Dai Neuroni alle Macchine che Apprendono: Un viaggio nell'Intelligenza Artificiale

Filippo Petroni

Professore di Matematica Applicata, Università di Chieti-Pescara,
Università LUISS Guido Carli

27 febbraio 2025

Il Cervello: Una Rete di Neuroni

Immagina il cervello come una **enorme rete** di cavi elettrici tutti specializzati: questi cavi sono i **neuroni**, e ognuno di noi ne ha circa **86 miliardi**!

- I neuroni si connettono tra loro attraverso delle **sinapsi** per trasmettere informazioni.
- Durante il processo di apprendimento, alcune connessioni si **rafforzano** e altre si **indeboliscono** o vengono **eliminate**.
- Questo fenomeno si chiama **plasticità sinaptica**, ed è la base della nostra capacità di apprendere.

Il Processo di Apprendimento Biologico

Come il cervello decide quali connessioni rafforzare?
La regola di Hebb:

Principio chiave

Neuroni che si attivano insieme, si connettono insieme!

Espressione matematica:

$$\Delta w_{ij} = \eta x_i x_j \tag{1}$$

- w_{ij} è la **forza della connessione** tra il neurone i e il neurone j.
- η è il **learning rate**, ovvero la velocità con cui il cervello modifica le connessioni.
- x_i e x_j rappresentano l'attività dei due neuroni.

Esempio: Apprendere a Suonare la Chitarra

- All'inizio, è difficile coordinare le dita.
- Più suoni, più si rafforzano i collegamenti tra i neuroni responsabili delle **mani** e della **memoria muscolare**.
- Dopo pratica sufficiente, suonare diventa automatico, perché il cervello ha ottimizzato le connessioni giuste!

Come è nata l'Intelligenza Artificiale?

Gli scienziati si sono chiesti:

Domanda chiave

Se il cervello umano apprende modificando le connessioni tra neuroni, possiamo costruire un modello matematico che faccia lo stesso?

McCulloch e Pitts (1943)

- Hanno sviluppato il primo modello matematico di **neurone artificiale**.
- Questo lavoro ha gettato le basi per le moderne **reti neurali artificiali**.

Il Neurone Artificiale: Come Funziona?

Un neurone artificiale è una semplificazione matematica di un neurone biologico.
Formula matematica:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2}$$

- x_i sono gli **input** (es. pixel di un'immagine, parole di una frase).
- w_i sono i **pesi sinaptici** (indicano l'importanza di ogni input).
- b è il **bias**, un valore che aiuta a regolare l'output.
- $f(x)$ è la **funzione di attivazione**, che decide se il neurone si accende.

Esempio: Un Neurone come un Professore che Corregge Compiti

Immagina un professore che assegna voti a un compito.
Considera diversi aspetti:

- **Correttezza dei risultati** ($w_1 = 0.7$)
- **Chiarezza della spiegazione** ($w_2 = 0.2$)
- **Ordinato e leggibile** ($w_3 = 0.1$)

Il professore somma questi aspetti e decide un voto finale.
Un **neurone artificiale** funziona allo stesso modo: prende più input, li pesa e calcola un output.

Il Neurone Decide se "Accendersi"

Le Funzioni di Attivazione

Un neurone artificiale non restituisce un numero qualsiasi, ma prima passa il risultato attraverso una **funzione di attivazione**, che decide se il neurone deve essere attivato o no.

Funzione Sigmoide

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

- Serve per problemi in cui l'output deve essere compreso tra 0 e 1 (es. probabilità).
- **Esempio:** Un modello che deve decidere se un'email è spam o no.

Funzione ReLU (Rectified Linear Unit)

$$f(x) = \max(0, x) \tag{4}$$

- Se il valore è negativo, il neurone **non si accende** (restituisce 0).
- Se il valore è positivo, il neurone **si attiva normalmente**.
- **Esempio:** Un'IA che riconosce immagini, accendendo i neuroni solo quando trova caratteristiche importanti.

Funzione Tangente Iperbolica (Tanh)

$$f(x) = \tanh(x) \tag{5}$$

- Simile alla sigmoide, ma restituisce valori tra -1 e 1.
- **Utile per dati con valori positivi e negativi**.

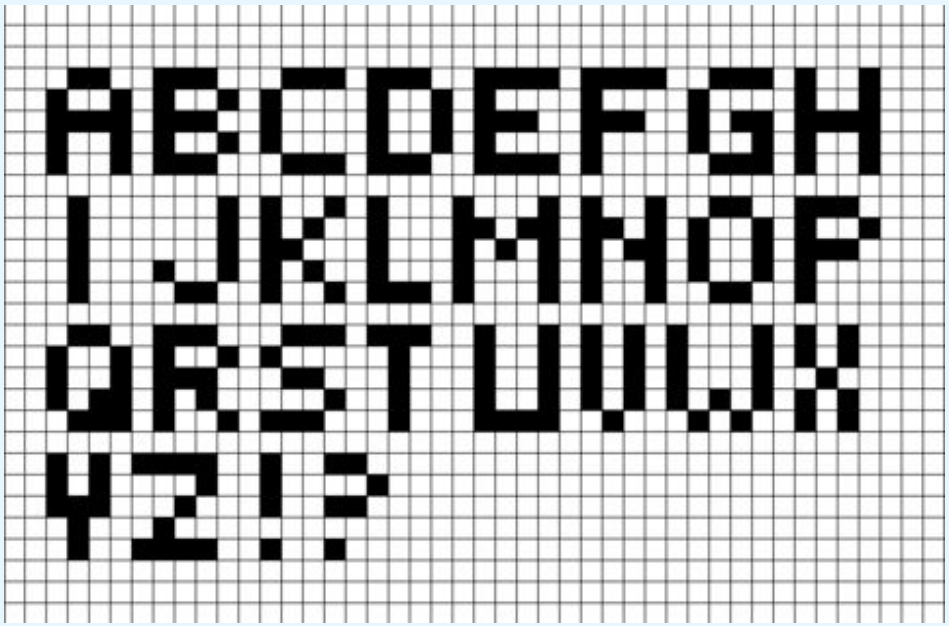
Costruire un'IA per Riconoscere Lettere

Supponiamo di voler costruire un'IA che **riconosca le lettere dell'alfabeto**.

1. Il computer trasforma l'immagine di una lettera in numeri (es. i pixel di una "A").
2. I numeri vengono inviati come **input** alla rete neurale.
3. Ogni neurone calcola l'importanza dei diversi pixel e decide se attivarsi.
4. Il risultato passa attraverso una funzione di attivazione.
5. Se il neurone "A" è il più attivo, la rete riconosce la lettera!

Conclusione

- Il processo è simile a **come il nostro cervello impara a leggere**: iniziamo a riconoscere lettere, poi parole, poi intere frasi.



Punti Chiave

- Il cervello impara rafforzando o indebolendo le connessioni tra neuroni (plasticità sinaptica).
- La regola di Hebb spiega perché i neuroni che si attivano insieme, si connettono insieme.
- I neuroni artificiali imitano quelli biologici, pesando gli input e usando funzioni di attivazione.
- Le reti neurali possono imparare a riconoscere immagini, parole e suoni proprio come il cervello umano.

Immagine di un Neurone Artificiale

Esempio di codice Python per un neurone artificiale:

```python
import numpy as np
class Neurone:
    def __init__(self, input_size):
        self.pesi = np.random.rand(input_size)
        self.bias = np.random.rand(1)
    def attivazione_relu(self, x):
        return np.maximum(0, x)
    def output(self, input_data):
        z = np.dot(input_data, self.pesi) + self.bias
        return self.attivazione_relu(z)
# Esempio di utilizzo
input_data = np.array([1.0, 2.0, 3.0])
neurone = Neurone(input_size=3)
output = neurone.output(input_data)
print("Output del neurone:", output)
```

Le Reti Neurali Artificiali

Le reti neurali artificiali sono ispirate al cervello umano: sono formate da **neuroni artificiali** connessi tra loro, che imparano regolando la forza delle loro connessioni, proprio come fa il nostro cervello con le sinapsi.
Come fanno a imparare? E come si sono evolute nel tempo?
Vediamolo insieme!

Come le Reti "Studiano"

L'apprendimento supervisionato (*supervised learning*) è un metodo in cui una rete neurale viene **addestrata con esempi**.
Immaginiamo di insegnare a un bambino a riconoscere i cani e i gatti:

1. Gli mostriamo una foto e gli diciamo *"Questo è un gatto"*.
2. Se sbaglia, lo correggiamo.
3. Dopo molte ripetizioni, impara a fare la distinzione da solo.

Le reti neurali artificiali imparano nello stesso modo!

Struttura dell'Apprendimento Supervisionato

- **Input**: un dato iniziale (es. un'immagine).
- **Output desiderato**: la risposta corretta.
- **Errore**: la differenza tra l'output della rete e il valore corretto.
- **Correzione dei pesi**: la rete si auto-corregge regolando i parametri interni.

Matematica dell'Apprendimento: Come si Aggiornano i Pesi?

L'errore tra l'output previsto y e quello reale y_{vero} viene misurato con una **funzione di perdita**:

$$E = \frac{1}{2} \sum (y - y_{\text{vero}})^2 \tag{6}$$

Per ridurre questo errore, usiamo un algoritmo chiamato **Backpropagation** (*retropropagazione dell'errore*), che aggiorna i pesi w_i seguendo la **discesa del gradiente**:

$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i} \tag{7}$$

Cosa Significa?

- La rete calcola **quanto** ogni peso ha contribuito all'errore.
- Se un peso ha spinto troppo verso la risposta sbagliata, lo riduciamo.
- Se un peso ha contribuito alla risposta giusta, lo aumentiamo.
- Ripetiamo il processo **molte volte**, finché la rete impara!

Esempio: Tirare una Palla in un Canestro

Immagina di imparare a tirare una palla in un canestro.

- Se tiri troppo forte, la prossima volta provi a tirare un po' più piano.
- Se tiri troppo piano, aumenti la forza.
- La rete neurale fa lo stesso: regola i suoi **pesi** per migliorare la precisione.

Il Perceptron: Il Neurone Artificiale più Semplice (1957)

Il Perceptron è il modello più basilare di neurone artificiale.
Funziona come un semplice "interruttore":

- Se l'input supera una certa soglia, produce un output **1** (acceso).
- Se non la supera, produce un output **0** (spento).

Funzione di Attivazione del Perceptron

$$f(x) = \begin{cases} 1, & \text{se } x > 0 \\ 0, & \text{se } x \leq 0 \end{cases} \tag{8}$$

Limiti del Perceptron

Problema: il Perceptron può risolvere solo problemi molto semplici.
Non può distinguere dati non linearmente separabili, come la funzione XOR.
Esempio pratico:

- Immagina di voler separare con una linea due gruppi di punti su un grafico.
- Se i punti sono distribuiti in modo complesso, una linea dritta non basta!
- Servono modelli più avanzati.

Aggiungiamo più Strati!

Per risolvere problemi più difficili, sono state introdotte le **reti neurali a più strati** (*MLP - Multi-Layer Perceptron*).
Idea chiave: Aggiungiamo **strati nascosti** tra input e output!

- Il primo strato analizza le **caratteristiche di base** (es. bordi di un'immagine).
- Il secondo strato combina queste informazioni per formare **strutture più complesse**.
- Lo strato finale prende la decisione.

Esempio Pratico: Squadra di Investigatori

1. Il primo detective raccoglie indizi (bordi, colori, forme).
2. Il secondo collega gli indizi tra loro (es. orecchie a punta $\rightarrow$ forse un gatto).
3. Il terzo prende la decisione finale (è sicuramente un gatto!).

Matematica: L'output ora passa attraverso **funzioni di attivazione** più complesse, come **ReLU** o **sigmoide**, migliorando la capacità di apprendimento.

Il Problema

Vogliamo addestrare una rete neurale a riconoscere se un'immagine mostra **un gatto o un cane**.

- **Input**: un'immagine trasformata in numeri (pixel tra 0 e 255).
- **Output desiderato**: 1 se è un gatto, 0 se è un cane.

Fase di Inizializzazione

Ogni immagine viene convertita in un vettore di numeri $x = [x_1, x_2, ..., x_n]$.
I pesi $w = [w_1, w_2, ..., w_n]$ vengono inizializzati casualmente.

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{9}$$

Se $y > 0.5$, l'immagine è un **gatto**; altrimenti è un **cane**.

Fase di Apprendimento

1. **Predizione iniziale:** Supponiamo che la rete dica **cane** invece di **gatto** ($y = 0.3$).
2. **Calcolo dell'errore:**

$$E = \frac{1}{2}(0.3 - 1)^2 = 0.245 \tag{10}$$

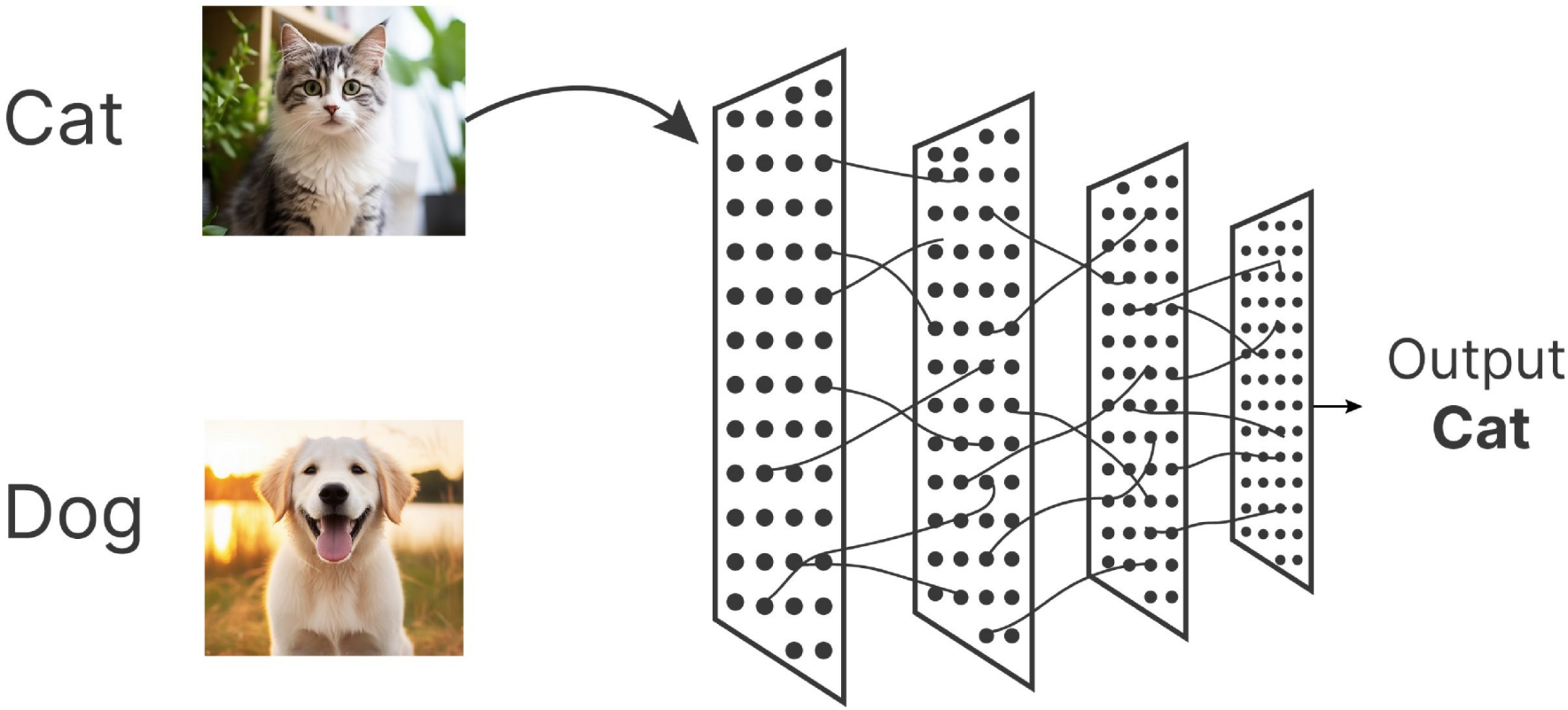
3. **Correzione dei pesi con la discesa del gradiente:**

$$w_i \leftarrow w_i + \eta(y_{\text{vero}} - y)x_i \tag{11}$$

Cosa Significa?

- Se un **pixel** è importante per riconoscere un gatto (es. le orecchie a punta), aumentiamo il suo peso.
- Se un **pixel** è più tipico di un cane (es. il muso lungo), riduciamo il suo peso.
- Dopo **migliaia di iterazioni**, la rete impara a distinguere cani e gatti sempre meglio!

Cosa Significa?



Punti Chiave

- Le **reti neurali** sono ispirate al cervello umano e imparano regolando i pesi delle loro connessioni.
- **L'algoritmo di Backpropagation** aiuta la rete a correggersi, riducendo l'errore.
- I **Perceptron** sono modelli semplici, mentre le **MLP** sono reti più avanzate con più strati nascosti.
- Grazie a queste tecniche, le reti neurali possono **riconoscere immagini, capire testi e risolvere problemi complessi**!

Reti di Hopfield: Come Funzionano e a Cosa Servono

Le **reti di Hopfield** sono un tipo speciale di **rete neurale ricorrente**, usata come **memoria associativa**.
Questo significa che possono **ricordare informazioni anche se ricevono dati incompleti o rumorosi**, proprio come quando riconosciamo una parola scritta male o un viso poco illuminato.

John Joseph Hopfield (Chicago, 15 luglio 1933)

Fisico statunitense, attivo nel campo della biofisica e della fisica statistica. Per il ruolo avuto nello sviluppo della teoria delle reti neurali, e quindi per la successiva nascita del concetto di machine learning, nel 2024 ha ricevuto, assieme a Geoffrey Hinton, il premio Nobel per la fisica.

Cos'è una Memoria Associativa?

Immagina di vedere un vecchio amico in lontananza, ma la sua faccia è un po' sfocata. Anche se non hai tutti i dettagli, il tuo cervello riesce comunque a **riconoscere chi è** basandosi su ciò che ricordi.
Le **reti di Hopfield** fanno qualcosa di simile:

- Se ricevono un'**immagine incompleta**, possono **ricostruire l'originale**.
- Se ricevono un **testo con errori**, possono **indovinare la versione corretta**.

Questa capacità le rende utili per **correzione di errori, riconoscimento di immagini e ottimizzazione di problemi complessi**.

Il Modello Matematico

In una **rete di Hopfield**, ogni neurone può assumere solo due stati: **+1 (acceso)** o **-1 (spento)**.
Matematicamente, lo stato di un neurone i è indicato con:

$$s_i = \pm 1 \tag{12}$$

L'energia totale della rete è:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} s_i s_j \tag{13}$$

Dove:

- E è l'**energia della rete** (più bassa è, più la rete è stabile).
- w_{ij} è il **peso della connessione** tra il neurone i e il neurone j.
- s_i, s_j sono gli **stati dei neuroni**.

Come la Rete Minimizza l'Energia

- Se la rete ha memorizzato bene un pattern, l'energia sarà **bassa**.
- Se la rete riceve un input incompleto, cercherà di **minimizzare l'energia**, modificando i neuroni fino a trovare il pattern più vicino.

Esempio pratico: Immagina una **biglia** su una superficie piena di buche. La biglia **rotola giù** fino a raggiungere il punto più basso.
Allo stesso modo, la rete di Hopfield cerca lo stato con energia **più bassa**, che corrisponde a un pattern memorizzato.

Come la Rete Impara i Pattern

Le reti di Hopfield **memorizzano informazioni** attraverso la **regola di Hebb**.
Se vogliamo memorizzare N immagini (o pattern), i pesi w_{ij} vengono aggiornati con:

$$w_{ij} = \frac{1}{N} \sum_{\mu} x_i^{\mu} x_j^{\mu} \tag{14}$$

Dove:

- x^{μ} rappresenta un **pattern di apprendimento** (es. un'immagine o una parola).
- N è il **numero di pattern memorizzati**.

Significato della Formula

- Se due neuroni si attivano insieme spesso (x_i e x_j hanno lo stesso valore), la loro **connessione si rafforza**.
- Se due neuroni si attivano in modo indipendente, la loro **connessione si indebolisce**.
- La rete impara **associando le connessioni giuste** per ogni pattern.

Esempio pratico: Se vuoi ricordare i volti dei tuoi amici, il cervello **collega dettagli importanti** (occhi, capelli, altezza). Se incontri qualcuno con tratti simili, il cervello **riconosce automaticamente** la persona giusta.

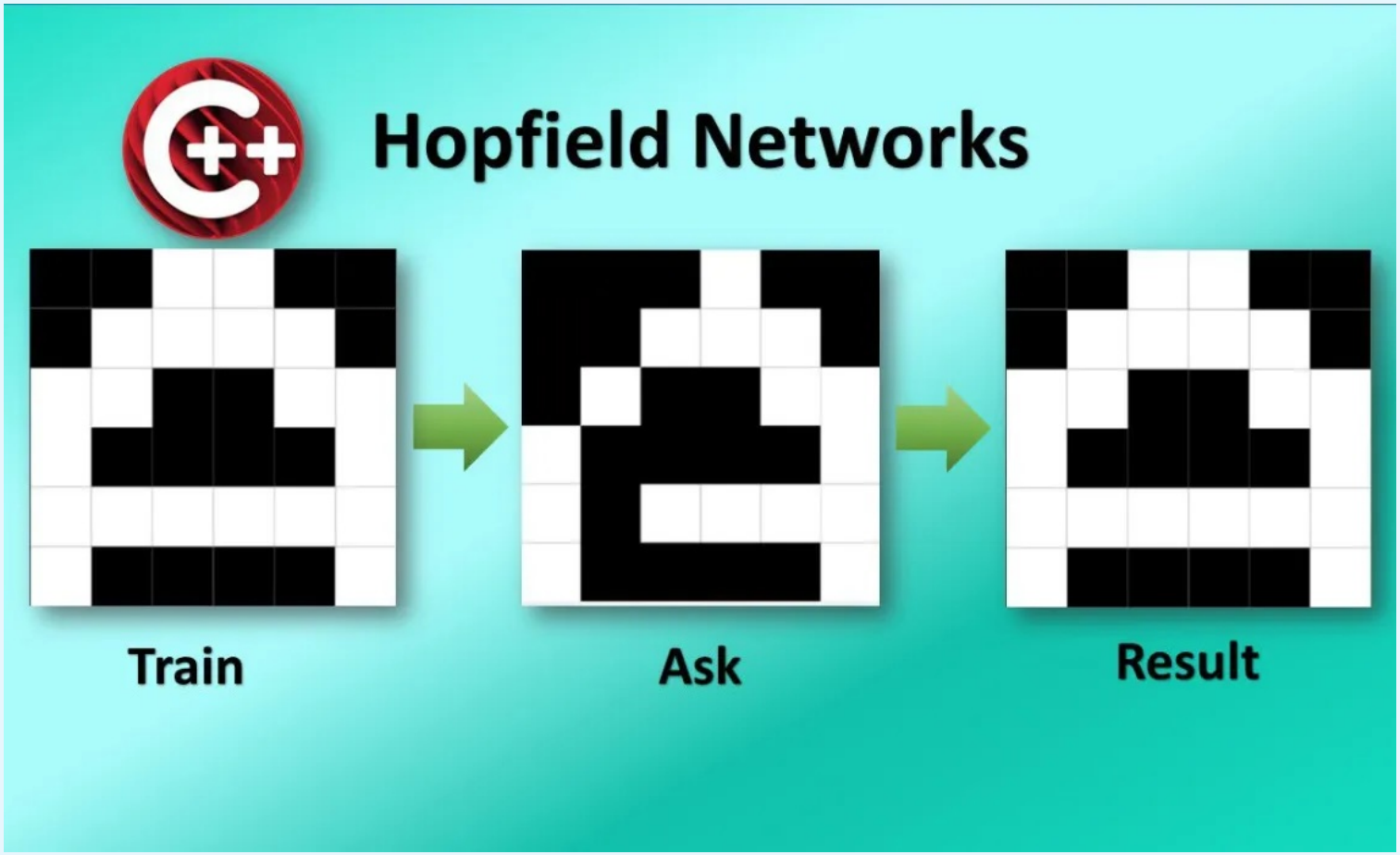
Dove si Usano le Reti di Hopfield?

- **Riconoscimento di Immagini**: Se una foto è danneggiata, la rete può **ricostruire l'originale**.
- **Correzione di Errori nei Testi**: Se una parola ha lettere mancanti o sbagliate, la rete può **indovinare la parola giusta**.
- **Memoria nei Computer**: Si possono usare per **ricordare informazioni senza bisogno di database enormi**.
- **Ottimizzazione Combinatoria**: Si usano per risolvere problemi complessi come il **commesso viaggiatore** (trovare il percorso più breve tra più città).

Riepilogo Finale

- **Le reti di Hopfield** funzionano come una **memoria associativa**.
- Possono **ricostruire informazioni** anche se ricevono dati incompleti o rumorosi.
- Minimizzano un'**energia totale**, come una **biglia che rotola verso il punto più basso**.
- Usano la **regola di Hebb** per memorizzare i pattern.
- Sono utili per **riconoscimento immagini, correzione errori e ottimizzazione**.

Esempio di utilizzo Rete di Hopfield



Introduzione ai Transformer

Negli ultimi anni, i **Transformer** hanno rivoluzionato l'intelligenza artificiale, specialmente nel campo dell'**elaborazione del linguaggio naturale (NLP)**.
Sono il cuore di modelli come **ChatGPT, Google Translate e BERT**.
Ma perché i Transformer sono così importanti? Per capirlo, facciamo un viaggio nella storia dei modelli di IA per il linguaggio.

Come siamo arrivati ai Transformer?

Immagina di dover completare la frase:

"Il gatto salta sul..."

Noi esseri umani possiamo facilmente indovinare che la parola più probabile sia **"tavolo"**, **"divano"** o **"letto"**, perché abbiamo esperienza del mondo e capiamo il contesto.
Ma come può farlo una macchina?

Evoluzione dei modelli di linguaggio

In passato, gli scienziati hanno sviluppato diversi metodi per far apprendere alle macchine il linguaggio:

1. **Modelli basati su regole** (anni '50-'90): Liste di regole grammaticali, troppo rigide.
2. **Modelli statistici** (anni '90-2000): Analizzavano parole più usate insieme, ma senza comprensione profonda.
3. **Reti Neurali Ricorrenti (RNN, 2000-2010)**: Leggevano il testo parola per parola, ma avevano problemi di memoria.
4. **LSTM (Long Short-Term Memory, 1997)**: Miglioravano la memoria delle RNN, ma erano lenti da addestrare.
5. **Transformer (2017 - "Attention Is All You Need")**: **Salto rivoluzionario!** Permettono di leggere tutta la frase in parallelo e dare più importanza alle parole utili.

Perché i Transformer sono rivoluzionari?

- Risolvono il problema della memoria a lungo termine nelle RNN.
- Sono molto più veloci nell'elaborare il testo, grazie alla **parallelizzazione**.
- Permettono alle IA di **comprendere e generare testo in modo più naturale**.

Architettura dei Transformer

I Transformer sono composti da due parti principali:

1. **Encoder**: Legge il testo di input e ne crea una rappresentazione numerica.
2. **Decoder**: Usa quella rappresentazione per generare testo in uscita (utile per traduzione, chatbot, ecc.).

Componenti chiave dei Transformer

Ogni **encoder e decoder** ha due elementi fondamentali:

- **Self-Attention**: Permette di capire quali parole della frase sono più importanti.
- **Reti Neurali Feed-Forward**: Elaborano le informazioni per creare la risposta finale.

Esempio intuitivo: Lettura selettiva

Immagina di studiare per un esame e avere **20 pagine di appunti**. Non leggi tutto parola per parola!
Il tuo cervello seleziona automaticamente le parti più importanti.
Il Transformer fa esattamente lo stesso: legge una frase e **"decide" quali parole sono più rilevanti** per il significato generale.

Il Meccanismo di Self-Attention

Self-Attention (Auto-Attenzione)

La Self-Attention (Auto-Attenzione) è il segreto del Transformer.
Permette di dare **più peso** alle parole più importanti della frase.

Esempio: Interpretare una frase in modo intelligente

Immagina la frase:

> *"La palla era così veloce che il portiere non è riuscito a prenderla."*

Se una RNN leggesse la frase parola per parola, quando arriva a **"prenderla"**, potrebbe aver dimenticato che si riferisce a **"palla"**.
Il Transformer invece **analizza tutta la frase in parallelo** e capisce che **"prenderla" si riferisce a "palla"**.

Matematica della Self-Attention

Per fare ciò, il Transformer usa tre "strumenti" per ogni parola della frase:

- **Query (Q)**: "Quali parole della frase sono importanti per me?"
- **Key (K)**: "Quanto sono importanti rispetto alle altre parole?"
- **Value (V)**: "Quali informazioni mi danno?"

Formula dell'attenzione

Il modello confronta ogni parola con le altre e assegna **pesi numerici** basati sulla loro importanza:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{15}$$

Idea chiave: Il Transformer assegna pesi alle parole esattamente come facciamo noi leggendo un testo!

Multi-Head Attention: Guardare la Frase in Modi Diversi

Un altro superpotere dei Transformer è il **Multi-Head Attention**.
Esempio: Analizzare un testo da più punti di vista

- Immagina di guardare una città dall'alto con un drone.
- Se hai una sola telecamera, vedi solo un angolo alla volta.
- Se hai più telecamere, puoi vedere tutto contemporaneamente da **diverse prospettive**.

Ogni "testa di attenzione" nel Transformer funziona in questo modo:

- Una testa può concentrarsi sulla **struttura grammaticale**.
- Un'altra sulla **relazione tra le parole**.
- Un'altra ancora sul **significato complessivo**.

Positional Encoding: Riconoscere l'Ordine delle Parole

Un problema dei Transformer è che, a differenza delle RNN, **non leggono il testo parola per parola in ordine**.
Per dare un **senso alla sequenza delle parole**, si aggiunge un **Positional Encoding**, una specie di **tag numerico** che indica la posizione di ogni parola.

Formula del Positional Encoding

Si introduce un **Positional Encoding**, definito come:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{16}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{17}$$

Esempio: Se scambi "Il cane morde il gatto" con "Il gatto morde il cane", il significato cambia!
Il Transformer tiene conto dell'ordine grazie ai **Positional Encoding**.

Il Processo di Addestramento dei Transformer

I Transformer vengono addestrati utilizzando **dataset enormi** e algoritmi di **ottimizzazione** come Adam.

Fasi di addestramento:

1. **Pre-Training**: Il modello viene addestrato su grandi quantità di testo (es. tutto il web) con task come **Masked Language Model** (BERT) o **Next Word Prediction** (GPT).
2. **Fine-Tuning**: Il modello viene adattato a compiti specifici, come **traduzione automatica**, **chatbot**, **analisi del sentiment**, ecc.

Differenze tra Transformer e Reti Precedenti

Caratteristica	RNN / LSTM	CNN	Transformer
Parallelizzazione	Bassa	Alta	Altissima
Memoria a lungo termine	Limitata	Limitata	Ottima
Efficienza su sequenze lunghe	Scarsa	Buona	Ottima
Addestramento GPU	Difficile	Facile	Facile
Applicazioni principali	Testo sequenziale	Immagini	Testo, immagini, audio

Il Successo dei Transformer: Dove Vengono Usati?

I Transformer sono ormai alla base di **quasi tutte le applicazioni di Intelligenza Artificiale più avanzate**.
Applicazioni principali:

- **Chatbot e assistenti virtuali** (ChatGPT, Siri, Alexa)
- **Traduzione automatica** (Google Translate)
- **Generazione di testo e scrittura automatica** (GPT-4)
- **Analisi del sentiment** (capire emozioni nei testi)
- **Riconoscimento vocale e immagini** (Vision Transformer)

Il Futuro dei Transformer

I Transformer sono stati una **rivoluzione** per l'Intelligenza Artificiale. Sono capaci di **capire il linguaggio umano** in modo molto più efficace rispetto ai modelli precedenti.
Nel futuro possiamo aspettarci:

- **Modelli più piccoli ed efficienti** per l'uso quotidiano.
- **Chatbot ancora più realistici** e interattivi.
- **Applicazioni in medicina, scienza e arte**, con IA che aiutano nella ricerca e nella creatività.

Riepilogo

I Transformer

I Transformer hanno dato all'Intelligenza Artificiale la capacità di leggere e capire il linguaggio come mai prima d'ora.

Cosa Possiamo Fare con un LLM come ChatGPT?

I **modelli di linguaggio** sono stati addestrati su enormi quantità di dati (GPT4 ha più di 175miliardi di parametri!) per comprendere e generare testo in modo simile agli esseri umani.

Esempi di Utilizzo:

- **Scrivere e Riassumere Testi**: Generare articoli, saggi, email; Riassumere testi lunghi in poche frasi.
- **Rispondere a Domande e Spiegare Concetti**: Spiegare concetti scientifici, storici, matematici.
- **Scrivere Codice e Risolvere Bug**: Aiutare nello sviluppo di software (Python, Java, C++...); Individuare e correggere errori nei programmi.
- **Tradurre e Analizzare Testi**: Tradurre tra lingue diverse; Analizzare il tono e il significato di un testo.
- **Fornire Suggerimenti Creativi**: Generare poesie, storie, idee per progetti artistici.
- **Aiutare nello Studio e nel Lavoro**: Generare appunti e schemi per gli studenti; Fornire consigli su come scrivere un CV.

Perché la Risposta Migliora con una Domanda più Dettagliata?

Più dettagli fornisci, più la risposta sarà precisa!

I modelli di linguaggio **non "pensano" come noi**, ma generano le risposte **prevedendo la parola più probabile successiva** in base alla domanda.

- **Se fai una domanda generica**, otterrai una risposta generica.
- **Se fai una domanda specifica**, il modello avrà più informazioni per formulare una risposta migliore.

Esempio (1/2)

Domanda generica:

"Dimmi qualcosa su Leonardo da Vinci."

Risposta generica:

"Leonardo da Vinci era un artista e scienziato rinascimentale."

Esempio (2/2)

Domanda specifica:

"Quali sono le principali invenzioni di Leonardo da Vinci e quale impatto hanno avuto sulla scienza moderna?"

Risposta più dettagliata:

"Leonardo progettò macchine volanti, dispositivi idraulici e strumenti bellici. Molte sue idee anticiparono le moderne scoperte nel campo dell'aerodinamica e dell'ingegneria meccanica."

Conclusione: Più dettagli forniamo, più il modello può **costruire una risposta mirata e approfondita**.

Perché ChatGPT Migliora Durante una Conversazione?

Un'altra caratteristica interessante è che le risposte **migliorano man mano che si interagisce**, perché il modello tiene traccia del **contesto della conversazione**.

Idea chiave:

- Ogni nuova domanda fornisce **informazioni aggiuntive** sulla direzione e sul dettaglio desiderato.
- Il modello **memorizza** quanto già emerso, evitando di dover "ripartire da zero".

Esempio di interazione (1/2)

Domanda:

"Chi era Pitagora?"

Risposta:

"Pitagora era un matematico e filosofo greco noto per il teorema che porta il suo nome."

Esempio di interazione (2/2)

Domanda successiva:

"Mi fai un esempio del suo teorema?"

Risposta:

"Certo! Il teorema di Pitagora afferma che in un triangolo rettangolo, la somma dei quadrati dei cateti è uguale al quadrato dell'ipotenusa: $a^2 + b^2 = c^2$*."*

Cosa è successo?

- Il modello ha **memorizzato il contesto** e risposto in modo coerente.
- Non ha dovuto "ricominciare da zero".

Conclusione: Se fai domande sequenziali su un argomento, ChatGPT può dare risposte sempre più **precise e utili**.

Hallucination (Allucinazioni)

Anche se i LLM sono incredibilmente potenti, non sono perfetti. Possono commettere errori e, a volte, generare informazioni completamente false: questo fenomeno si chiama **allucinazione**.

Cos'è un'Allucinazione in un LLM?

Un'allucinazione si verifica quando il modello **inventa informazioni** che sembrano plausibili ma in realtà non sono vere.
Esempio di allucinazione:

Domanda

"Qual è il nome dell'unico astronauta italiano che è stato su Marte?"

Risposta (Errata)

"L'astronauta italiano Marco Rossi è stato il primo a mettere piede su Marte nel 2025."

Problema: Nessun essere umano è mai stato su Marte! Il modello ha **creato una risposta** basandosi su informazioni errate, perché non ha una vera **comprensione della realtà**.

Perché i LLM Sbagliano?

I motivi principali per cui i LLM possono commettere errori:

- **Dati Imperfetti**: I testi umani possono contenere errori o opinioni soggettive.
- **Mancanza di Conoscenza Aggiornata**: Senza accesso a dati recenti, il modello può risultare obsoleto.
- **Nessuna "Comprensione" Vera**: Calcolano *probabilità*, non ragionano come un essere umano.
- **Domande Ambigue**: Se la domanda è confusa, la risposta può esserlo altrettanto.

Esempio di errore per dati imperfetti:
Se il modello è addestrato su testi vecchi, potrebbe sostenere che **Plutone** sia ancora un pianeta, ignorando che la sua classificazione è cambiata nel 2006.

Riepilogo Finale

Punti chiave:

- I LLM come ChatGPT sono **strumenti versatili** per scrivere, tradurre, programmare e altro.
- **Più dettagli** forniamo nelle domande, **migliori** saranno le risposte.
- Durante una conversazione, il modello **migliora** perché **tiene traccia del contesto**.
- Possono **commettere errori (allucinazioni)** inventando risposte *apparentemente vere*.
- Non "pensano" come noi, ma **generano testo basandosi su probabilità**.

Come Evitare le Allucinazioni nei Modelli di Linguaggio

I **LLM** (Large Language Models) come ChatGPT sono **potentissimi**, ma a volte generano **informazioni errate o inventate**.

Cosa fare?

- Formulare **domande precise** e ben strutturate.
- Verificare le informazioni fornite, se possibile **consultando fonti esterne**.
- Essere consapevoli che il modello **non ha** una vera comprensione della realtà.
- Non dare per scontato che il modello **ci corregga** se la domanda contiene già un errore.

Obiettivo: Usare questi strumenti **in modo consapevole** e ridurre il rischio di allucinazioni.

Come si Generano le Allucinazioni nei LLM?

I modelli di linguaggio **non "sanno"** cosa sia vero o falso, poiché non possiedono una comprensione effettiva della **realtà**.

Perché succedono le allucinazioni?

- I LLM **generano testo** parola per parola in base alle probabilità apprese.
- Domande **fuorvianti** o errate possono indurre risposte inventate.
- Se il modello **non ha abbastanza informazioni**, potrebbe comunque **tentare di rispondere**, producendo errori.

Esempio di Allucinazione e Come Evitarla

Domanda fuorviante:
"Qual è il nome dell'unico astronauta italiano che è stato su Marte?"
Risposta inventata:
"L'astronauta Marco Rossi è stato il primo italiano a mettere piede su Marte nel 2025."
Errore: Nessun essere umano è mai stato su Marte!
Domande alternative (formulate meglio):

- *"Ci sono astronauti italiani che sono stati su Marte?"*
- *"Quali astronauti italiani hanno partecipato a missioni spaziali?"*

Conclusione: Se una domanda **presuppone un'informazione errata**, il modello potrebbe non correggerci e **inventare** una risposta coerente ma falsa.

Strategie per Evitare le Allucinazioni

Evitare le allucinazioni

Ora vediamo alcune **tecniche pratiche** per evitare risposte errate o fuorvianti.

1. Fai Domande Aperte e Neutre

Evita di dare per scontata una risposta e lascia spazio al modello per fornire una risposta basata sui **dati reali**.

Domanda Errata

"Quale studioso medievale ha dimostrato che la Terra è piatta?"

Domanda Corretta

"Qual era la concezione della forma della Terra nel Medioevo?"

Altro Esempio

- *"Quando l'IA ha superato l'intelligenza umana?"*
- *"Quali sono i limiti attuali dell'IA rispetto all'intelligenza umana?"*

2. Chiedi Fonti o Riferimenti

Se una risposta ti sembra sospetta, **chiedi** *da dove provengono le informazioni*.

- *"Puoi fornire una fonte attendibile per questa informazione?"*
- *"Questa informazione è supportata da ricerche scientifiche?"*

Attenzione: ChatGPT e altri LLM **non hanno accesso a Internet in tempo reale**, quindi non sempre possono verificare informazioni aggiornate.

3. Verifica le Informazioni con Fonti Affidabili

Non prendere mai per *oro colato* ciò che dice un modello di IA! Meglio **controllare con fonti esterne**.

- Confronta la risposta con siti attendibili (**NASA, Treccani, Wikipedia, Nature**).
- Se l'argomento è complesso, **verifica con più fonti indipendenti**.

Esempio pratico: Se il modello dice *"L'Italia ha lanciato una missione su Marte nel 2023,"* controlla sul sito dell'**ESA** o della **NASA** per verificare se è vero.

4. Usa Domande di Controllo per Verificare la Coerenza

Fare più domande sullo stesso argomento e confrontare le risposte è un'ottima strategia.

Esempio di Contraddizione

"Chi è stato il primo astronauta su Marte?" $\Rightarrow$ *"Nessun astronauta è stato su Marte."*
"Chi è l'astronauta italiano che è stato su Marte?" $\Rightarrow$ *"L'astronauta Marco Rossi è stato su Marte nel 2025."*

Errore: Il modello si **contraddice**! Prima nega che qualcuno sia stato su Marte, poi **inventa** un nome.
Conclusione: Se le risposte **non sono coerenti**, l'informazione potrebbe essere falsa.

5. Evita Domande Troppo Specifiche su Eventi Recenti

I LLM **non sempre hanno informazioni aggiornate**.

- **Domanda errata:** *"Chi ha vinto il premio Nobel per la Fisica quest'anno?"*
- **Domanda corretta:** *"Dove posso trovare l'elenco dei vincitori del Nobel per la Fisica?"*

Soluzione: Per eventi recenti, **cerca siti ufficiali** o fonti **autorevoli**.

Riepilogo Finale

Punti Chiave:

- Le **allucinazioni** sono risposte **false o inventate** dai modelli di linguaggio.
- Una **domanda mal formulata** può portare a una risposta altrettanto **errata**.
- Per evitare errori, **fai domande aperte** e **neutrali**.
- **Verifica** sempre le informazioni con **fonti esterne**.
- Usa domande di **controllo** per **testare la coerenza** del modello.