



Le Reti Neurali

***Introduzione alle Reti Neurali
Storia, Evoluzione e Impatti nelle Applicazioni Moderne***

Dott. Ing. Francesco Micillo
Presidente Commissione Informatica
Ordine degli Ingegneri della Provincia di Roma

Roma, 2 dicembre 2025



Chi sono:

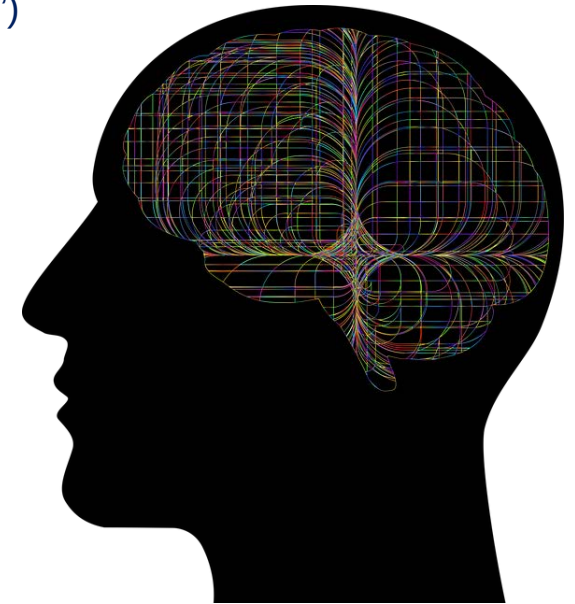
Dott. Ing. Francesco Micillo



Introduzione

Punti chiave:

- Cos'è una rete neurale
- Evoluzione storica e applicazioni
- Collegamento all'intervento del collega ("Modello connessionista – Pietre miliari ed esempi")



Perché è importante capire il modello connessionista oggi:

- Le reti neurali sono oggi alla base della maggior parte dei progressi dell'Intelligenza Artificiale.
- Comprenderne i principi aiuta a capire come funzionano i sistemi moderni e quali sono i loro limiti.

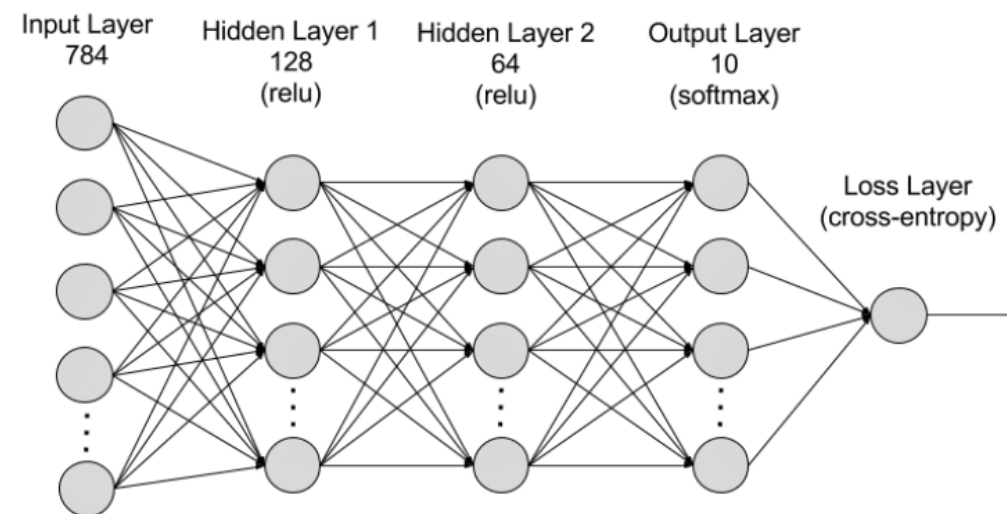
Cosa Sono le Reti Neurali?

Definizione:

- Una rete neurale è un modello computazionale ispirato al cervello umano, costituito da unità interconnesse chiamate "unità computazionali".
- Le reti neurali sono utilizzate per riconoscere pattern, fare previsioni e apprendere dai dati.

Struttura:

- **Input Layer:** I dati in ingresso
(ad esempio, immagini, testo, numeri).
- **Hidden Layers:** Strati intermedi che elaborano le informazioni.
- **Output Layer:** Il risultato finale (ad esempio, una previsione o una classificazione).





Perché le Reti Neurali Sono Utili?

- **Capacità di Apprendere dai Dati:**

- A differenza degli algoritmi tradizionali, le reti neurali possono migliorare nel tempo semplicemente esponendole a più dati, senza la necessità di specificare regole programmate.
- Questo le rende particolarmente utili per compiti complessi come il riconoscimento di immagini, la traduzione automatica, la sintesi vocale.

- **Applicazioni:**

- **Visione Artificiale:** Riconoscimento facciale, diagnostica medica.
- **Elaborazione del Linguaggio Naturale:** Traduzione automatica, chatbots, analisi del sentiment.
- **Previsioni:** Reti neurali applicate alla previsione di serie temporali, ad esempio, nel settore finanziario.

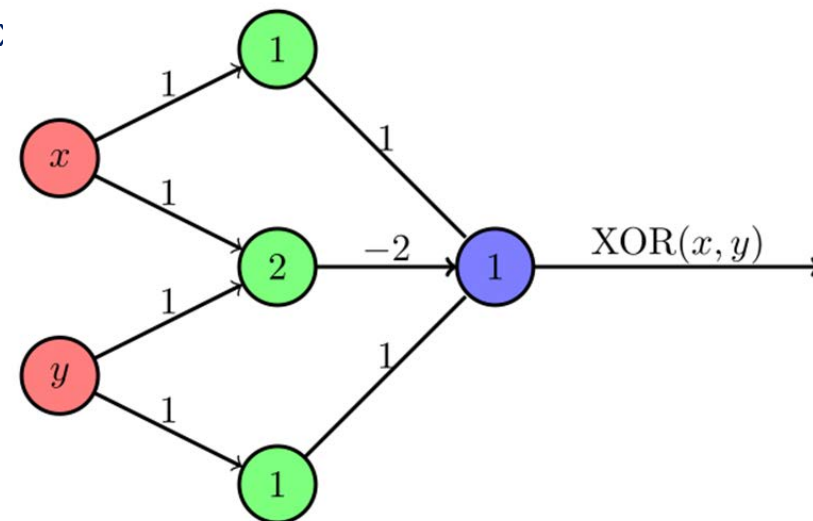
Una caratteristica fondamentale è la capacità delle reti neurali di apprendere rappresentazioni gerarchiche dei dati: dai dettagli più semplici alle caratteristiche più complesse.

Origini delle Reti Neurali

- **Anni '50 e '60: Il Perceptron**

- **Frank Rosenblatt (1958)** crea il perceptron, un modello di neurone artificiale capace di fare classificazioni binarie.
- **Limiti del Perceptron:** Non riusciva a risolvere problemi non linearmente separabili, come il famoso problema XOR.
- *Grafico:* mostra la limitazione del perceptron nel separare i punti nel caso :

Il problema XOR bloccò per anni la ricerca



- **Anni '70:** la ricerca subì una forte battuta d'arresto proprio a causa del limite del perceptron, noto come AI Winter

- **Anni '80: Backpropagation e Reti Multistrato**

- L'algoritmo di **Backpropagation** (retropropagazione dell'errore) sviluppato da **Rumelhart, Hinton e Williams (1986)** permette di allenare reti con più strati (multilayer perceptrons - MLP).

Conseguenza: Le reti neurali possono ora imparare funzioni non lineari complesse.



● Perché il Perceptron non può apprendere lo XOR

Il perceptron semplice può apprendere solo funzioni che sono linearmente separabili, cioè funzioni in cui esiste una retta (in 2D), un piano (in 3D) o un iperpiano (in generale) che separi completamente le classi.

1. 🔍 Cos'è lo XOR?

La funzione XOR prende due ingressi binari e restituisce 1 se *sono diversi*, 0 se *sono uguali*:

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0



2. ● Distribuzione dei punti

Se rappresentiamo i punti nel piano:

- $(0,0) \rightarrow$ classe 0
- $(1,1) \rightarrow$ classe 0
- $(0,1) \rightarrow$ classe 1
- $(1,0) \rightarrow$ classe 1

Il risultato è questo:

classe 1: $(0,1)$ $(1,0)$

classe 0: $(0,0)$ $(1,1)$

3. ✗ Nessuna retta può separarli

I punti appartenenti alle due classi si trovano “incrociati” a scacchiera. Non esiste **una sola retta** che separi simultaneamente i punti con output 1 da quelli con output 0.

👉 Questo significa che la funzione **non è linearmente separabile**.

Il perceptron semplice esegue la classificazione tramite:

$$y = \text{step}(w_1x_1 + w_2x_2 + b)$$

che corrisponde geometricamente a dividere il piano con una *linea*.

Ma per lo XOR servirebbe almeno una divisione “a croce”, non lineare.



4. 🧠 Perché il perceptron fallisce matematicamente?

Il perceptron cerca pesi w_1 , w_2 , b che soddisfino:

- $w_1(0) + w_2(0) + b < 0 \rightarrow (0,0)$
- $w_1(1) + w_2(1) + b < 0 \rightarrow (1,1)$
- $w_1(0) + w_2(1) + b > 0 \rightarrow (0,1)$
- $w_1(1) + w_2(0) + b > 0 \rightarrow (1,0)$

Queste quattro condizioni sono **incompatibili simultaneamente**: non esiste alcuna combinazione di pesi che le soddisfi tutte.

5. ✔ Come si risolve allora il problema?

Serve un modello in grado di creare **frontiere non lineari**, ad esempio:

- una rete neurale a due layer (MLP)
- una funzione di attivazione non lineare (sigmoid, ReLU...)
- kernel methods (SVM con RBF)

Una rete neurale con **almeno un livello nascosto** può infatti combinare più perceptron, creando una frontiera non lineare che separa lo XOR.



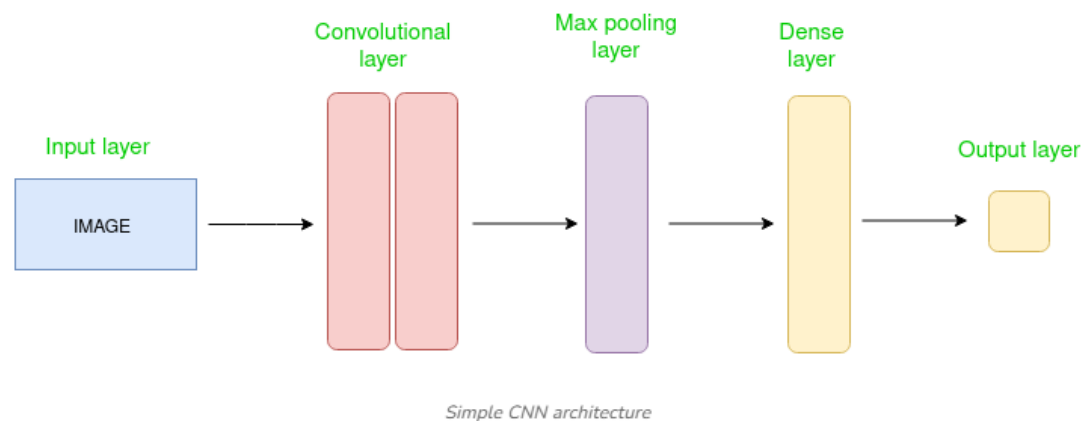
In sintesi

Il perceptron semplice fallisce nel caso XOR perché:

1. È un classificatore lineare.
2. Lo XOR non è linearmente separabile.
3. I vincoli sui pesi richiesti dallo XOR sono impossibili da soddisfare.
4. Serve **almeno un livello nascosto** per modellare la non linearità.

Reti Neurali Avanzate

- **Reti Convoluzionali (CNN):**
 - o Progettate per il riconoscimento delle immagini, utilizzano operazioni di convoluzione per identificare caratteristiche come bordi, texture e forme.
 - o Applicazioni: Riconoscimento di oggetti, diagnosi medica tramite immagini (es. radiografie).
 - o *Esempio:* Reti CNN utilizzate per il riconoscimento di immagini nei sistemi di guida autonoma.
- **Reti Ricorrenti (RNN):**
 - o Utilizzano feedback loops per memorizzare informazioni e risolvere problemi sequenziali (es. traduzione automatica, riconoscimento vocale).
 - o *Esempio:* Il sistema di traduzione di Google Translate utilizza RNN.
- **Long Short-Term Memory (LSTM):**
 - o Un tipo particolare di RNN che risolve il problema della "scomparsa del gradiente", migliorando la capacità di memorizzare informazioni a lungo termine.



- **Transformer (2017):** Architettura oggi dominante per linguaggio, visione e multimodalità. Alla base di GPT, Gemini, ecc.



■ Input layer

- È l'immagine stessa che dai in input.
- Può essere, ad esempio, una foto 28×28 , 64×64 , 224×224 , ecc.

■ Convolutional layer (Layer convoluzionali)

- Sono i "filtri" che estraggono caratteristiche dall'immagine.
- Ogni filtro rileva qualcosa: bordi, texture, forme...
- L'output è un insieme di **feature maps**.

👉 È come quando guardi un'immagine cercando linee, spigoli, pattern.

■ Max pooling layer

- Riduce la dimensione delle feature maps.
- Mantiene solo le informazioni più importanti (es. prende il valore massimo in una piccola finestra).

👉 Serve a:

- ridurre il numero di parametri,
- ridurre overfitting,
- aumentare robustezza alle piccole variazioni dell'immagine.

■ Dense layer (Layer completamente connesso)

- Dopo la convoluzione e il pooling, l'immagine è stata "compressa" in un vettore di caratteristiche.
- Questo layer usa quelle caratteristiche per prendere una decisione.

👉 È come la parte "logica" della rete.

■ Output layer

- Fornisce la predizione finale.
- Può essere:
 - una classe (es. gatto / cane),
 - più classi (softmax),
 - un valore continuo (regressione).

🧠 Riassunto

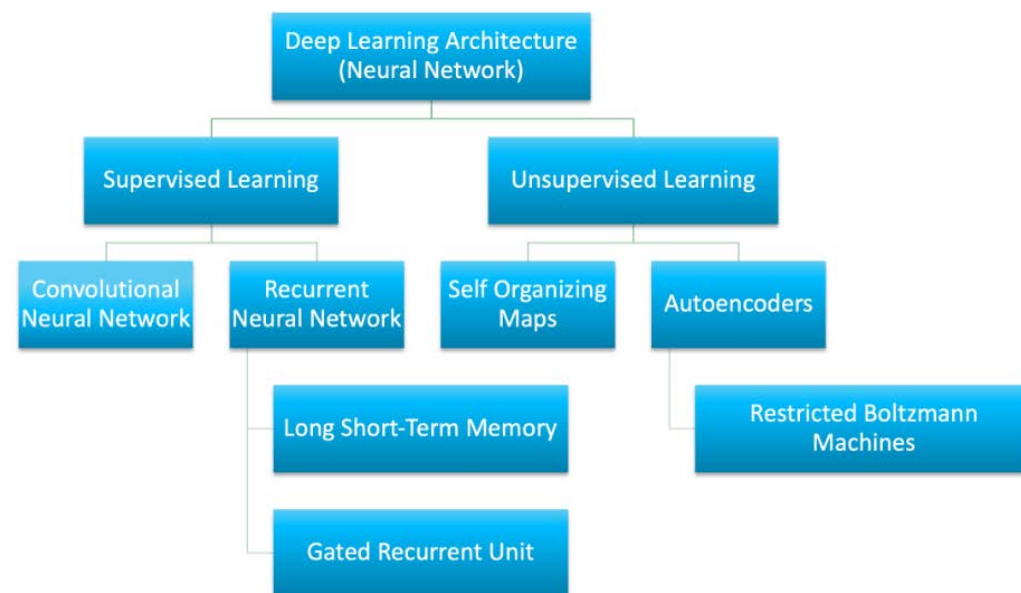
Il flusso è:

Immagine → Convoluzioni → Pooling → Flatten → Dense → Output

È il tipico pipeline per riconoscimento di immagini.

La Rivoluzione del Deep Learning

- **Deep Neural Networks (DNN):**
 - Reti molto profonde che contengono decine, centinaia di strati. Maggiore profondità permette di estrarre rappresentazioni più complesse dai dati.
 - **Applicazioni di successo:** AlphaGo di Google, GPT-5 / modelli multimodali 2025.
- **Generative Adversarial Networks (GANs):**
 - Reti che generano nuovi dati simili a quelli reali (es. immagini sintetiche di volti umani).
 - *Esempio:* Deepfakes, creazione di arte digitale.
- **Progressi tecnologici:**
 - Framework come **TensorFlow** e **PyTorch** hanno semplificato enormemente la costruzione e l'addestramento di reti neurali complesse.
 - Le performance dei modelli migliorano aumentando dati, parametri e capacità computazionale: questo principio è noto come scaling laws.





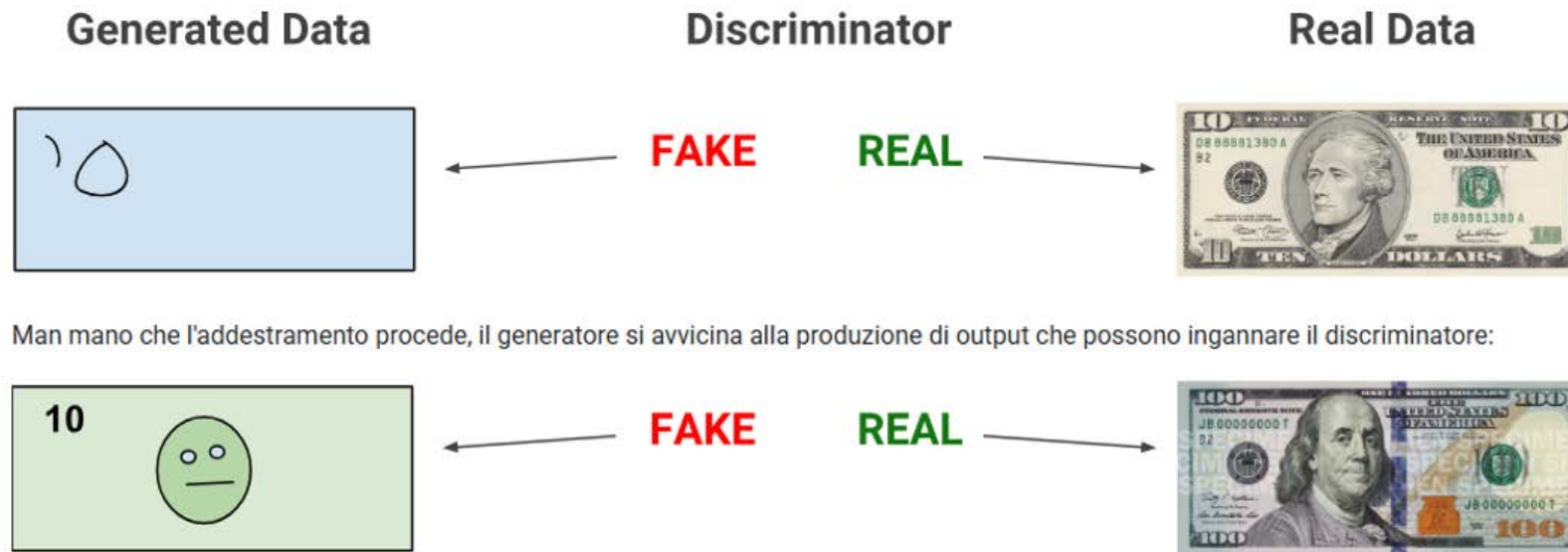
Stato Attuale delle Reti Neurali

- **Applicazioni nel Mondo Reale:**
 - **Settore medico:** Diagnosi automatica di malattie, analisi di immagini mediche.
 - **Automotive:** Guida autonoma, riconoscimento delle immagini stradali.
 - **Intrattenimento:** Sistemi di raccomandazione (Netflix, YouTube), generazione di contenuti (musica, arte).
 - **Sistemi anti-frode**
- **Tecnologie Recenti:**
 - **GPT-4:** Modelli linguistici che possono scrivere, tradurre, rispondere a domande.
 - **AlphaFold:** Previsioni accurate delle strutture proteiche, con un impatto enorme sulla biologia.
 - **AI Agantica / Sistemi autonomi:** Sistemi capaci di pianificare e svolgere sequenze di azioni per raggiungere obiettivi complessi.



La figura rappresenta **come funzionano le GAN (Generative Adversarial Networks)**, cioè le reti neurali generative composte da due parti:

1. **Generatore** → crea dati falsi (fake)
2. **Discriminatore** → cerca di capire se i dati sono veri (real) o falsi (fake)

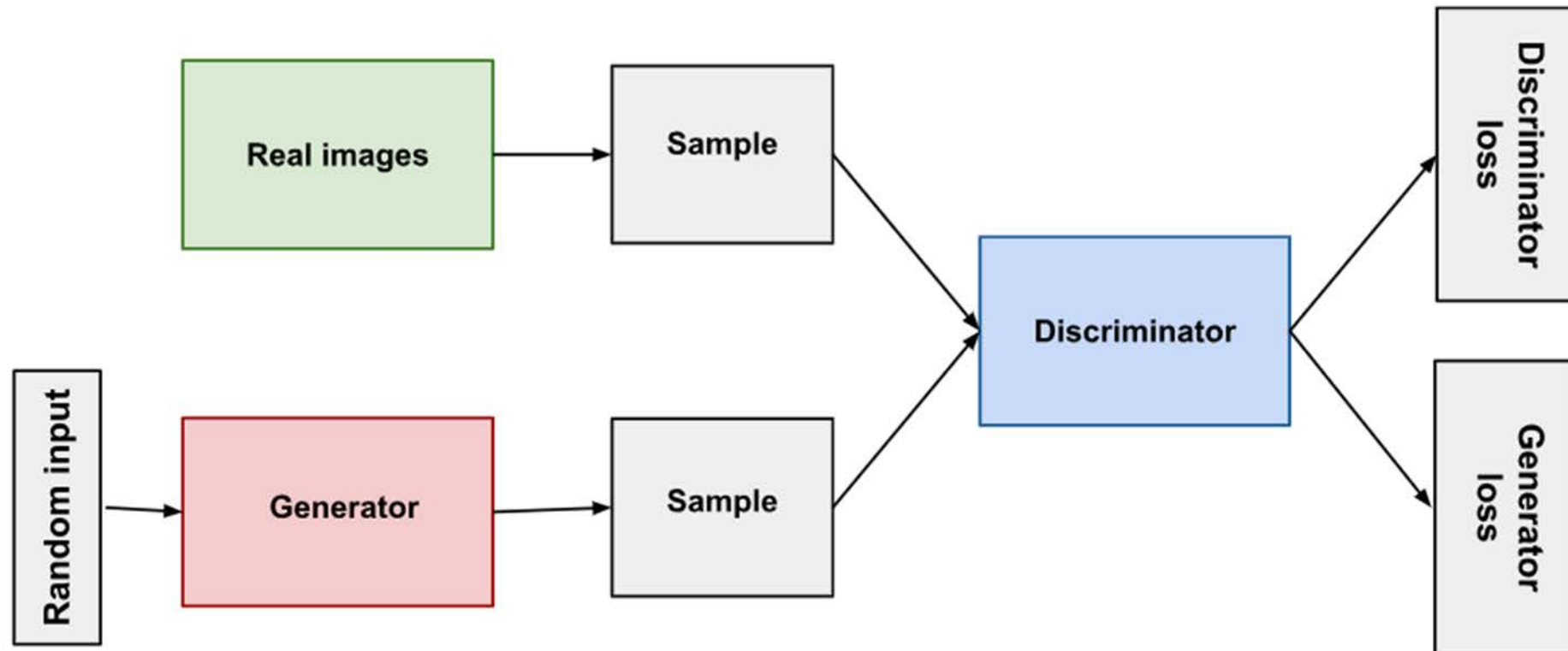


- **Quando il generatore diventa molto bravo**, il discriminatore fa più fatica a distinguere vero da falso.
- Quindi **la sua accuratezza diminuisce**: inizia a classificare i dati falsi come se fossero reali.

Questo è esattamente il principio delle GAN:

- ♦ *due reti si “sfidano” finché il generatore non diventa estremamente realistico.*

Ecco una foto dell'intero sistema:





● Panoramica generale

Una GAN è composta da due reti neurali che competono tra loro:

1. Generator (Generatore)

- Riceve in input un vettore casuale (*Random input*).
- Produce immagini sintetiche.
- Il suo obiettivo è **ingannare il Discriminatore**, generando immagini che sembrano reali.

2. Discriminator (Discriminatore)

- Riceve sia immagini reali che immagini generate.
- Deve capire quali sono vere e quali sono false.
- Il suo obiettivo è non farsi ingannare, migliorando nel distinguere reale da falso.

■ Flusso delle Real images (immagini reali)

1. Le immagini reali entrano nel sistema.
2. La casella "Sample" indica semplicemente che vengono prese batch (campioni) dal dataset.
3. Questi campioni vanno al **Discriminator**.



■ Flusso delle immagini Generate

1. Dal *Random input* (rumore casuale) il **Generator** produce una finta immagine.
2. Anche queste immagini passano attraverso un modulo "Sample".
3. Il campione va al **Discriminator**.

◆ *Discriminator loss*

Misura quanto il discriminatore sbaglia:

- penalizzato se dice che un'immagine falsa è reale
- penalizzato se dice che un'immagine reale è falsa

◆ *Generator loss*

Misura quanto il generatore riesce a ingannare il discriminatore:

- penalizzato se il discriminatore riconosce le immagini generate come false
- premiato se il discriminatore le giudica reali

🌀 In sintesi

Il **Generator** cerca di ingannare il **Discriminator**, mentre il **Discriminator** cerca di smascherare il Generator.

Con il tempo entrambi migliorano, e il risultato è che il generatore produce immagini sempre più realistiche.



Sfide Attuali

- **Spiegabilità:**
 - Le reti neurali sono spesso considerate "scatole nere" perché è difficile capire come arrivano alle loro conclusioni. La ricerca su **AI interpretable** è in crescita.
- **Bias nei Dati:**
 - Le reti neurali possono amplificare pregiudizi nei dati, portando a decisioni discriminatorie (es. nel riconoscimento facciale).
- **Efficienza Energetica:**
 - Le reti neurali, in particolare quelle profonde, richiedono molta potenza di calcolo, il che solleva preoccupazioni in termini di sostenibilità ambientale.
- **Sicurezza e etica:**
 - dipendenza dai dati proprietari.



Conclusioni e Prospettive Future

- **Prospettive Future:**
 - **AI Generale:** Sviluppo di reti neurali in grado di compiere una varietà di compiti come un essere umano.
- **Reti Neurali Quantistiche:** Ipotetiche reti neurali che utilizzano la computazione quantistica per risolvere problemi complessi più velocemente.
- **Spunto: etica, sostenibilità e collaborazione uomo-macchina:** Impatto su lavoro e competenze: Evolveranno anche le competenze: dall'utilizzo di modelli AI alla supervisione responsabile e alla collaborazione uomo-macchina



GRAZIE PER L'ATTENZIONE

Dott. Ing. Francesco Micillo
francescomicillo@libero.it