

# Analisi della complessità di reti neurali generate tramite algoritmi genetici

Mattia Ceccarelli

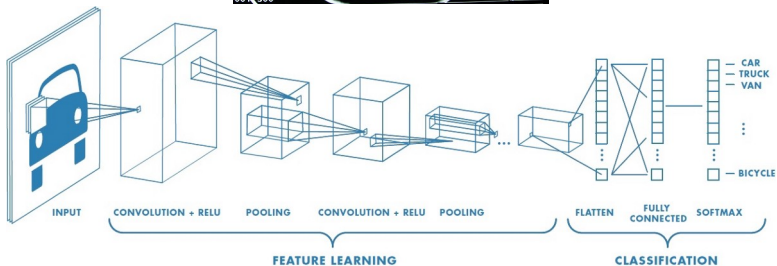
Corso di Laurea in Fisica  
Università di Bologna

A.A. 2017-2018

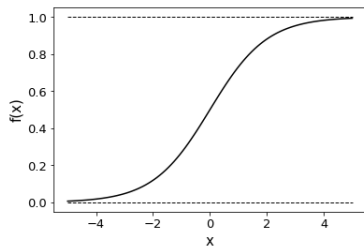
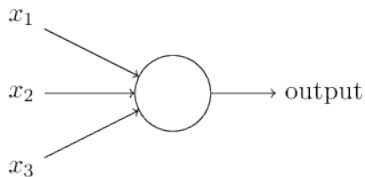
Relatore: Dott. Enrico Giampieri

Correlatore: Dott. Nico Curti

# Introduzione



# Reti Neurali: perceptron



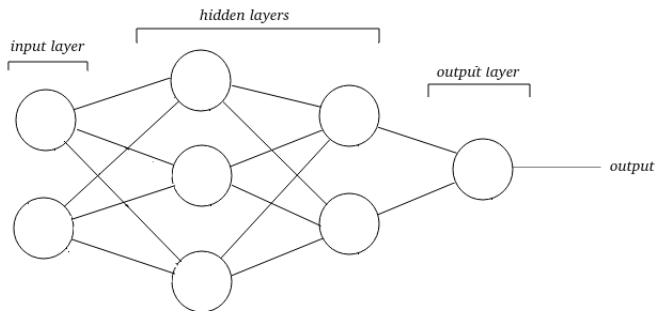
A sinistra, rappresentazione di un perceptron in cui:

$$output = f\left(\sum_{i=1}^n x_i w_i + bias\right)$$

A destra, la funzione di attivazione sigmoide definita come:

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

# Reti Neurali: MultiLayer Perceptron

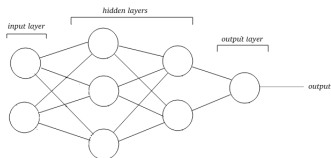


Un esempio di MultiLayer Perceptron semplice con 2 neuroni in input, 2 hidden layer con 3 e 2 neuroni ed un neurone in output. Il vettore di output del Layer  $i$ -esimo è dato da:

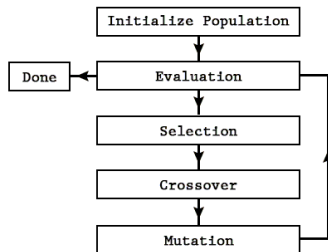
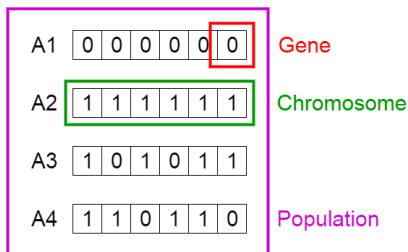
$$\mathbf{o}_i = f(W_{i,i-1} \cdot \mathbf{o}_{i-1} + \mathbf{b}_i)$$

Dove  $W_{ij}$  indica la matrice dei pesi tra il layer  $i$ -esimo e  $j$ -esimo

# Algoritmi Genetici



La rete precedente è identificata dal cromosoma [3,2]



# Funzione di fitness

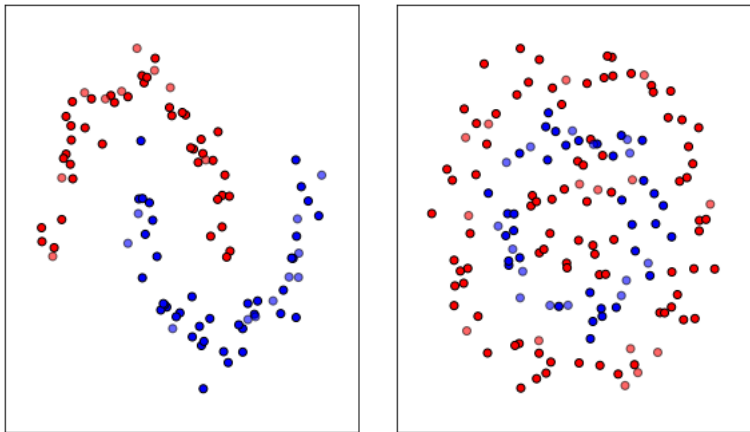
- la valutazione della rete avviene dopo l'addestramento, le due fasi avvengono in parti separate del dataset (train set e test set)
- Consente di selezionare gli individui migliori
- Logarithmic Loss function consente di valutare l'incertezza della rete. È definita come:

$$-\log P\left(\frac{y_t}{y_p}\right) = -(y_t \log(y_p) + (1 - y_t) \log(1 - y_p))$$

Dove:

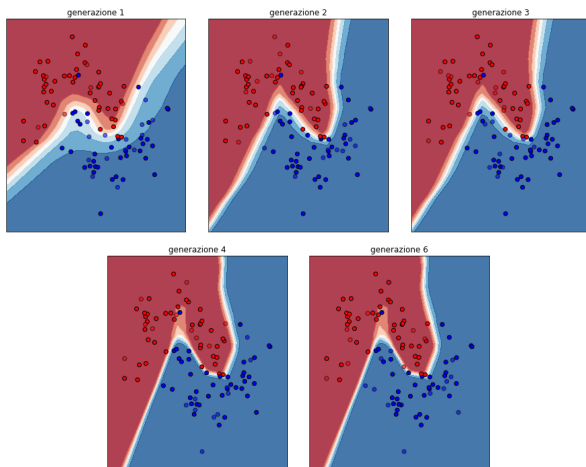
- $y_t$  Classificazione corretta (0 o 1)
- $y_p$  Probabilità stimata che  $y_t = 1$

## Dataset



A sinistra *Moons* e a destra *Circles+* con  $noise = 0.2$ . I dataset sono sintetici e il numero di punti è 100

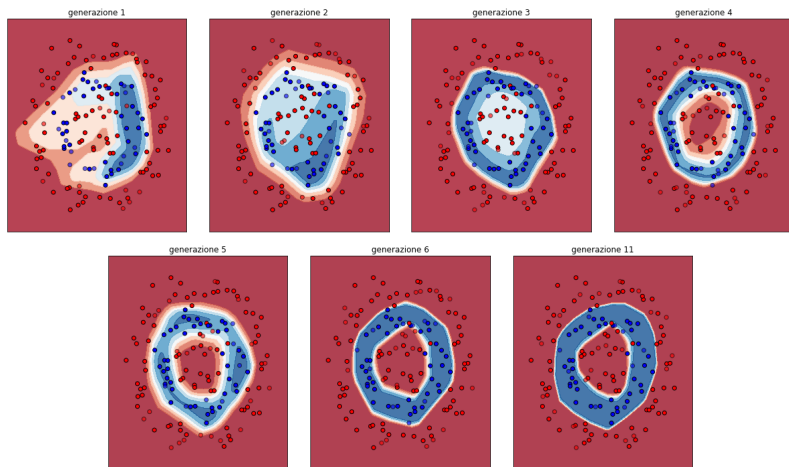
# Classificazione su *Moons*



Esempio di evoluzione nella classificazione di *Moons*



# Classificazione su *Circles+*



Esempio di evoluzione nella classificazione di *Circles+*

# Analisi della complessità

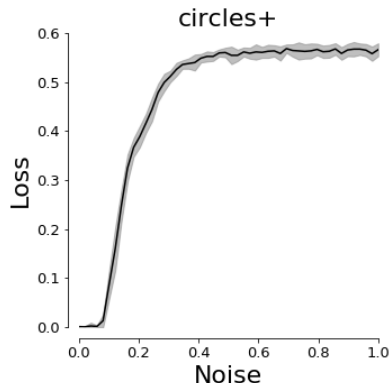
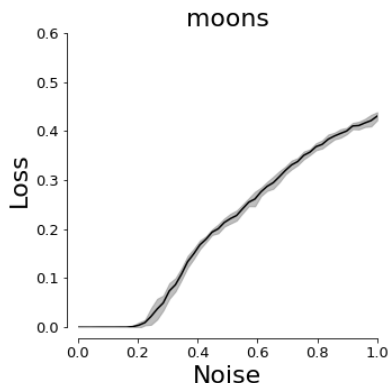
La rumorosità dei dataset modifica alcuni parametri della rete:

- Numero di collegamenti (Links)
- Lunghezza (o profondità) ossia il numero di hidden layer
- Numero di neuroni nel layer più piccolo

Per ognuno di essi sono state effettuate 70 run per dataset su 50 valori di noise compresi tra 0 e 1.

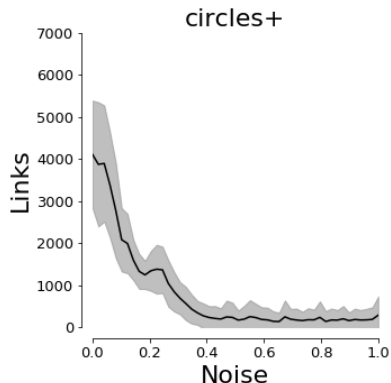
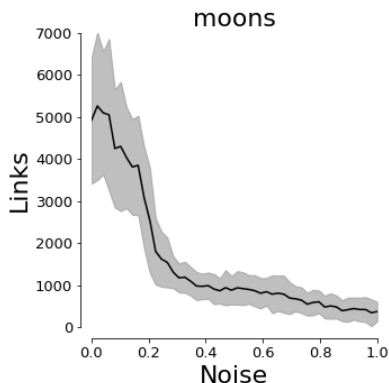
I grafici seguenti mostreranno media e deviazione standard.

## Loss su *Moons* e *Circles+*



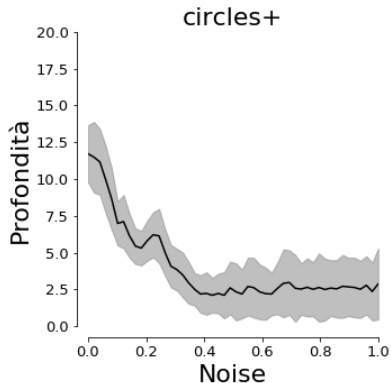
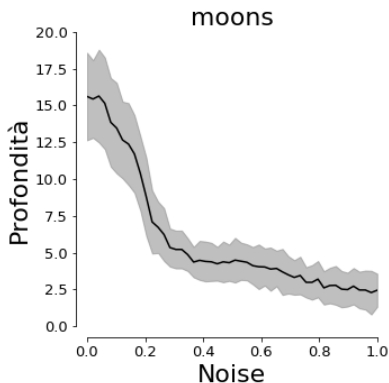
Andamento dello score in funzione del noise in *Moons* (a sinistra) e sul dataset *circles+* (a destra)

## Links su *Moons* e *Circles+*



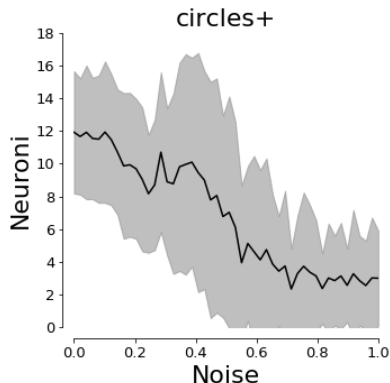
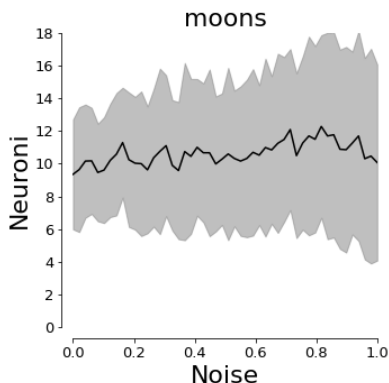
Numero di collegamenti in funzione del noise su *Moons* (a sinistra) e *Circles+* (a destra)

# Lunghezza



Numero di Hidden Layer della rete in funzione del noise su *Moons* (a sinistra) e *Circles+* (a destra)

## Layer più piccolo della rete



Numero di neuroni nel layer più piccolo della rete su *Moons* (a sinistra) e su *Circles+* (a destra) in funzione del rumore

# Conclusioni

- L'algoritmo è in grado di evolvere una popolazione e restituire una rete capace di classificare i dataset testati
- La separazione tra train set e test set tra addestramento e valutazione della rete risulta sufficiente a far sì che la complessità rimanga finita

# Sviluppi Futuri

- Applicazione a dataset reali
- Applicazioni a reti neurali a convoluzione
- Aggiunta nell'algoritmo di una penalità esplicita per reti troppo complesse e confronto dei risultati.



Grazie per l'attenzione.



# Contents