AI LITERACY LAB

Reti neurali, passo dopo passo

Una guida visuale per capire neuroni, pesi, bias, apprendimento e collegamento con l'IA generativa

PERCHÉ QUESTA DISPENSA

Capire una rete neurale non significa imparare a memoria formule. Significa costruire un modello mentale: quali segnali entrano, come vengono trasformati, come cambiano i pesi durante l'addestramento e come questi principi si collegano ai sistemi di IA generativa.

Dispensa per studenti e corsisti

Ideazione didattica e progettuale: Prof. Erasmo Modica

Realizzazione tecnica dell'app: ChatGPT (OpenAI), sulla base delle indicazioni dell'autore.

Versione ampliata e visuale • Agosto 2026

0 Come usare questa guida

Non serve alcuna preparazione di informatica: procedi nell'ordine e prova subito i concetti nella web app.

OBIETTIVO

Alla fine dovresti saper raccontare con parole tue che cosa fa un neurone artificiale, perché una rete usa più strati, che cosa significa "imparare i pesi" e perché un modello generativo produce il token successivo sulla base di punteggi e probabilità.

La mappa del percorso

Tappa	Domanda guida	Parola chiave
1	Che cosa entra nella rete?	input
2	Come ogni segnale viene amplificato o frenato?	peso
3	Come un neurone combina i segnali?	somma + bias
4	Come decide quanto segnale far passare?	attivazione
5	Come molti neuroni cooperano?	layer
6	Come la rete corregge gli errori?	addestramento
7	Che cosa c'entra con ChatGPT e simili?	token + probabilità

REGOLA DI LETTURA

Quando trovi una formula, leggila prima in parole. Quando trovi una metafora, ricorda che è solo un aiuto: la rete non pensa, non desidera e non "capisce" nel senso umano.

1 Una rete neurale in 90 secondi

Prima il quadro generale, poi entreremo nei dettagli.

Una rete neurale artificiale è una macchina matematica che trasforma numeri in altri numeri. I dati vengono rappresentati numericamente, attraversano una sequenza di strati e producono un output. La caratteristica decisiva è che molti coefficienti interni - i parametri - non vengono scelti a mano uno per uno: vengono adattati durante l'addestramento.

IN UNA FRASE

Input → trasformazioni numeriche con pesi e bias → output. Durante l'addestramento, pesi e bias vengono modificati per ridurre l'errore.

I cinque mattoni

Elemento	Che cosa significa	Domanda utile
Input	i numeri che descrivono il dato	Che cosa sta entrando?
Peso	un moltiplicatore associato a una connessione	Quanto influenza questo segnale?
Bias	un valore aggiunto al totale	Quanto sposto la soglia?
Attivazione	una funzione che trasforma il totale	Quanto segnale passa avanti?
Layer	un gruppo di neuroni che opera in parallelo	Quale trasformazione sta facendo questo strato?

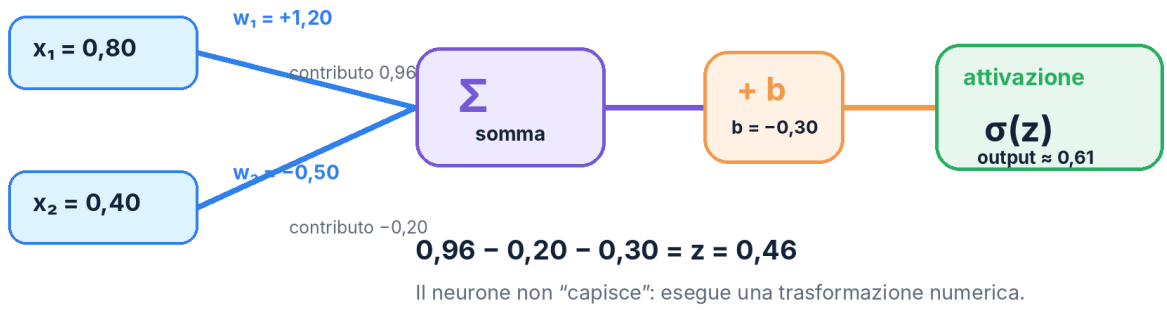
ATTENZIONE AL LINGUAGGIO

"Neurone" è un nome storico. Un neurone artificiale non è una piccola mente: è una funzione matematica parametrizzata.

2 Dentro un neurone: il calcolo fondamentale

È il passaggio più importante dell'intera dispensa.

Dentro un neurone artificiale



0,96 − 0,20 − 0,30 = z = 0,46

Il neurone non "capisce": esegue una trasformazione numerica.

Esempio con due input: ogni freccia porta un segnale e ha un peso.

$$z = x_1 \cdot w_1 + x_2 \cdot w_2 + \ldots + b \quad \rightarrow \quad \text{output} = f(z)$$

Moltiplica → somma → aggiungi il bias → applica la funzione di attivazione.

Leggiamolo senza formule

- Ogni input arriva con un certo valore.
- Ogni connessione moltiplica quel valore per il proprio peso.
- Il neurone somma tutti i contributi.
- Aggiunge il bias, che sposta il totale.
- La funzione di attivazione trasforma il totale e produce l'output del neurone.

> **DA NON CONFONDERE**
>
> Un peso negativo non è un peso "cattivo": significa soltanto che quel collegamento tende a spingere il calcolo nella direzione opposta.

3 Pesi, bias e attivazioni

Tre concetti simili solo in apparenza.

Concetto	Immagine mentale	Ruolo matematico
Peso	manopola del volume su una singola voce	moltiplica un input
Bias	regolazione generale della soglia	si aggiunge alla somma
Attivazione	filtro che trasforma il totale	applica f(z)

Tre funzioni di attivazione da riconoscere

Funzione	Comportamento	Perché è utile
Sigmoid	comprime ogni valore tra 0 e 1	utile per leggere un output come grado/probabilità in alcuni problemi
ReLU	0 se z<0, altrimenti lascia z	semplice ed efficace in molte reti profonde
Tanh	comprime tra -1 e +1	mantiene valori positivi e negativi

PERCHÉ SERVE LA NON LINEARITÀ

Se tutti gli strati facessero solo somme e moltiplicazioni lineari, impilarne molti non renderebbe la rete davvero più espressiva. Le attivazioni non lineari permettono di costruire trasformazioni più complesse.

4 Dalla singola unità alla rete

Il segnale si propaga da sinistra a destra: è la forward propagation.

In una rete feed-forward, l'output di uno strato diventa l'input dello strato successivo. Ogni neurone riceve molti segnali, li pesa, li combina e produce un nuovo numero. I numeri intermedi possono essere interpretati come una nuova rappresentazione del dato.

> **ESEMPIO DI RAPPRESENTAZIONE**
>
> In un sistema per immagini, i primi strati possono reagire a contrasti e bordi; strati successivi possono combinare tali segnali in strutture più complesse. Non bisogna però immaginare che ogni singolo neurone corrisponda sempre a un concetto umano leggibile.

Che cosa osservare nella web app

- Cambia un input e guarda quali attivazioni si modificano.
- Aumenta un peso in valore assoluto: quella connessione diventa più influente.
- Cambia il bias di un neurone: la sua risposta può cambiare anche se gli input restano uguali.
- Confronta Sigmoid e ReLU: la stessa somma z produce output diversi.

5 AND: il caso più semplice da capire

Un singolo neurone a soglia può rappresentare la regola AND.

CHE COSA SIGNIFICA AND

L'uscita vale 1 solo quando $x_1=1$ E $x_2=1$. In tutti gli altri casi vale 0.

x_1	x_2	AND	Lettura
0	0	0	nessun input attivo
0	1	0	solo x_2
1	0	0	solo x_1
1	1	1	entrambi attivi

Costruiamolo con un neurone

$$z = 1 \cdot x_1 + 1 \cdot x_2 - 1{,}5 \quad ; \quad \text{output} = 1 \text{ se } z \geq 0, \text{ altrimenti } 0$$

I due pesi valgono +1; il bias −1,5 fa sì che servano entrambi gli input per superare la soglia.

x_1	x_2	$z = x_1+x_2-1{,}5$	output
0	0	−1,5	0
0	1	−0,5	0
1	0	−0,5	0
1	1	+0,5	1

IL PUNTO CHIAVE

AND è "linearly separable": esiste una singola frontiera lineare che separa il caso 1 dai tre casi 0. Per questo basta un neurone a soglia.

6 XOR: perché uno strato nascosto cambia tutto

XOR significa “uno oppure l’altro, ma non entrambi”.

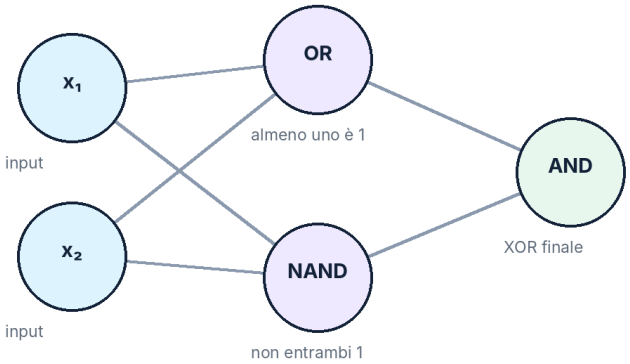
Perché XOR è speciale: serve una rappresentazione intermedia

Tabella di verità XOR

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

Vero solo quando gli input sono diversi.

Una soluzione con uno strato nascosto



Idea chiave: lo strato nascosto costruisce due “indizi” utili (OR e NAND). L’output li combina. È un esempio semplice di rappresentazione appresa.

Una costruzione didattica di XOR usando due neuroni nello strato nascosto.

PERCHÉ UN SOLO NEURONE NON BASTA

Nei quattro casi di XOR, i due punti con output 1 sono “incrociati” rispetto ai due punti con output 0. Non puoi tracciare una sola retta che separi perfettamente le due classi.

La strategia dello strato nascosto

Invece di cercare subito XOR, costruiamo due indizi intermedi: OR (“almeno uno è 1”) e NAND (“non sono entrambi 1”). Poi facciamo AND dei due indizi. Il risultato è esattamente XOR.

x_1	x_2	h_1 = OR	h_2 = NAND	y = AND(h_1,h_2)
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

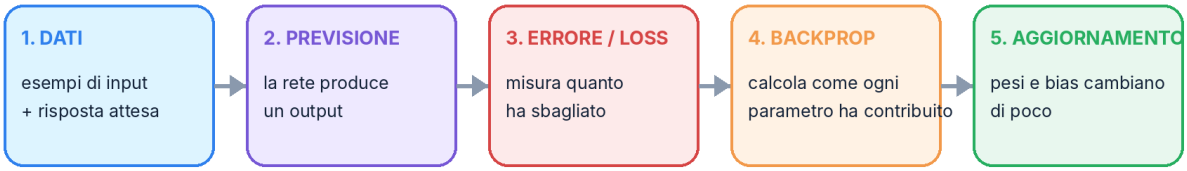
PERCHÉ È IMPORTANTE

Questo minuscolo esempio mostra l'idea delle rappresentazioni intermedie: uno strato nascosto può riorganizzare il problema in caratteristiche che rendono più semplice il calcolo finale. È uno dei motivi per cui reti con più strati possono esprimere relazioni molto più complesse.

7 Come impara una rete

Imparare = modificare parametri per ridurre un errore misurabile.

Come impara una rete: un ciclo ripetuto moltissime volte



↻ **Ripeti: se la loss diminuisce, la rete sta adattando i parametri ai dati di addestramento.**

Le quattro parole da conoscere

Parola	Significato semplice	Da non dire
Loss	un numero che misura quanto la previsione è lontana dall'obiettivo	"è la percentuale di errori" in ogni situazione
Backpropagation	propaga all'indietro l'informazione su come i parametri influenzano la loss	"la rete torna indietro e ragiona"
Gradiente	indica in quale direzione cambiare i parametri per modificare la loss	"dice la risposta corretta"
Learning rate	dimensione del piccolo passo con cui aggiorniamo i parametri	"velocità della rete"

UN'IMMAGINE UTILE

Immagina un paesaggio di colline: l'altezza è la loss. Il gradiente indica la pendenza locale; l'ottimizzatore prova piccoli passi verso zone più basse. È una metafora, ma aiuta a capire la discesa del gradiente.

8 Addestramento e inferenza non sono la stessa cosa

È una distinzione fondamentale per capire i modelli generativi.

Fase	Che cosa succede ai pesi?	Scopo
Addestramento	vengono aggiornati ripetutamente	adattare il modello ai dati
Inferenza / generazione	normalmente restano fissi	usare ciò che è stato appreso per produrre un output

CORREZIONE DI UN EQUIVOCO

Quando un chatbot genera una risposta, non sta di solito “assegnando nuovi pesi” ai neuroni parola per parola. Sta usando i parametri già appresi. I valori che cambiano durante la generazione sono soprattutto le attivazioni interne prodotte dal contesto corrente.

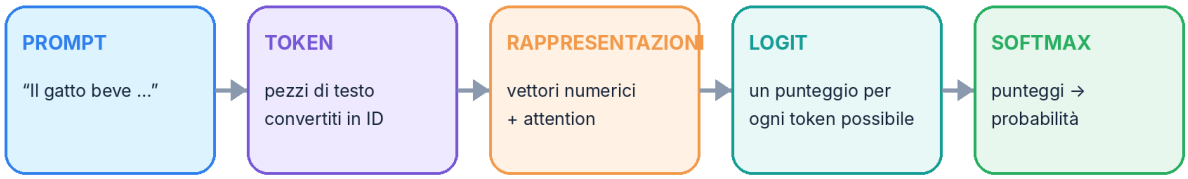
Che cosa cambia davvero mentre scrivi un prompt?

- Cambiano i token in ingresso e quindi le rappresentazioni interne.
- Cambiano le attivazioni dei diversi strati.
- Cambiano i punteggi assegnati ai possibili token successivi.
- I pesi del modello, in condizioni normali, rimangono gli stessi durante quella risposta.

9 Che cosa c'entra con l'IA generativa?

La piccola rete della web app non è un Transformer, ma alcuni principi di base sono gli stessi.

Dalla frase al token successivo: collegamento con l'IA generativa



Esempio didattico (non sono valori reali di ChatGPT):

acqua 0,80 latte 0,12 dorme 0,03 ...

Il modello sceglie / campiona un token, poi ripete il processo.

Un modello linguistico riceve una sequenza di token, costruisce rappresentazioni numeriche del contesto e produce un punteggio (logit) per moltissimi token candidati. Una trasformazione chiamata softmax converte i punteggi in una distribuzione di probabilità. Viene scelto o campionato un token; poi il procedimento ricomincia includendo il token appena prodotto.

ESEMPIO: "IL GATTO BEVE ..."

In una simulazione didattica, "acqua" deve risultare il candidato più plausibile. "Latte" può avere una probabilità più bassa per associazioni presenti nei dati, ma non va presentato come il completamento principale. I valori mostrati dall'app sono illustrativi, non sono le probabilità reali di ChatGPT.

Dove entrano "i neuroni"?

Nei Transformer ci sono grandi blocchi di calcolo con trasformazioni lineari, attivazioni e meccanismi di attention. Durante una singola generazione, moltissime unità producono attivazioni diverse in funzione del contesto. Non è corretto immaginare un neurone isolato che "sceglie la parola": la previsione emerge dal calcolo distribuito dell'intera rete.

10 Attention e Transformer: la differenza da non perdere

Per collegare correttamente le reti neurali moderne all'IA generativa.

ATTENTION IN PAROLE SEMPLICI

L'attention permette a ogni posizione della sequenza di costruire una nuova rappresentazione tenendo conto, con pesi dinamici, delle altre posizioni rilevanti nel contesto. Non sono gli stessi "pesi appresi" delle connessioni: i coefficienti di attention dipendono dall'input corrente.

Tre livelli da distinguere

Livello	Che cosa è	Cambia durante la risposta?
Parametri / pesi appresi	numeri ottimizzati durante l'addestramento	normalmente no
Attivazioni	valori intermedi prodotti dai neuroni/blocchi	sì
Pesi di attention	coefficienti calcolati per mettere in relazione i token del contesto	sì, dipendono dall'input

PERCHÉ QUESTO CONTA

Dire genericamente "il modello attribuisce un peso alle parole" può creare confusione: esistono pesi appresi permanenti e coefficienti di attention calcolati dinamicamente. Sono concetti diversi.

11 Probabilità non significa verità

Un modello linguistico ottimizza la plausibilità del testo, non possiede automaticamente un verificatore dei fatti.

La parola o frase statisticamente plausibile può essere falsa nel mondo reale. Il modello può produrre una sequenza molto fluida ma contenente un dato inventato. Questo aiuta a capire il fenomeno delle “allucinazioni”: non è necessario immaginare che il sistema stia mentendo; sta generando testo in base a pattern e probabilità appresi.

REGOLA DI AI LITERACY

Fluidità linguistica ≠ comprensione umana ≠ accuratezza fattuale. Per informazioni importanti servono verifica, fonti e supervisione.

Che cosa può influenzare l’output?

- Il contesto del prompt.
- I parametri appresi durante l’addestramento.
- Le istruzioni di sistema e gli strumenti eventualmente disponibili.
- La strategia di campionamento e la temperatura.

TEMPERATURA

In modo semplificato: temperatura più bassa rende la distribuzione più “concentrata” sui candidati già favoriti; temperatura più alta rende più probabili alternative meno favorite. Non crea conoscenza nuova: modifica la modalità di campionamento.

12 Laboratorio guidato con Neural Network Explorer

Usa la web app come banco prova, non come semplice animazione.

Esperimento A - “Quanto conta un peso?”

- Imposta due input diversi, per esempio 0,8 e 0,2.
- Tieni tutti gli altri valori fissi e modifica un solo peso.
- Annota come cambia z e poi l’attivazione del neurone.

> **DOMANDA**
> Se raddoppi il peso ma l’input associato è quasi zero, l’effetto è grande o piccolo? Perché?

Esperimento B - AND e XOR

- Carica AND e verifica tutti i quattro casi della tabella di verità.
- Poi carica XOR e osserva che è necessario sfruttare lo strato nascosto.
- Prova a spiegare a voce quale “informazione intermedia” potrebbe essere utile per XOR.

Esperimento C - IA generativa

- Apri la simulazione del token successivo.
- Confronta i candidati per “Il gatto beve …”.
- Modifica la temperatura e descrivi che cosa cambia nella distribuzione.

13 Autoverifica

Se sai rispondere con parole tue, hai costruito il modello mentale essenziale.

1. Qual è la differenza tra input, peso e bias?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

2. Che cosa rappresenta z nel calcolo di un neurone?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

3. Perché una funzione di attivazione non lineare è importante?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

4. Perché un singolo neurone a soglia può realizzare AND?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

5. Perché XOR richiede una trasformazione intermedia?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

6. Che cosa misura la loss?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

7. Che cosa cambia durante l'addestramento e che cosa normalmente resta fisso durante l'inferenza?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

8. In un modello generativo, che cosa sono i logit?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

9. Che cosa fa la softmax?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

10. Perché “token più probabile” non significa “informazione vera”?

Scrivi una risposta di 2-4 righe, senza usare definizioni a memoria.

14 Soluzioni essenziali e glossario

Usale solo dopo aver provato a rispondere.

Q	Risposta essenziale
1	Input = valore che entra; peso = moltiplicatore di una connessione; bias = valore aggiunto alla somma.
2	La somma pesata degli input più il bias, prima dell'attivazione.
3	Permette di comporre trasformazioni che non si riducono a una sola trasformazione lineare.
4	I quattro casi sono separabili con una soglia: solo 1+1 supera la soglia scelta.
5	I casi positivi e negativi di XOR non sono separabili con una sola frontiera lineare; uno strato nascosto può costruire caratteristiche intermedie.
6	Un numero che quantifica quanto la previsione è lontana dall'obiettivo secondo una funzione scelta.
7	In addestramento si aggiornano pesi e bias; in inferenza normalmente i parametri restano fissi mentre cambiano input e attivazioni.
8	Punteggi grezzi assegnati ai token candidati prima della normalizzazione.
9	Converte i logit in valori positivi che sommano a 1, interpretabili come distribuzione di probabilità.
10	Perché il modello stima plausibilità in base ai pattern appresi: non garantisce una corrispondenza con i fatti.

Glossario lampo

Termine	Definizione breve
Parametro	numero appreso: per esempio un peso o un bias

Forward propagation	calcolo dall'input verso l'output
Epoch	un passaggio completo sui dati di addestramento
Token	unità di testo elaborata dal modello
Embedding	rappresentazione vettoriale di un token/elemento
Logit	punteggio grezzo per un candidato
Softmax	trasformazione che produce una distribuzione
Transformer	architettura neurale basata soprattutto su attention e reti feed-forward
Inferenza	uso del modello addestrato per produrre un risultato

DA PORTARE CON TE

Una rete neurale non "memorizza regole" nel modo in cui le scriveremmo noi: distribuisce l'informazione in molti parametri e trasforma rappresentazioni numeriche. L'IA generativa moderna è molto più complessa della piccola rete dell'app, ma pesi, attivazioni, ottimizzazione e trasformazioni numeriche restano concetti fondamentali.

15 Esempi guidati: vedere la rete “al lavoro”

Questa sezione trasforma i concetti in situazioni concrete. Prima prova a prevedere il risultato, poi controlla il calcolo.

ESEMPIO 1 - Un peso può amplificare o attenuare

Supponiamo che un neurone riceva x = 0,8. Se il peso vale 0,2, il contributo è 0,8 × 0,2 = 0,16. Se il peso vale 1,5, lo stesso input contribuisce 1,20. L’informazione in ingresso non è cambiata: è cambiata l’importanza che la rete le attribuisce.

- Peso vicino a 0 → l’ingresso conta poco.
- Peso grande in valore assoluto → l’ingresso può contare molto.
- Peso negativo → l’ingresso spinge il neurone nella direzione opposta.

Input x	Peso w	Contributo x·w
0,8	0,2	0,16
0,8	1,5	1,20
0,8	-1,0	-0,80

ESEMPIO 2 - Il bias come “soglia spostata”

Immagina un sensore che debba attivarsi solo quando il segnale totale è abbastanza forte. Con input ponderati che sommano 1,2 e bias -1,0 otteniamo z = 0,2: il neurone supera di poco lo zero. Con lo stesso input e bias -2,0 otteniamo z = -0,8: la risposta cambia. Il bias non è un input aggiuntivo del mondo esterno: è un parametro interno appreso.

ESEMPIO 3 - Pesi positivi e negativi: un filtro antispam

Esempio puramente didattico: un neurone riceve tre indicatori. “Contiene la parola GRATIS” vale 1 con peso +1,4; “mittente noto” vale 1 con peso -1,2; “molti punti esclamativi” vale 1 con peso +0,8. Il totale prima del bias è 1,4 - 1,2 + 0,8 = 1,0. Qui alcuni segnali aumentano il punteggio “spam”, altri lo riducono.

- Non significa che un vero filtro antispam usi solo queste tre regole.
- Serve a capire che una rete combina indizi che possono spingere in direzioni opposte.

ESEMPIO 4 - Stessa somma, funzione di attivazione diversa

Se z = -2, ReLU produce 0, mentre una Sigmoid produce un piccolo valore positivo (circa 0,12). Se z = 2, ReLU produce 2, mentre la Sigmoid produce circa 0,88. La funzione di attivazione decide come il neurone trasforma il totale z.

z	ReLU	Sigmoid (circa)	Che cosa osservare
-2	0	0,12	ReLU blocca i valori negativi
0	0	0,50	Sigmoid è a metà
2	2	0,88	Sigmoid comprime tra 0 e 1

ESEMPIO 5 - Forward propagation in due passaggi

Primo strato: $x_1 = 1$ e $x_2 = 0,5$. Un neurone nascosto ha pesi 0,6 e 0,8, bias -0,2. Quindi $z = 1 \cdot 0,6 + 0,5 \cdot 0,8 - 0,2 = 0,8$. Se usiamo ReLU, $h = 0,8$. Secondo strato: il neurone di output riceve h con peso 1,5 e bias -0,4: $z\_out = 0,8 \cdot 1,5 - 0,4 = 0,8$. Il segnale è stato trasformato due volte.

15.1 AND, OR e XOR con numeri concreti

Le porte logiche sono esempi giocattolo, ma sono utilissime per vedere come pesi e bias implementano regole.

AND - "Devono essere vere entrambe"

Usiamo una funzione a soglia: output = 1 se $z > 0$, altrimenti 0. Scegliamo $w_1 = 1$, $w_2 = 1$ e bias $b = -1,5$. Il neurone calcola $z = x_1 + x_2 - 1,5$.

x_1	x_2	Calcolo di z	Output	Perché
0	0	0+0-1,5 = -1,5	0	nessun input è attivo
0	1	0+1-1,5 = -0,5	0	uno solo non basta
1	0	1+0-1,5 = -0,5	0	uno solo non basta
1	1	1+1-1,5 = +0,5	1	insieme superano la soglia

OR - "Ne basta almeno uno"

Manteniamo $w_1 = w_2 = 1$ ma cambiamo il bias in -0,5. Adesso basta un solo 1 per ottenere $z > 0$. Il confronto AND/OR mostra bene che cambiare il bias modifica la regola decisionale anche con gli stessi input e gli stessi pesi.

XOR - "Esattamente uno dei due"

XOR vale 1 per (1,0) e (0,1), ma vale 0 per (0,0) e (1,1). Un singolo neurone a soglia non riesce a tracciare una sola frontiera che separi correttamente questi quattro casi. Lo strato nascosto può invece costruire due indizi intermedi: OR e NAND.

x_1	x_2	h_1 = OR	h_2 = NAND	AND(h_1,h_2) = XOR
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

XOR nella vita quotidiana: “una luce con due interruttori”

In un circuito a due deviatori, semplificando l’idea, la luce può cambiare stato quando cambia uno dei due interruttori. XOR è utile per pensare alla regola “uno sì e l’altro no”. Non è una descrizione completa dell’impianto elettrico: è un’analogia logica per ricordare il pattern 0-1-1-0.

15.2 Come avviene l’apprendimento: un micro-esempio numerico

Non serve fare calcolo differenziale per intuire il meccanismo: il parametro viene corretto nella direzione che riduce l’errore.

ESEMPIO 6 - Un solo peso che impara

Modello semplificato: $\hat{y} = w \cdot x$. Vogliamo che, quando $x = 1$, l’output sia $y = 1$. Partiamo da $w = 0{,}20$: $\hat{y} = 0{,}20$, quindi siamo lontani dal target. Dopo un aggiornamento il peso può diventare, per esempio, 0,36; poi 0,49; poi 0,59... Ogni passo riduce l’errore. Nelle reti reali l’aggiornamento è calcolato usando il gradiente e avviene contemporaneamente su moltissimi parametri.

Passo	Peso w	Output $\hat{y}$	Errore $\lvert 1-\hat{y} \rvert$
inizio	0,20	0,20	0,80
dopo 1 update	0,36	0,36	0,64
dopo 2 update	0,49	0,49	0,51
dopo 3 update	0,59	0,59	0,41

ATTENZIONE - “Impara” non significa “capisce come una persona”

Nel linguaggio tecnico diciamo che la rete “impara” perché modifica i parametri in modo da ridurre una funzione di errore sui dati. Questa parola non implica coscienza, intenzioni o comprensione umana.

15.3 Dalla rete neurale alla generazione di testo

I numeri seguenti sono inventati a scopo didattico: servono a visualizzare il flusso, non a imitare un modello commerciale reale.

ESEMPIO 7 - "Il gatto beve ..."

Dopo aver elaborato il contesto, il modello assegna punteggi a moltissimi token candidati. In una simulazione possiamo immaginare che "acqua" riceva un logit più alto di "latte", "sedia" o "blu". La softmax trasforma questi punteggi in probabilità. Il token con probabilità maggiore non è necessariamente scelto sempre: dipende anche dalla strategia di campionamento.

Token candidato	Punteggio didattico	Probabilità illustrativa	Commento
acqua	3,2	≈ 82%	coerente con il contesto e il mondo reale
latte	1,5	≈ 15%	associazione possibile ma meno appropriata
sedia	-0,4	≈ 2%	grammaticalmente possibile come token, semanticamente debole
blu	-1,2	≈ 1%	molto poco coerente

ESEMPIO 8 - "La capitale d'Italia è ..."

Un modello può assegnare una probabilità molto alta a "Roma" perché quella continuazione è fortemente supportata dai pattern appresi. Ma il meccanismo resta probabilistico: non equivale a consultare automaticamente un archivio di fatti verificati. Per dati importanti occorre verificare le fonti.

ESEMPIO 9 - Temperatura bassa e alta

Immagina probabilità iniziali: Roma 0,90; Milano 0,05; Torino 0,03; Napoli 0,02. Con temperatura più bassa la distribuzione tende a concentrarsi ancora di più sul candidato dominante; con temperatura più alta tende ad appiattirsi. La temperatura non "rende il modello più intelligente": modifica quanto la scelta è conservativa o varia.

15.4 Attention: un esempio linguistico semplice

Nei Transformer, l'attention permette a ogni posizione di combinare informazione proveniente da altre posizioni in funzione del contesto.

ESEMPIO 10 - A che cosa si riferisce "era stanco"?

Frase: "Il cane che inseguiva i gatti era stanco". Per interpretare "era stanco", il modello deve usare il contesto e collegare fortemente quella parte alla rappresentazione di "cane", non semplicemente alla parola più vicina "gatti". L'attention non è una regola grammaticale esplicita: è un meccanismo numerico che attribuisce coefficienti diversi alle relazioni tra posizioni.

ESEMPIO 11 - Una stessa parola cambia rappresentazione

"Banca" in "ho depositato denaro in banca" e in "una banca di sabbia" parte dallo stesso token o da token simili, ma il contesto porta a rappresentazioni interne diverse. Questo aiuta a capire perché nei modelli moderni non si può pensare a un neurone = una parola = un significato fisso.

15.5 Errori, generalizzazione e overfitting

Una rete non deve soltanto ricordare gli esempi visti: deve funzionare anche su casi nuovi.

ESEMPIO 12 - Memorizzare non basta

Immagina di addestrare un classificatore di foto di cani usando solo immagini di cani su prati verdi. Se il modello associa troppo "verde = cane", può fallire davanti a un cane sul divano. Ha imparato una correlazione accidentale invece di una caratteristica utile. Questo è un modo intuitivo per introdurre overfitting e bias nei dati.

PROVA TU - Quattro domande da usare con la web app

1) Se raddoppi un peso positivo, che cosa ti aspetti che accada al contributo di quella connessione? 2) Se il bias diventa molto negativo, il neurone tenderà ad attivarsi più facilmente o più difficilmente? 3) Perché XOR è un buon esempio per mostrare l'utilità di uno strato nascosto? 4) Durante la generazione di una risposta, cambiano soprattutto i pesi o le attivazioni?

16 Nota di trasparenza sull’impiego dell’IA

DICHIARAZIONE

Ideazione didattica e progettuale: Prof. Erasmo Modica · Realizzazione tecnica: ChatGPT (OpenAI), sulla base delle indicazioni dell’autore. L’impiego dell’IA è dichiarato per scelta di trasparenza; l’app è progettata anche come strumento di AI literacy, in coerenza con le finalità richiamate dall’art. 4 del Regolamento (UE) 2024/1689 — AI Act.

Questa dispensa accompagna Neural Network Explorer. Le simulazioni numeriche relative alla generazione linguistica hanno finalità didattica e non rappresentano né espongono i parametri interni reali di specifici modelli commerciali.