

Java Object Oriented Neural Engine

**Un progetto Open Source nel campo delle
Reti Neurali Artificiali**

Paolo Marrone
paolo@marrone.org

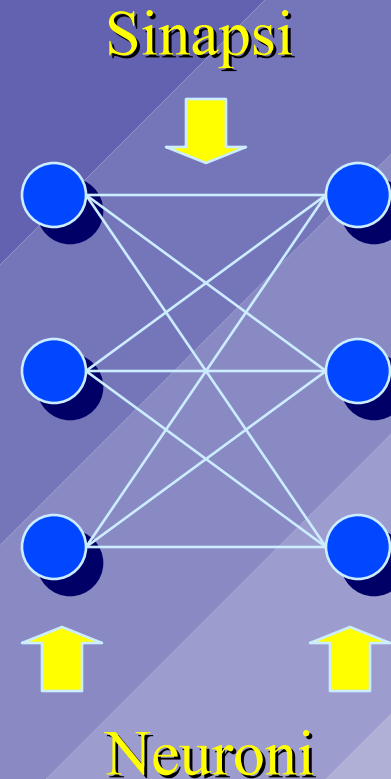
JugRoma 25 Gennaio 2006

Indice

- **Le Reti Neurali:** Un pò di teoria
- **Joone:** Cos'è e come funziona
- **Demo:** Un pò di pratica

Le Reti Neurali

Una rete neurale è composta da elementi di elaborazione (**neuroni**), connessi tra di loro tramite collegamenti (**sinapsi**), in modo da formare uno schema assimilabile ad un semplicissimo tessuto nervoso

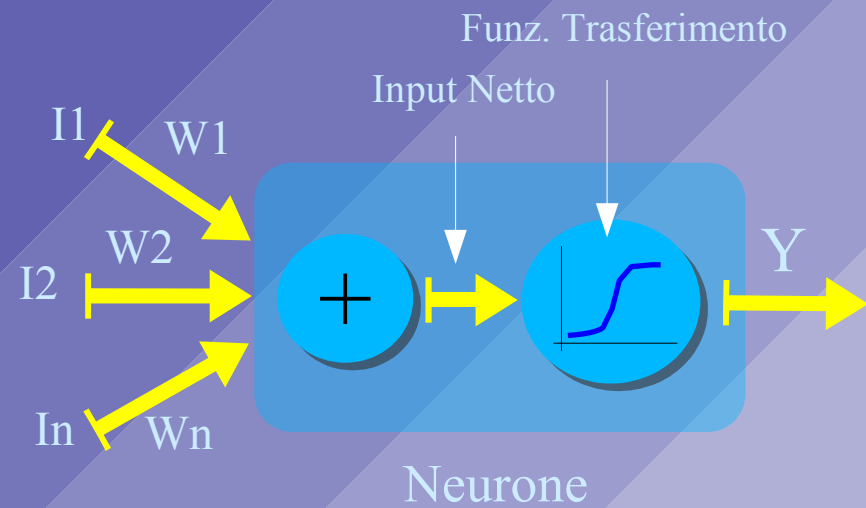


Le Reti Neurali

Ogni Neurone, attraverso le connessioni, riceve in input il segnale da ogni neurone dello strato precedente

Gli input, così pesati, vengono sommati, ed al risultato finale (l'input netto), viene applicata una funzione di trasferimento

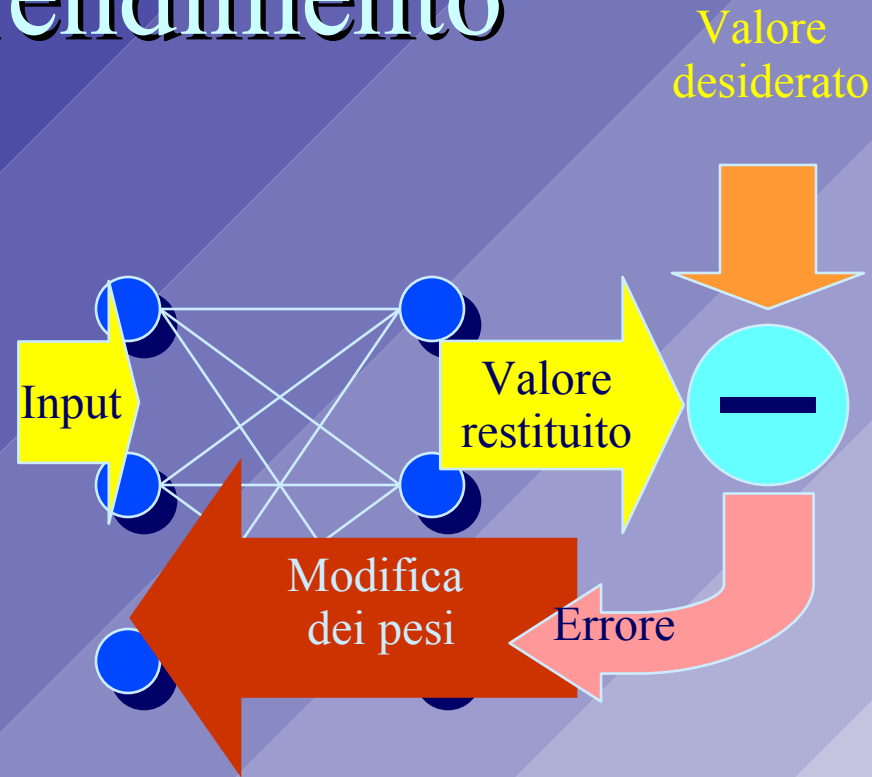
Il valore di output così calcolato viene presentato in ingresso ai neuroni dello strato successivo... e così via.



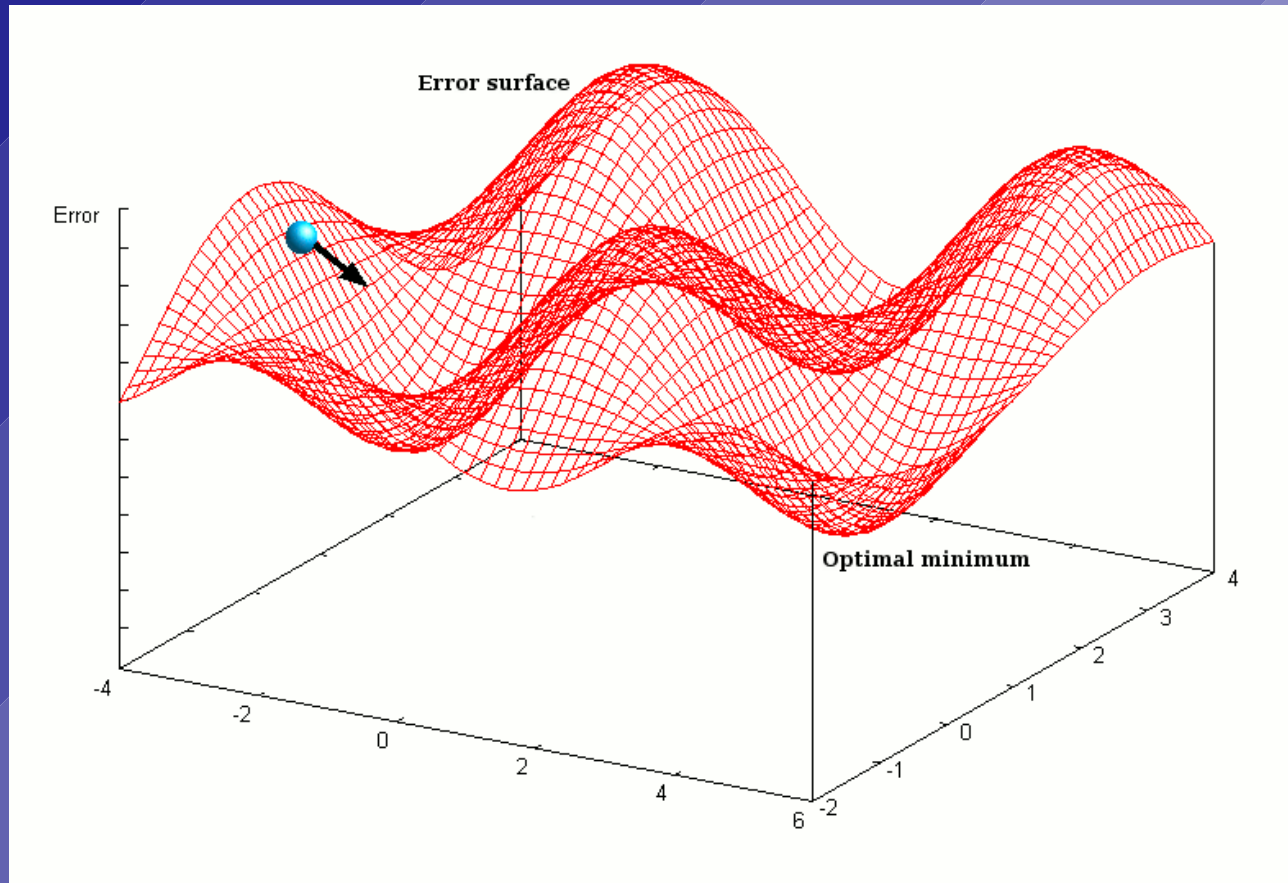
La fase di apprendimento

Una rete neurale è in grado di imparare a partire dagli esempi che gli vengono proposti. L'addestramento avviene misurando ad ogni ciclo l'errore commesso, espresso come funzione della distanza tra il valore desiderato e quello restituito dalla rete.

L'errore viene reimpresso nella rete al contrario (partendo dai neuroni di uscita) e retropropagato all'indietro, modificando lungo il passaggio i valori delle connessioni secondo precise formule matematiche (backpropagation)



La fase di apprendimento



Caratteristiche principali

- **Categorizzazione:** Una rete neurale è in grado di classificare gli esempi in ingresso (patterns) 'estraendo' le caratteristiche peculiari dai dati usati per il training
- **Generalizzazione:** Una rete neurale è in grado di fornire risposte corrette anche con dati di ingresso diversi rispetto a quelli usati per il training
- **Robustezza:** Una rete neurale è in grado di fornire comunque una risposta corretta anche in presenza di dati incompleti o parzialmente errati

Possibili Applicazioni

- **Banking/Finanza**
 - Scoring per l'assegnazione del credito
 - Analisi di portafoglio
 - Previsione andamento dei mercati
- **Assicurazioni**
 - Valutazione del rischio
 - Individuazione frodi
- **Industria**
 - Controllo di qualità
 - Previsioni di vendita
- **Medicina**
 - Supporto alla diagnosi
 - Prevenzione
- ...

Joone: Obiettivi Iniziali

- **Joone è nato con lo scopo di...**
 - Creare un framework per reti neurali Open Source, disponibile liberamente a tutti
 - Creare un prodotto innovativo sfruttando una architettura modulare e scalare
 - Utilizzare tecnologie standard ed aperte
 - Rappresentare il fulcro di nuovi progetti che implementino prodotti per i più svariati campi applicativi; in altre parole:

Costruire le basi per una diffusione su larga scala delle tecnologie basate sulle Reti Neurali

Moduli

- Joone è composto da:
 - Un **core engine** per la costruzione di reti neurali embedded in altre applicazioni
 - Un **editor visuale** per la creazione, addestramento e test di reti neurali attraverso una interfaccia grafica user-friendly
 - Un **ambiente di training distribuito** basato su tecnologia Jini, con il quale è possibile addestrare in parallelo diverse reti neurali su di un numero illimitato di macchine connesse in LAN

Training distribuito: Perché?

- Capacità di trovare una buona soluzione attraverso l'esplorazione in parallelo di diverse possibilità
- Permette di superare i limiti di performance di un linguaggio interpretato come java
- Altamente scalabile, sia in verticale (su diverse CPU), sia in orizzontale (su diverse macchine)
- Permette di applicare alle Reti Neurali tecniche evolutive di ottimizzazione (p.es. Algoritmi Genetici)

Un po' di pratica...

- **Costruiremo due tipologie di reti neurali:**
 - Supervised Neural Network: utilizzando l'algoritmo di backpropagation addestreremo una rete neurale a tre strati a risolvere la funzione logica XOR
 - Unsupervised Neural Network: impiegando una rete non supervisionata realizzeremo un riconoscitore di caratteri scritti a mano

La funzione XOR

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	0

