

# RETI NEURALI

Esperimento di Guidelli Michael

## Abstract

In questa relazione sarà analizzato il concetto di ***Rete Neurale***, e verrà dimostrato un esperimento di classificazione di un dataset contenente due diverse razze di gatti. Utilizzando una rete composta da tre neuroni (due di ingresso, uno di uscita), con apprendimento supervisionato. L'obiettivo è quello di addestrare la rete a riconoscere le diverse razze di gatti e calcolare la performance della rete intesa come precisione della stessa.

# Indice

<b>1</b>	<b>Introduzione alla Rete Neurale</b>	<b>3</b>
1.1	Come funziona una Rete Neurale? . . . . .	3
1.2	Quali problemi risolve una Rete Neurale? . . . . .	3
1.3	Come apprendono le Reti Neurali? . . . . .	4
<b>2</b>	<b>Com'è composta una Rete Neurale?</b>	<b>5</b>
<b>3</b>	<b>Come si addestra una Rete Neurale?</b>	<b>6</b>
<b>4</b>	<b>Funzioni matematiche utilizzate</b>	<b>7</b>
<b>5</b>	<b>Aree di applicazione delle Reti Neurali</b>	<b>8</b>
<b>6</b>	<b>Esperimento e analisi dei risultati</b>	<b>9</b>
6.1	Tabella dei test effettuati: . . . . .	10
6.2	Dataset utilizzati: . . . . .	11
6.3	Affidabilità della rete . . . . .	12
<b>7</b>	<b>Conclusioni e ipotesi di lavoro future</b>	<b>13</b>
<b>8</b>	<b>Bibliografia e sitografia</b>	<b>14</b>
<b>9</b>	<b>Strumenti</b>	<b>14</b>

# Lista delle figure

1	Rappresentazione di una rete neurale composta da tre neuroni.	5
2	Rappresentazione di una rete neurale con tre neuroni con l'implementazione dell'algoritmo Back-Propagation. . . . .	6
3	Tabella dei test effettuati. . . . .	10
4	Rappresentazione grafica dei dataset . . . . .	11
5	Istogramma affidabilità della rete. . . . .	12

# 1 Introduzione alla Rete Neurale

Le reti neurali sono sistemi informatici con nodi interconnessi che funzionano in modo molto simile ai neuroni nel cervello umano. Utilizzando algoritmi, possono riconoscere modelli nascosti e correlazioni nei dati grezzi, raggrupparli e classificarli e, nel tempo, apprendere e migliorare continuamente.

## 1.1 Come funziona una Rete Neurale?

Nel primo strato la rete neurale riceve i dati sotto forma di input, attivandosi elabora i dati secondo la capacità di classificazione per cui è stata addestrata, e passa le informazioni ottenute allo strato successivo per propagazione. Lo stesso procedimento viene ripetuto per il totale di epoche scelto, le informazioni di partenza vengono perfezionate secondo le interpretazioni dei diversi nodi e secondo le modalità per cui sono stati programmati per attivarsi. Infine i dati elaborati giungono allo strato di uscita, che raccoglie i risultati, li modella per poi passarli al blocco successivo, fino alla rappresentazione della soluzione definitiva al problema sottoposto.

## 1.2 Quali problemi risolve una Rete Neurale?

Una rete neurale può essere utilizzata per risolvere tre tipi di problemi:

1. **Classificazione:** in cui la variabile da prevedere è di tipo categorico, cioè una classe; (es. abbiamo un insieme di immagini di gatti e le vogliamo usare per riconoscere i gatti in altre foto)
2. **Regressione:** in cui la variabile da prevedere è di tipo continuo, cioè un numero; (es. abbiamo lo storico del valore di FIAT in borsa e vogliamo utilizzarlo per prevedere il valore futuro)
3. **Clustering:** problema molto simile alla classificazione ma in cui non abbiamo a disposizione delle classi; (es. abbiamo un insieme di immagini di tumori, vogliamo riconoscere quali sono maligni e quali benigni).

### 1.3 Come apprendono le Reti Neurali?

Gli algoritmi di apprendimento delle reti neurali possono essere suddivisi in tre categorie:

1. **Apprendimento supervisionato:** la rete analizza un training set (cioè un insieme di input ai quali corrispondono output noti) e apprende come calcolare nuove regole di associazione input-output, processando input esterni al training set;
2. **Apprendimento non supervisionato:** la rete riceve solo un insieme di variabili di input ed elabora dei cluster rappresentativi per categorizzarle;
3. **Apprendimento per rinforzo:** la rete apprende esclusivamente dall'interazione con l'ambiente, attraverso un circuito di incentivi e disincentivi per ottenere il risultato desiderato.

Non esiste un metodo di apprendimento migliore di un altro, ma la scelta dipende dalla tipologia di dati a disposizione e dai risultati che si vogliono ottenere. Da questo punto di vista è molto importante la preparazione e l'esperienza dell'operatore che implementa il modello.

## 2 Com'è composta una Rete Neurale?

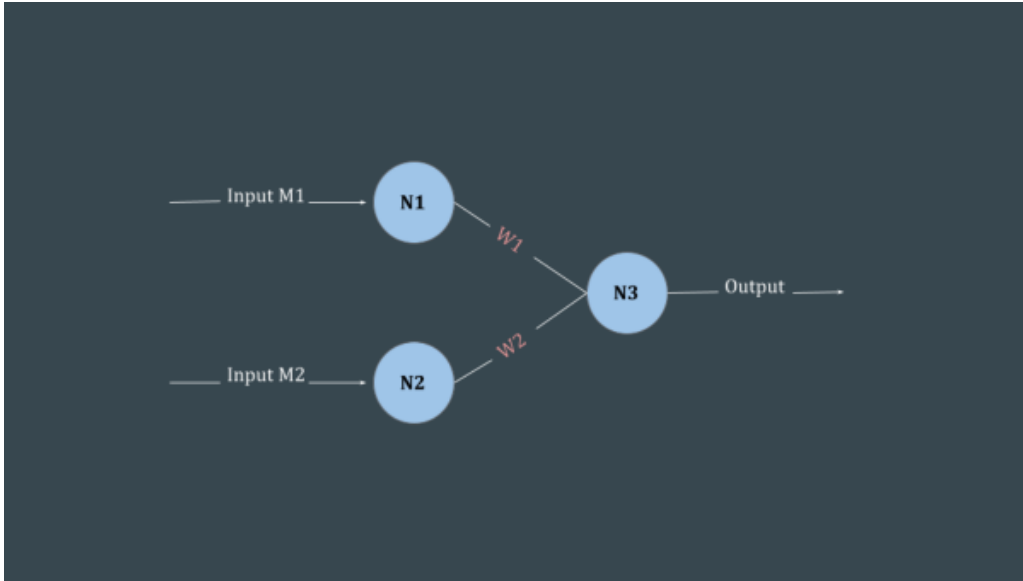


Figure 1: Rappresentazione di una rete neurale composta da tre neuroni.

La rete neurale utilizzata [Figura[1]] sarà composta da diversi elementi, essendo una rete neurale molto semplice non disporrà dei livelli nascosti ma solo di uno strato di input ed uno di output, evidenziamo gli elementi:

- **M1 e M2:** rappresentano i dati ricavati dal dataset, che andranno inseriti nei neuroni di input;
- **N1 e N2:** rappresentano i neuroni di ingresso;
- **W1 e W2:** rappresentano i pesi dei collegamenti tra i neuroni, che verranno utilizzati per calcolare tramite funzioni matematiche il nostro output;
- **N3:** rappresenta il nostro neurone di uscita.

### 3 Come si addestra una Rete Neurale?

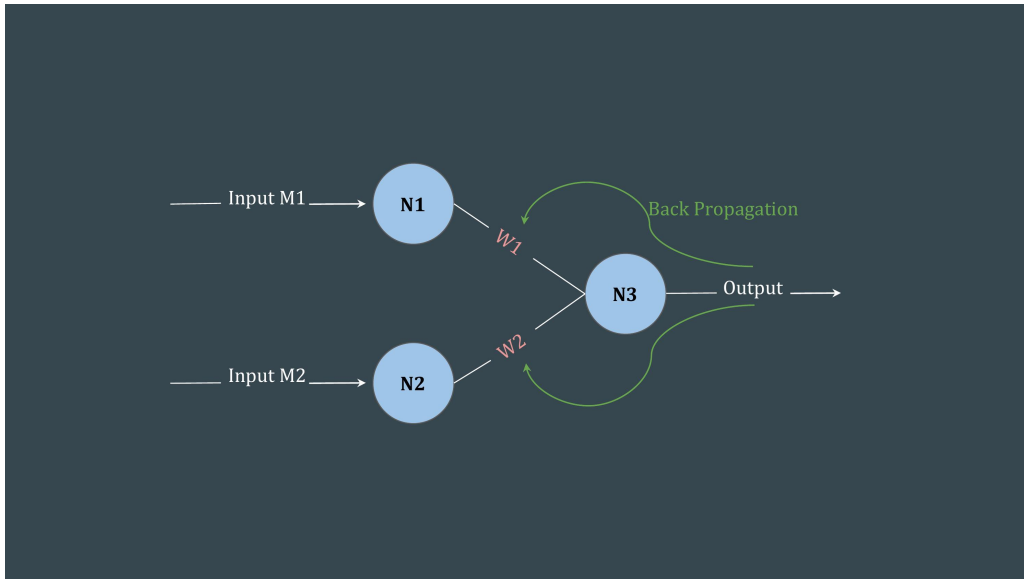


Figure 2: Rappresentazione di una rete neurale con tre neuroni con l'implementazione dell'algoritmo Back-Propagation.

L'algoritmo che addestra la rete neurale è chiamato **Back-Propagation** [figura[2]]. Ovvero l'azione di prendere il risultato di ogni epoca e usarlo per modificare pesi e bias, questa operazione si ripeterà fino all'ottenimento di una previsione con il minor margine di errore possibile ed è la caratteristica che rende le reti neurali non deterministiche.

Addestrando la rete potrebbe essere inefficace se allenata con un dataset insufficiente quindi non genererebbe una soluzione, invece con un dataset eccedente riuscirebbe a generare una soluzione solo per quel determinato dataset.

## 4 Funzioni matematiche utilizzate

- **Funzione di Costo:** ha in input i dati e la previsione per procurarci l'errore commesso, più il valore in uscita sarà vicino allo zero più accurata sarà la previsione. la funzione di costo non fa altro che calcolare l'errore quadratico.
- **Errore Quadratico:** non è altro che l'elevazione al quadrato della sottrazione tra previsione e l'obiettivo da raggiungere.
- **Calcolo delle Derivate:** la modifica dei pesi avviene attraverso il calcolo delle derivate, trovando il punto minimo della funzione di costo e sottraendo ad ogni peso soltanto una piccola percentuale, ottenendo la seguente formula matematica:

$$Peso = \text{Peso} - \text{Piccola Percentuale} * (\text{Funzione di Costo} / \text{Peso}) \quad (1)$$

- **Funzione di attivazione:** ha l'obiettivo di non rendere lineare la rete neurale e comprimere i valori dell'output in un intervallo controllato. In questa rete verrà utilizzata la funzione sigmoideale per l'attivazione. Formula:

$$\text{Sigmoide}(\text{M1} * \text{W1} + \text{M2} * \text{W2} + \text{BIAS}) \quad (2)$$

## 5 Aree di applicazione delle Reti Neurali

Le reti neurali possono essere utilizzate con successo soprattutto se è disponibile una grande quantità di dati da valutare e solo una scarsa conoscenza sistematica della soluzione. I casi di applicazione classici sono il riconoscimento di testo, immagini e voce, casi in cui i computer esaminano i dati per determinate caratteristiche al fine di effettuare un'assegnazione. I settori in cui l'utilizzo delle reti neurali rappresenta una realtà affermata sono numerosi ed ormai è quasi impossibile elencarli tutti. I principali sono:

- Finanza, con numerose applicazioni: previsioni sull'andamento dei mercati inclusi quelli valutari, analisi del rischio di credito, analisi del portafoglio, etc;
- Riconoscimento ed elaborazione delle immagini;
- Analisi del parlato e riconoscimento vocale;
- Simulazione di sistemi biologici, da quelli intracellulari alle reti neurali;
- Diagnosi mediche, inclusi i referti di TAC e risonanze magnetiche;
- Robot steering (pilota automatico);
- Controllo di qualità su scala industriale;
- Data mining.



## 6 Esperimento e analisi dei risultati

L'obiettivo dell'esperimento è quello di individuare a quale razza appartenga ogni gatto all'interno del dataset e di calcolare l'affidabilità della rete. In questo caso nel dataset saranno presenti 10 gatti appartenenti a due razze diverse quali: gatti della giungla e gatti della sabbia. Utilizzando una rete neurale con tre neuroni [Figura[1]]. L'esperimento si dividerà in due fasi:

1. Fase di allenamento, in questa fase la rete verrà addestrata ad individuare a quale razza appartenga ogni gatto. Sarà utilizzato un dataset contenente lunghezza, peso e obiettivo per ogni gatto al suo interno. Per addestrare la rete verranno utilizzate 2280 epoche e un learning rate pari allo 0.1%.
2. Utilizzo della rete, completato l'addestramento e congelati i pesi, la rete verrà testata con un dataset simile a quello di addestramento, i dati all'interno del dataset saranno deviati di circa l'1%. Infine si calcolerà l'affidabilità della rete in percentuale.

In quest'esperimento verranno effettuati cinque test utilizzando due diversi dataset, il primo ideato dal canale [Youtube Riccardo Talarico](#), il secondo molto simile a quello di Talarico, ideato da me. Talarico inoltre offre un codice Python già scritto e funzionante per testare ed utilizzare la rete:

[GitHub codice Talarico](#)

Dopodiché ho effettuato un refactoring completo del codice ed ho implementato la possibilità di generare un istogramma per facilitare la visualizzazione dei risultati dei test. inoltre introdurre nel codice la manipolazione dei file per addestrare e testare la rete utilizzando i file CSV (Comma Separated Value).











[GitHub codice refactored](#)

## 6.1 Tabella dei test effettuati:.

Dataset Allenamento	Dataset Esperimento	Affidabilità
Dataset Talarico	Dataset Talarico	100%
Dataset Talarico prima metà	Dataset Talarico seconda metà	80%
Dataset Test	Dataset Test	100%
Dataset Test prima metà	Dataset Test seconda metà	80%
Dataset Tutorial	Dataset Test	100%

Figure 3: Tabella dei test effettuati.

## 6.2 Dataset utilizzati:

Dataset Allenamento										
Razza										
Lunghezza	9	2	3.2	9.1	1.6	8.4	8	3.1	6.3	3.4
Peso	7.0	5.0	4.94	7.46	4.83	7.46	7.28	4.58	9.14	5.36











Dataset Test										
Razza										
Lunghezza	8	3	3.3	9	1.8	8.6	7.5	2.99	6.5	3.5
Peso	6.99	5.5	4.99	7.45	4.88	7	7.3	4.6	9	5.4

Figure 4: Rappresentazione grafica dei dataset

### 6.3 Affidabilità della rete

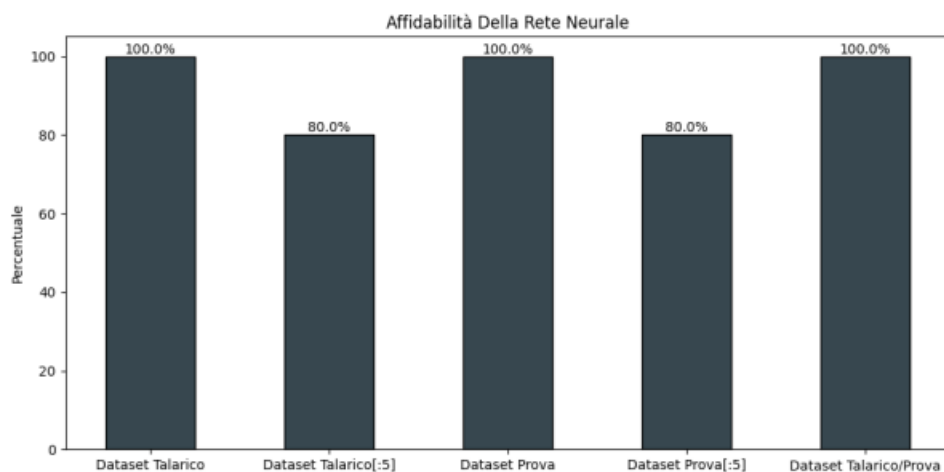


Figure 5: Istogramma affidabilità della rete.

Addestrando la rete utilizzando l'algoritmo di **Back-Propagation** [figura[2]] e la funzione Sigmoidale [Funzione[2]] per l'attivazione della rete, il codice genererà un'istogramma per facilitare la visualizzazione dell'affidabilità della rete.

Com'è possibile notare dall'istogramma la rete risulta efficace nei seguenti casi:

1. Allenamento: Talarico, Verifica: Talarico, Affidabilità: 100%
2. Allenamento: Talarico prima metà, Verifica: Talarico seconda metà, affidabilità: 80%
3. Allenamento: Test, Verifica: Test, affidabilità: 100%
4. Allenamento: Test prima metà, Verifica: Test seconda metà, Affidabilità: 80%
5. Allenamento: Talarico, Verifica: Test, Affidabilità: 100%

## 7 Conclusioni e ipotesi di lavoro future

In conclusione si può affermare che la rete sottoposta a dataset sconosciuti riesca a generare buoni risultati, solo se il dataset è abbastanza grande. Si può ipotizzare che utilizzando piccole quantità di dati la rete non riesca a generalizzare una soluzione, viceversa le troppe epoche o un dataset troppo grande potrebbe produrre buoni risultati solo su dataset conosciuti.

In futuro si potrebbe ampliare il dataset, diminuire il numero di epoche e implementare una funzione di attivazione diversa da quella Sigmoidale [Funzione[2]] e riscontrare le differenze tra le due reti.

## 8 Bibliografia e sitografia

1. [Introduzione alle Reti Neurali del canale Youtube Riccardo Talarico](#) - per il dataset e il codice.
2. Corso "Fondamenti delle Reti Neurali" del Prof. Arcella Andrea - Per le basi. sull'argomento.
3. [Documentazione Numpy](#) - Libreria di Python per il calcolo numerico.
4. [Documetazione Matplotlib](#) - Libreria di Python per la rappresentazione grafica.
5. [Documentazione CSV](#) - Libreria di Python per la manipolazione dei file.
6. [GeekForGeeks](#), [StackOverflow](#) - Per risoluzioni di problemi riscontrati durante la programmazione del codice.
7. [GitHub](#), [StackOverflow](#) - Profili personali.

## 9 Strumenti

1. [Python](#) - Linguaggio di Programmazione.
2. [Visual Studio Code](#) - Editor di Testo.
3. [GitHub](#) - Pubblicazione del codice.
4. [Google Presentazioni](#) - Per la realizzazione delle immagini.