grado di eseguire operazioni logiche fondamentali.

Hebbian learning (1949): Donald Hebb propose un principio chiave dell'apprendimento neurale, noto come "Legge di Hebb", secondo il quale le connessioni tra neuroni si rafforzano quando questi si attivano simultaneamente. Questo concetto ha influenzato lo sviluppo di modelli di apprendimento per le reti neurali.

Perceptron: Anni '50 - '60

Il Perceptron di Rosenblatt (1958): Frank Rosenblatt sviluppò il Perceptron, uno dei primi modelli di rete neurale che **poteva apprendere** a risolvere problemi semplici di classificazione. Il perceptron era un modello a strato singolo con neuroni che potevano apprendere pesi per le connessioni tramite un algoritmo di apprendimento supervisionato.

Limiti del Perceptron (1969): Nel loro libro "Perceptrons", Marvin Minsky e Seymour Papert dimostrarono matematicamente che i perceptron a strato singolo non erano in grado di risolvere problemi complessi come il problema XOR (una funzione logica che richiede non linearità). Questo portò a una battuta d'arresto nello sviluppo delle reti neurali, un periodo spesso definito **"inverno delle reti neurali"**.

Sviluppi successivi: Anni '70 - '80

Backpropagation (1986): Nel 1986, David Rumelhart, Geoffrey Hinton e Ronald Williams riscoprono e migliorarono l'algoritmo di backpropagation, che permise l'addestramento efficace delle reti neurali multi-strato (rete neurale feedforward con più strati nascosti). **Questo fu un progresso fondamentale, poiché risolveva il problema di apprendere con successo reti più complesse e non lineari.**

Hopfield network (1982): John Hopfield introdusse un nuovo tipo di rete neurale, chiamata rete di Hopfield, che poteva essere utilizzata come un modello di memoria associativa, in cui **le reti potevano ricordare schemi** sulla base di input incompleti o parziali.

Reti neurali ricorrenti (RNN): Nello stesso periodo, vennero sviluppati modelli di reti neurali ricorrenti (RNN), in cui i neuroni possono avere connessioni cicliche, permettendo la **modellazione di dati sequenziali e temporali.**

L'inverno delle reti neurali: Anni '90

Nonostante i progressi degli anni '80, le reti neurali caddero nuovamente in disgrazia negli anni '90, in parte a causa di una **mananza di potenza di calcolo** e difficoltà nel gestire reti molto grandi e complesse.

Durante questo periodo, altre tecniche di machine learning, come le macchine a vettori di supporto (SVM) e gli alberi decisionali, guadagnarono popolarità per la loro capacità di risolvere problemi pratici con successo.

La rinascita: Anni 2000 - 2010

Deep Learning: Con l'aumento della **potenza di calcolo** (grazie alle GPU) e la **disponibilità di grandi dataset**, le reti neurali profonde (ovvero reti con molti strati nascosti) iniziarono a ottenere risultati notevoli in vari campi, dando inizio a una vera e propria rinascita delle reti neurali, ora note come deep learning.

Convolutional Neural Networks (CNN): Yann LeCun, già negli anni '90, aveva sviluppato una particolare architettura neurale per il riconoscimento delle immagini chiamata Convolutional Neural Network (CNN), che è stata utilizzata con grande successo nell'analisi delle immagini. Nel 2012, la CNN di Alex Krizhevsky **vinse la competizione ImageNet**, dimostrando la potenza delle reti profonde.

Reti neurali ricorrenti (RNN) avanzate: Negli anni 2000, con l'introduzione delle varianti delle RNN come le LSTM (Long Short-Term Memory) di Sepp Hochreiter e Jürgen Schmidhuber (1997), le reti neurali ricorrenti divennero molto **efficaci nell'elaborazione di dati sequenziali**, come il linguaggio naturale (NLP) e le serie temporali.

Ascesa del Deep Learning: Anni 2010 - oggi

Reti generative (GAN): Nel 2014, Ian Goodfellow introdusse le Generative Adversarial Networks (GAN), un'architettura in cui due reti (una generativa e una discriminante) si sfidano per migliorarsi reciprocamente, portando a risultati notevoli nella generazione di immagini e altre forme di dati sintetici.

Transformers (2017): I Transformer, introdotti da Vaswani et al. nel 2017, hanno rivoluzionato il campo del Natural Language Processing (NLP) e delle reti neurali. **Il modello Transformer è alla base di modelli come GPT** (incluso GPT-3 e GPT-4) e BERT, che hanno raggiunto prestazioni all'avanguardia in molti compiti di comprensione e generazione del linguaggio naturale.

Apprendimento profondo in applicazioni reali: Il deep learning è diventato onnipresente in molte applicazioni moderne, tra cui la visione artificiale, il riconoscimento vocale, la traduzione automatica, la diagnostica medica, la guida autonoma e i sistemi di raccomandazione.

Recap...

- ▶ Tutti i modelli in discussione (Transformers, LLM, reti generative) sono reti neurali profonde. Cioè il Deep Learning fornisce l'infrastruttura (layers, backpropagation, ottimizzazione).
- ▶ Il Transformer è una rete neurale profonda specializzata per dati sequenziali (testo, audio, immagini). I modelli generativi basati su Transformer predicono il prossimo token, quindi generano contenuto (es. GPT = Generative Pretrained Transformer).
- ▶ Gli LLM usano i Transformer per generare testo (quindi sono modelli generativi). Sono pre-addestrati su enormi quantità di dati per generare contenuti coerenti, realistici e contestuali.

Prospettive future

Le reti neurali continuano a evolversi, con ricerche su architetture più efficienti e spiegabili, nuovi modelli generativi, **reti neurali quantistiche** e l'integrazione con altre tecnologie emergenti.

La sfida ora è migliorare ulteriormente le prestazioni, ridurre il consumo energetico dei modelli complessi e **affrontare questioni etiche come il bias e la trasparenza dei modelli**.

In sintesi, la storia delle reti neurali è un viaggio attraverso molteplici fasi di entusiasmo, stasi e rinascita, culminando **nell'era attuale del deep learning** che ha trasformato il modo in cui risolviamo problemi complessi utilizzando l'intelligenza artificiale.

Prospettive future: questioni etiche

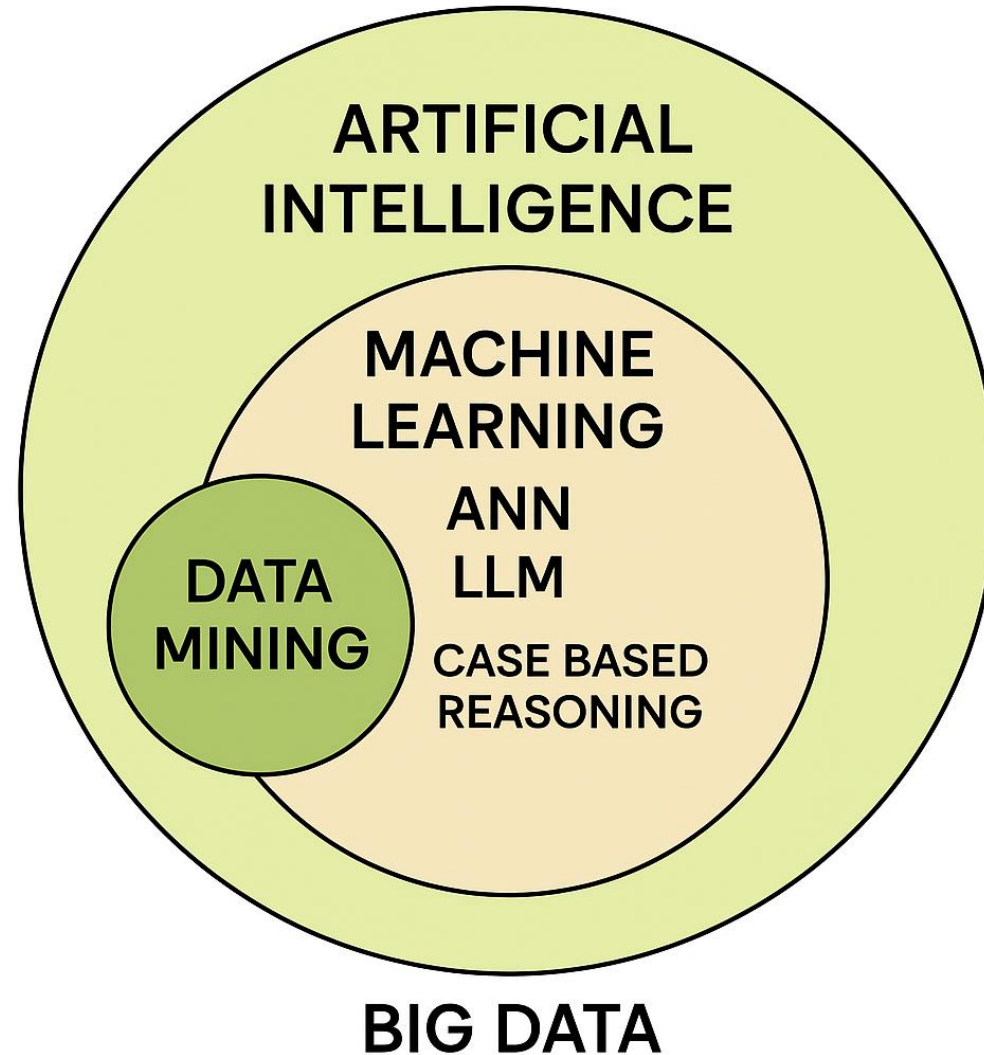
Nel contesto dell'intelligenza artificiale e del machine learning, il bias indica una distorsione o parzialità nei dati o negli algoritmi che porta a risultati ingiusti o discriminatori. Può avvenire, ad esempio, quando i dati di addestramento non rappresentano adeguatamente tutte le categorie della popolazione (per genere, etnia, età, ecc.), causando decisioni che riflettono questi pregiudizi. Ecco alcuni esempi di bias e questioni etiche:

Discriminazione: Un algoritmo di selezione del personale che, a causa di dati storici, favorisce inconsapevolmente i candidati di un certo genere o etnia.

Mancaza di trasparenza: Le decisioni prese da algoritmi complessi, come le reti neurali, possono essere difficili da comprendere e spiegare, il che rende problematica la loro applicazione in ambiti critici, come la giustizia o la sanità.

Responsabilità: Quando un'IA commette un errore o causa un danno, può essere difficile attribuire responsabilità, poiché l'algoritmo potrebbe aver appreso il suo comportamento da dati distorti.

Relazione tra le tecniche di AI



Machine Learning: il modello delle RETI NEURALI (ANN)

Reti neurali e Machine Learning

Le reti neurali artificiali (ANN) sono uno dei modelli più potenti all'interno del campo del machine learning (apprendimento automatico)

Le ANN costituiscono il fondamento delle tecniche di deep learning (apprendimento profondo).

Il legame tra le reti neurali e il machine learning è profondo, poiché le reti neurali sono una delle architetture più efficaci per apprendere modelli complessi da grandi quantità di dati.

Reti neurali

In AI si parla di **classi di algoritmi** perché ne esistono davvero tante e i ricercatori le classificano in base alle loro caratteristiche tecniche. E' un dato di fatto che la classe di algoritmi delle «**reti neurali**» ha fortemente contribuito alla rivoluzione dell'ultimo decennio dell'IA.

L'aggettivo «neurale» deriva da un modello matematico, sviluppato all'inizio del Novecento, che mirava a descrivere il funzionamento del neurone; che poi risultò errato!

Una rete neurale è un insieme di neuroni collegati tra loro secondo determinate geometrie. I neuroni di cui parliamo, sono funzioni matematiche, molto lontane dai neuroni che troviamo nel cervello.

Reti neurali e ML

In **machine learning**, l'obiettivo è costruire modelli che possano imparare automaticamente dai dati per fare previsioni o prendere decisioni senza essere programmati esplicitamente.

Le **reti neurali** si inseriscono in questo contesto come un'architettura flessibile e potente per apprendere dai dati.

Una rete neurale è composta da neuroni (unità di calcolo) organizzati in strati:

- ▶ **Strato di input:** Riceve i dati di input (features).
- ▶ **Strati nascosti:** Effettuano le trasformazioni e l'elaborazione dei dati.
- ▶ **Strato di output:** Produce la previsione o classificazione finale.

Ogni neurone riceve degli input, li combina con pesi associati e passa il risultato attraverso una funzione di attivazione non lineare per generare un output. Il processo di apprendimento delle reti neurali consiste nel trovare i valori ottimali per questi pesi, in modo che la rete possa fare previsioni accurate.

Reti neurali e il «perceptron»

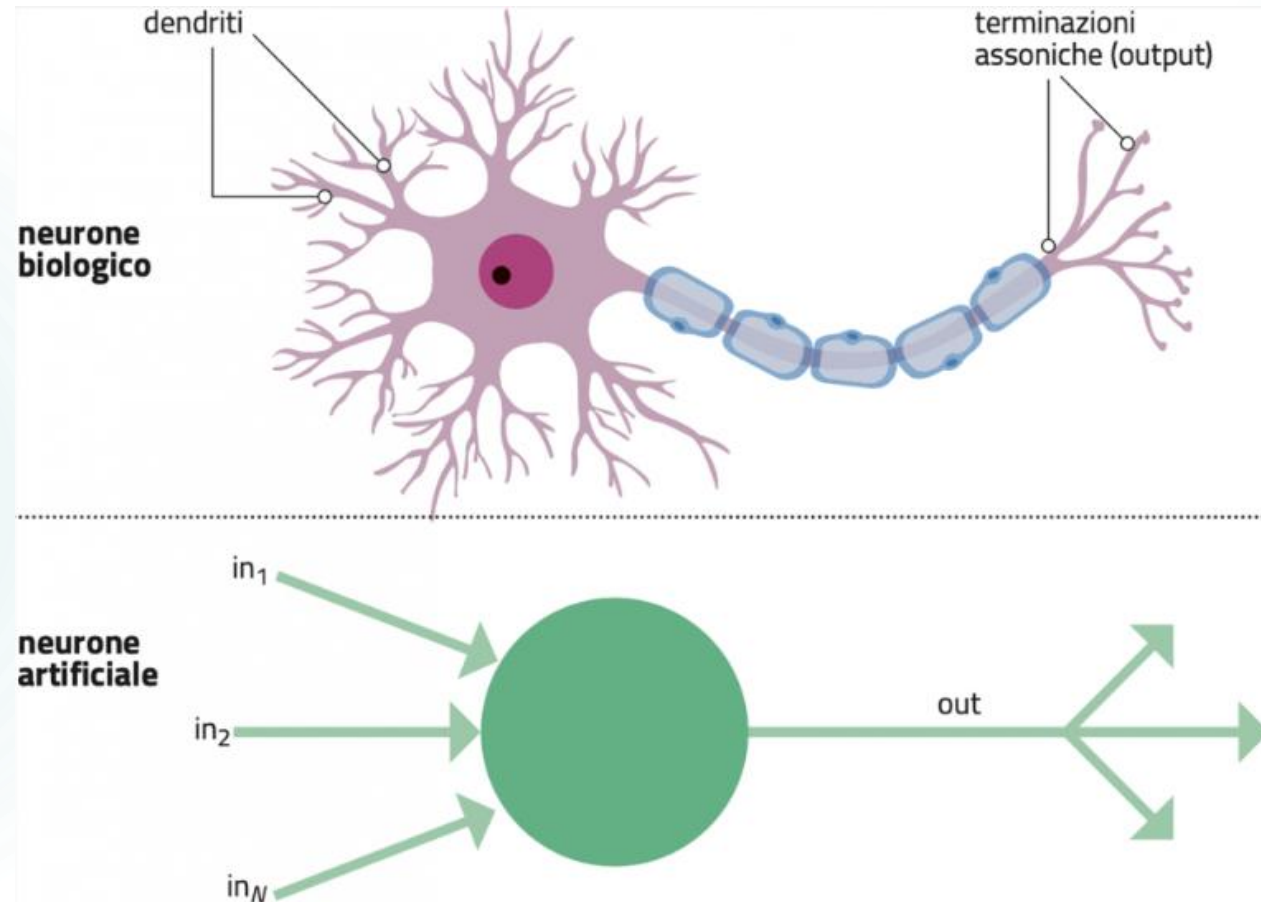
L'architettura più semplice di «rete neurale» è quella del «perceptrone» che è composta da un singolo neurone.

Il perceptron è un algoritmo di classificazione binaria di apprendimento supervisionato, originariamente sviluppato da Frank Rosenblatt nel 1957.

Classifica i dati di input in uno dei due stati, separati sulla base di una procedura di addestramento eseguita su dati di input precedenti.

Ispirazione e analogie

Modelli biologico e artificiale a confronto...



Machine Learning

Tipologie di ML

Le tipologie di Machine Learning si dividono principalmente in quattro categorie, in base al tipo di dati disponibili e all'obiettivo dell'apprendimento:

- ▶ Apprendimento Supervisionato (Supervised Learning)
- ▶ Apprendimento Non Supervisionato (Unsupervised Learning)
- ▶ Apprendimento per Rinforzo (Reinforcement Learning)
- ▶ Apprendimento Auto-Supervisionato (Self-Supervised Learning)

Supervised Learning

Caratteristiche:

- ▶ I dati di addestramento sono etichettati: ogni input ha un output conosciuto.
- ▶ L'obiettivo è predire un'etichetta per nuovi dati.

Esempi:

- ▶ **Classificazione**: email spam / non spam, diagnosi malattie
- ▶ **Regressione**: previsione dei prezzi di case, stime di vendita

Regressione

Le tecniche di regressione sono un sottoinsieme del ML supervisionato e vengono utilizzate per predire un valore continuo (ad esempio: prezzo, temperatura, tasso di crescita, ecc.) in base a uno o più input.

Modella la relazione tra variabili indipendenti (input) e una variabile dipendente (output continuo).

Alcune tecniche di regressione:

- ▶ Lineare
- ▶ Polinomiale
- ▶ Decision Tree
- ▶ con Reti Neurali (ANN)
- ▶ Ecc.

Proprietà

Alcune Tecniche di Regressione

Tecnica	Non linearità	Overfitting rischio	Interpretabile	Complessità
Regressione Lineare	NO	Basso	Alta	Bassa
Polinomiale	SI	Medio/Alto	SI	Media
Alberi / Random Forest	SI	Medio (RF: basso)	Alberi	Media/Alta
Reti Neurali	SI	Alto	Bassa	Alta

Overfitting

Il rischio di overfitting è un concetto centrale nel Machine Learning e significa che un modello ha imparato troppo bene i dati di addestramento, al punto da memorizzarli anziché generalizzare.

L'overfitting si verifica quando un modello si adatta troppo ai dati di training, catturando rumore, anomalie o dettagli specifici che non si ripetono nei dati nuovi (di test o reali).

Regressione Lineare

La regressione lineare è una delle tecniche fondamentali e più semplici nel machine learning supervisionato, usata per prevedere un valore continuo a partire da una o più variabili di input. Si basa sulla seguente formula:

$$y = \beta_0 + \beta_1 \cdot x + \varepsilon$$

$$\text{Errore}_i = y_i - \hat{y}_i$$

y : variabile dipendente (target)

x : variabile indipendente (input)

β_0 : intercetta (dove la retta incrocia l'asse y)

β_1 : coefficiente angolare (quanto cambia y per ogni unità di x)

ε : errore (distanza tra valore previsto \hat{y}_i e reale y_i)

Le metriche più comuni

Metriche più comuni per il calcolo dell'Errore Totale		
Nome	Formula	Descrizione
SSE (Sum of Squared Errors)	$\sum_{i=1}^n (y_i - \hat{y}_i)^2$	Errore totale non negativo. Penalizza fortemente errori grandi.
MSE (Mean Squared Error)	$\frac{1}{n} \cdot \sum_{i=1}^n (y_i - \hat{y}_i)^2$	SSE diviso per il numero di punti.
RMSE (Root Mean Squared Error)	$\sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (y_i - \hat{y}_i)^2}$	MSE reso interpretabile nella stessa unità di y.

Obiettivo del modello

Trovare i valori di β_0 e β_1 che minimizzano l'errore tra i valori previsti e quelli osservati, spesso usando il metodo dei minimi quadrati:

$$\textit{Errore totale} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Tipi di Regressione Lineare	
Tipo	Descrizione
Semplice	1 sola variabile indipendente x
Multipla	2 o più variabili indipendenti x_1, x_2, \dots, x_n
Polinomiale	Le variabili x appaiono con potenze (es. x^2)
Ridge / Lasso / ElasticNet	Regressioni regolarizzate per ridurre overfitting

Project Work: individuare le Proprietà

Individuare le proprietà del Supervised Learning se applicato nei seguenti scenari

Supervised Learning		
Scenario	Supervisionato?	Spiegazione
Prevedere il prezzo di una casa		
Riconoscere se un'email è spam		
Segmentare clienti in gruppi di acquisto		
Ridurre dimensionalità per visualizzare dati		
Riconoscere cifre scritte a mano		
Rilevare anomalie in traffico di rete		

Proprietà del Supervised Learning

Supervised Learning		
Scenario	Supervisionato?	Spiegazione
Prevedere il prezzo di una casa	SI	Input: caratteristiche casa → Output: prezzo
Riconoscere se un'email è spam	SI	Etichettato: spam / non-spam
Segmentare clienti in gruppi di acquisto	NO	Non si sa in anticipo quanti gruppi esistano
Ridurre dimensionalità per visualizzare dati	NO	Nessun output noto da predire
Riconoscere cifre scritte a mano	SI	Input: immagine → Output: cifra (0–9)
Rilevare anomalie in traffico di rete	NO	Il sistema cerca comportamenti "diversi"

Project Work: prezzo case

Scrivere un algoritmo in Python che, assegnati dei dati reali (addestramento), sia poi in grado di «prevedere» il prezzo di una casa in base alla sua metratura fornita in input.

Dati Reali	
Metratura (x)	Prezzo (y)
50	100.00 €
80	150.00 €
100	200.00 €

Calcolare i coefficienti β_0 : **intercetta** e β_1 : **coefficiente angolare**

Calcolare il prezzo previsto per una casa di 120 m²

PW: prezzo case

Aiuto:

Utilizzare le seguenti librerie in Python:

```
from sklearn.linear_model import LinearRegression  
import numpy as np
```

La regressione cercherà una funzione del tipo:

$$\text{Prezzo} = \beta_0 + \beta_1 \text{ Metratura}$$

Utilizzare il compilatore online:

<https://www.mycompiler.io/it/new/python>

PW: prezzo case – Soluz.1

Python ▾



```
1 from sklearn.linear_model import LinearRegression
2 import numpy as np
3
4 # Dati di esempio
5 X = np.array([[50], [80], [100]]) # Input: metrature
6 y = np.array([100000, 150000, 200000]) # Output: prezzi
7
8 # Modello
9 model = LinearRegression()
10 model.fit(X, y)
11
12 # Coefficienti
13 print("Intercetta:", model.intercept_)
14 print("Coefficiente:", model.coef_)
15
16 # Previsione
17 print("Prezzo previsto per 120 m²:", model.predict([[120]]))
```

Output:

Intercetta (β_0): -1315.789

Coefficiente (β_1): 1973.684

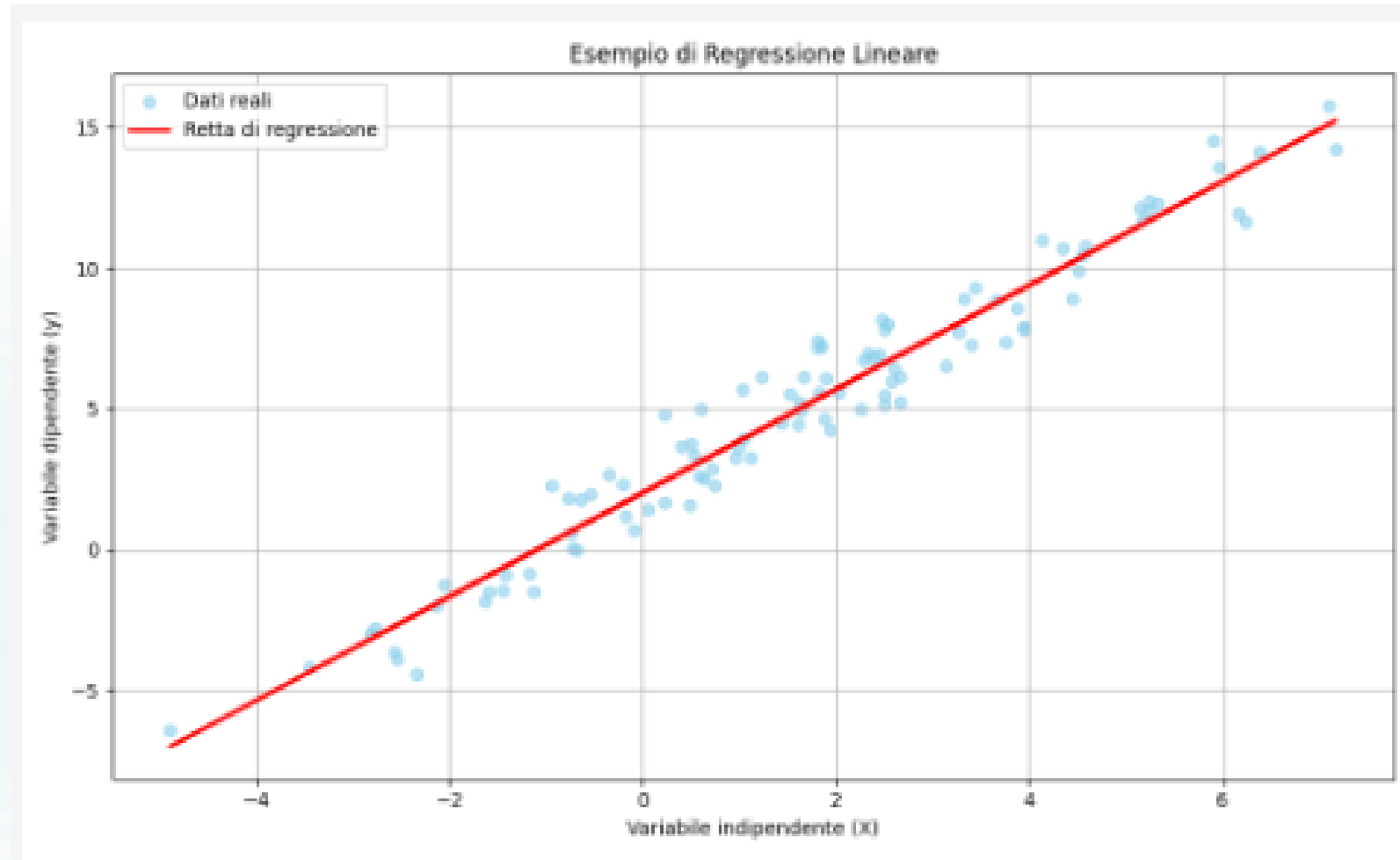
Prezzo previsto per 120 m²:

235.526,3 Euro

PW: prezzo case – Soluz.2: codice

```
reg_lin_grafico.py
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.linear_model import LinearRegression
4
5 # 1. Generazione dei dati (simulati)
6 np.random.seed(0)
7 X = 2.5 * np.random.randn(100, 1) + 1.5 # Variabile indipendente
8 y = 2 + 1.8 * X + np.random.randn(100, 1) # Variabile dipendente con rumore
9
10 # 2. Addestramento del modello
11 model = LinearRegression()
12 model.fit(X, y)
13 y_pred = model.predict(X)
14
15 # 3. Visualizzazione dei dati e della retta di regressione
16 plt.figure(figsize=(10, 6))
17 plt.scatter(X, y, color='skyblue', label='Dati reali', alpha=0.6) # Punti originali
18 plt.plot(X, y_pred, color='red', label='Retta di regressione', linewidth=2) # Retta
19 plt.title("Esempio di Regressione Lineare")
20 plt.xlabel("Variabile indipendente (X)")
21 plt.ylabel("Variabile dipendente (y)")
22 plt.legend()
23 plt.grid(True)
24 plt.tight_layout()
25 plt.show()
```

PW: prezzo case – Soluz.2: grafico



Unsupervised Learning

Caratteristiche:

- ▶ I dati non sono etichettati.
- ▶ L'algoritmo cerca pattern, gruppi, strutture nei dati.

Esempi:

- ▶ **Clustering:** segmentazione clienti
- ▶ **Associazione:** regole di acquisto ("Market Basket Analysis")
- ▶ **Riduzione dimensionalità:** compressione e visualizzazione dati

Supervised vs Unsupervised Learning

Supervised vs Unsupervised Learning		
Caratteristica	Apprendimento Supervisionato	Apprendimento Non Supervisionato
Dati di input	Etichettati (ogni esempio ha un output noto)	Non etichettati (nessun output fornito)
Obiettivo	Prevedere un output noto (Y) da un input (X)	Scoprire pattern nascosti nei dati
Tipo di problema	Classificazione, regressione	Clustering, riduzione dimensionale
Feedback	È guidato: il modello è “corretto” durante il training	Nessun feedback esplicito su cosa sia “giusto”
Esempi noti	Sì ($X \rightarrow Y$)	Solo X, nessuna Y

Reinforcement Learning

Caratteristiche:

- ▶ Un agente interagisce con un ambiente, compie azioni e riceve ricompense o penalità.
- ▶ Impara a massimizzare la ricompensa cumulativa.

Esempi:

- ▶ Giochi da scacchiera
- ▶ Robotica
- ▶ Sistemi di raccomandazione adattivi

Self-Supervised Learning

Caratteristiche:

- ▶ I dati vengono etichettati automaticamente a partire dagli stessi dati grezzi.
- ▶ È molto usato nel deep learning e nei modelli di linguaggio (come GPT).

Esempi:

- ▶ Completamento di testo
- ▶ Pretraining di modelli linguistici o visivi

Self-Supervised Learning

Self-Supervised Learning	
Aspetto	Descrizione
Etichette umane	Non necessarie
Tipo di apprendimento	Autogenerato dai dati
Utilizzo principale	Pretraining, rappresentazioni generali
Ambiti	NLP, Visione, Audio, Biologia, ecc.

Bibliografia

[Esteva et al., nature 2017] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau & Sebastian Thrun, "Dermatologist-level classification of skin cancer with deep neural networks", Nature 542, 115–118 (2017). <https://doi.org/10.1038/nature21056>

[Panti et al., CoopIS-01] Maurizio Panti, Luca Spalazzi, Loris Penserini, "Cooperation Strategies for Information Integration", in Proc. of Sixth International Conference on Cooperative Information Systems (CoopIS 2001), Springer Verlag, LNCS 2172, Trento, Italy, September 5-7, 2001.

[Panti et al., IJCAI-01] Maurizio Panti, Luca Spalazzi, Loris Penserini, "A Distributed Case-Based Query Rewriting", in Proc. of 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Morgan Kaufmann Publishers, vol.2, p.1005-1010, Seattle, Washington, USA, August 4-10, 2001.

[Ridi, AIB studi 2020] Riccardo Ridi, "La piramide dell'informazione: una introduzione", journal AIB studi, n. vol. 59/1-2, p.69-96, 2020.
<https://dx.doi.org/10.2426/aibstudi-12216>

[Raschka et al., 2020] Sebastian Raschka, Vahid Mirjalili, Machine Learning con Python – Costruire algoritmi per generare conoscenza. Apogeo, 2020. ISBN 978-88-503-3524-4

[DeepMind] AlphaFold reveals the structure of the protein universe. <https://deepmind.google/discover/blog/alphafold-reveals-the-structure-of-the-protein-universe/>