



LABORATORIO DI:

METODI E MODELLI MATEMATICI IN PYTHON

A CURA DI: **ANTONIO MIRARCHI** & GIUSEPPE TROTTA

<http://www.labmetodiemodelli.it/>

Dove eravamo rimasti?



Alberi Decisionali

IL PROGRAMMA

INTRODUZIONE A
PYTHON

01

LE STRUTTURE
DATI IN
PYTHON

02

LE LIBRERIE PER LA
DATA SCIENCE
(PARTE 2)
+ Test Intermedio

04

LE LIBRERIE PER LA
DATA SCIENCE
(PARTE 1)

03

LA DATA ANALYSIS
E LA DATA
VISUALIZATION

05

10

IL PROGRAMMA

COSTRUIRE MODELLI
PREDITTIVI (PARTE 1)



COSTRUIRE MODELLI
PREDITTIVI (PARTE 2) +
Test Intermedio



COSTRUIRE MODELLI
PREDITTIVI (PARTE 3)



COSTRUIRE MODELLI
PREDITTIVI (PARTE 4)



RETI NEURALI &
DEEP LEARNING
+ Test Intermedio



10

3° Test intermedio

Manuale di sopravvivenza al laboratorio



4 esercizi ne
bastano 2 svolti
correttamente



Argomenti lezioni
dalla 1 alla 9



Consegna via mail entro le ore 17:00

antonio.mirarchi@thinkopen.it

Nominare il file come:

Nome_Cognome_NumeroDiMatricola
nell'oggetto della mail inserite: [V13]

<https://www.labmetodiemodelli.it/>

1

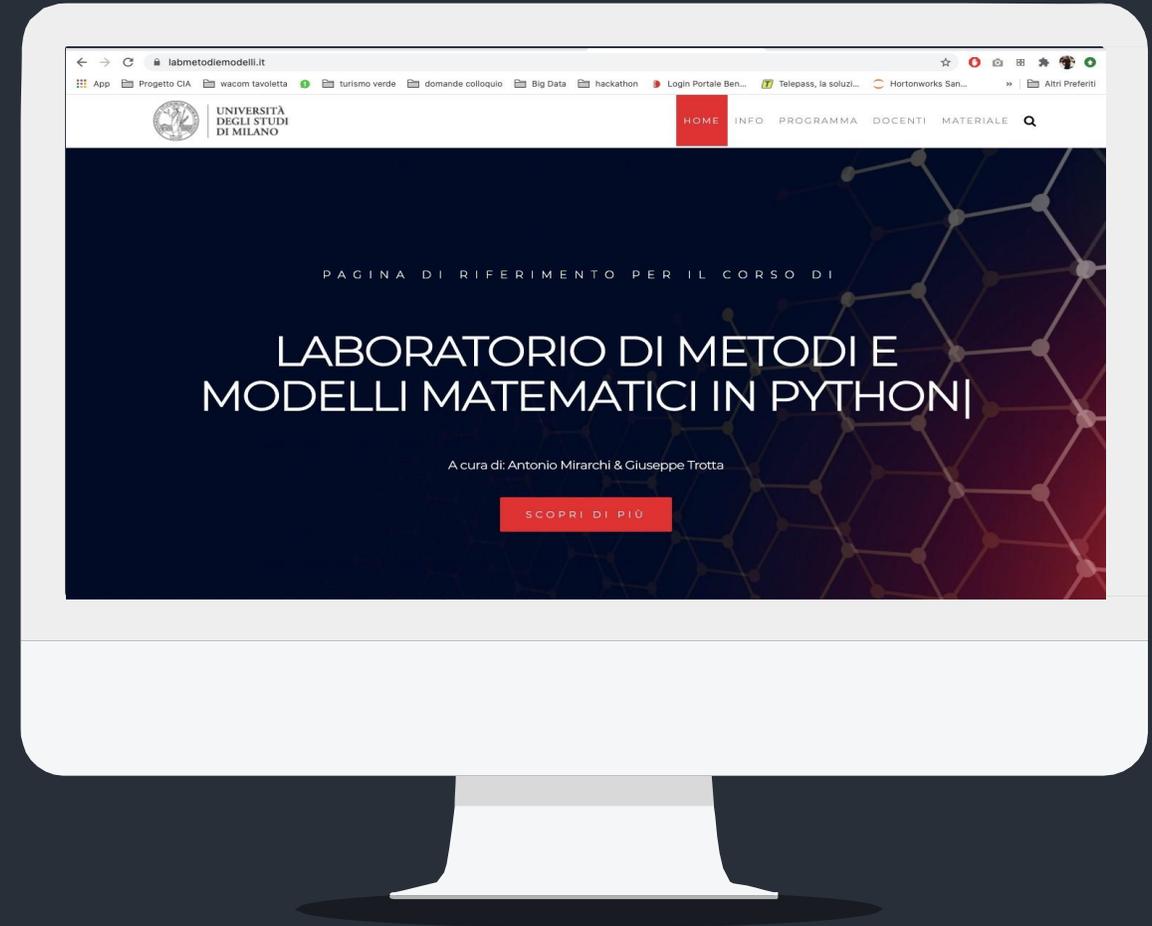
Random Forest

2

Le Reti Neurali

3

Use Cases





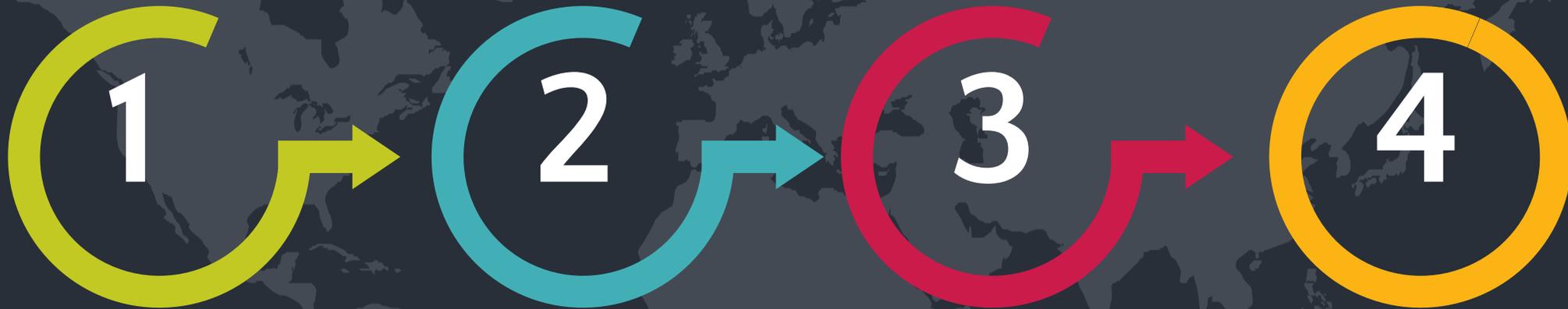
Bagging

Un metodo ensemble è una tecnica che combina le previsioni di più algoritmi di apprendimento automatico per fare previsioni più accurate rispetto a qualsiasi singolo modello.

Il bagging rappresenta una procedura generale che può essere utilizzata per ridurre la varianza di quegli algoritmi che hanno una varianza elevata

<https://www.labmetodiemodelli.it/>

Random Forest



Algoritmo di
Apprendimento
Supervisionato

Rappresenta un tipo
di modello
ensemble, che si
avvale del bagging
come metodo di
ensemble e l'albero
decisionale
come modello
individuale.

una foresta casuale
combina molti
alberi decisionali in
un unico modello.

Il risultato finale restituito dal
Random Forest altro non è che la
media del risultato numerico
restituito dai diversi alberi nel
caso di un problema di
regressione, o la classe restituita
dal maggior numero di alberi nel
caso la Random Forest sia stata
utilizzata per risolvere un
problema di classificazione.

Random Forest

Come Funziona



Elemento 1

Campionamento casuale di punti dati di allenamento durante la costruzione di alberi;

Elemento 2

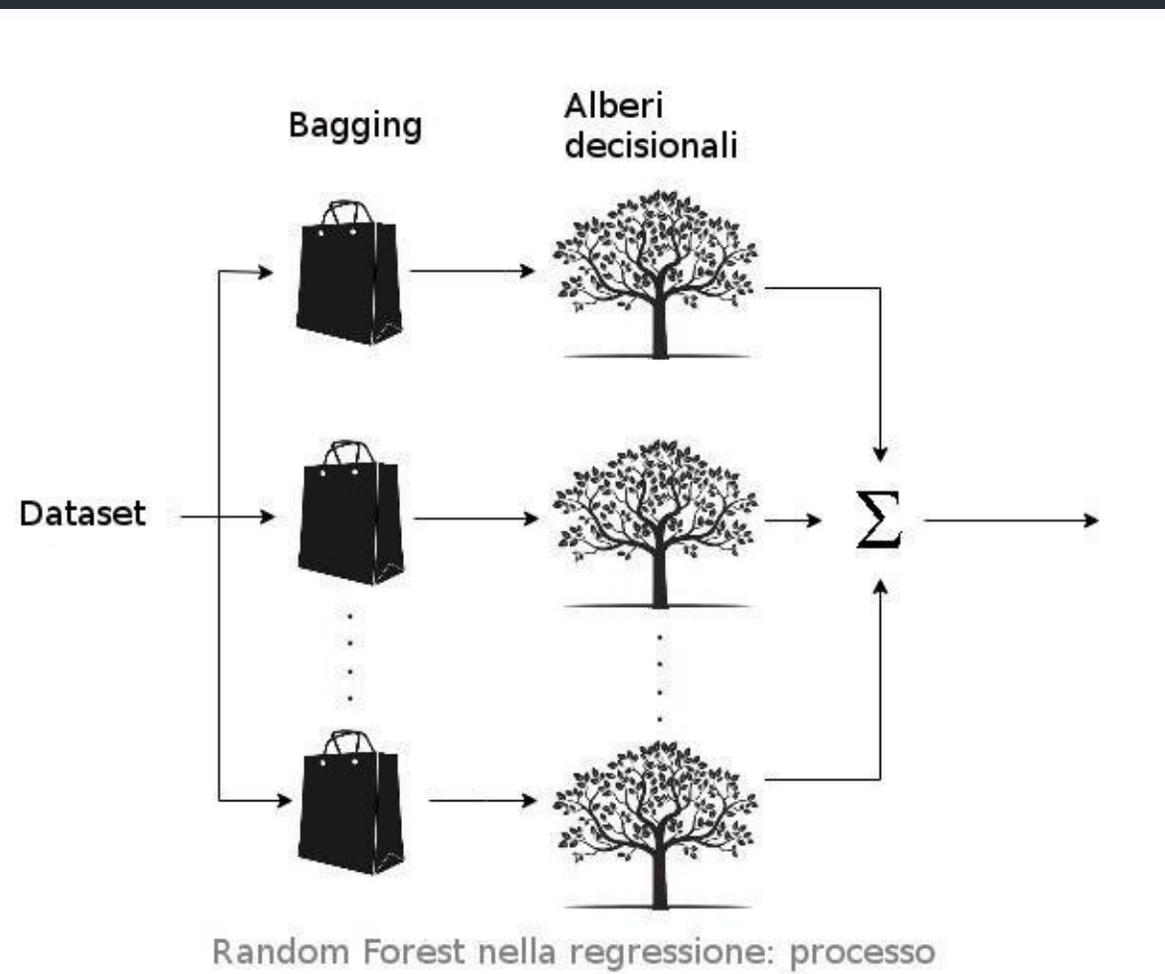
Sottoinsiemi casuali di funzionalità considerate durante la divisione dei nodi.

Campionamento Casuale Osservazioni di Allenamento

Durante l'allenamento, ogni albero in una foresta casuale impara da un **campione casuale di punti dati**. I campioni vengono disegnati con la sostituzione, nota come bootstrap, il che significa che alcuni campioni verranno utilizzati più volte in un singolo albero.

L'idea è che addestrando ciascun albero su campioni diversi, sebbene ogni albero possa presentare una varianza elevata rispetto a una particolare serie di dati di addestramento, nel complesso l'intera foresta avrà una **varianza inferiore** ma non a costo di aumentare la distorsione.

Al momento del test, le previsioni vengono effettuate calcolando la media delle previsioni di ciascun albero decisionale



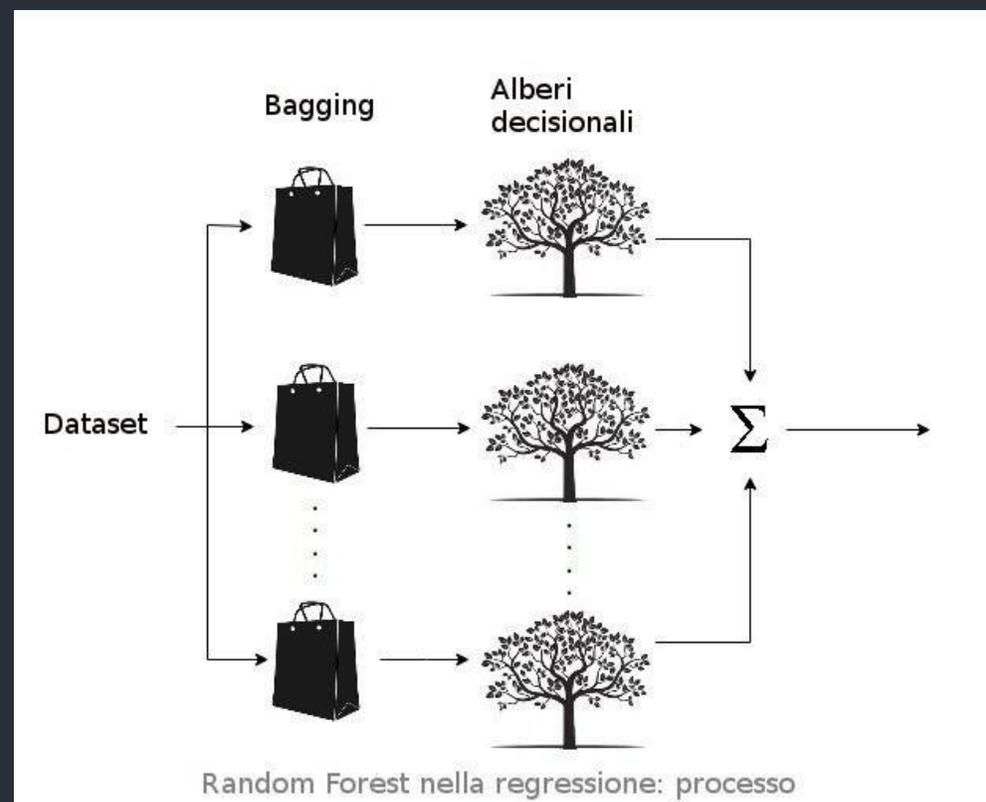
Sottoinsiemi casuali di caratteristiche per la divisione dei nodi

viene considerato solo un sottoinsieme di tutte le caratteristiche per suddividere ciascun nodo in ciascun albero decisionale.

Generalmente questo è impostato su $\sqrt{n_features}$ per la classificazione, il che significa che:

se ci sono 16 caratteristiche in ciascun nodo in ogni albero, saranno prese in considerazione solo 4 caratteristiche casuali per dividere il nodo (la foresta casuale può anche essere addestrata considerando tutte le features di ogni nodo come è comune nella regressione).

In pratica, la foresta casuale combina centinaia o migliaia di alberi decisionali, si allena ciascuno su un insieme leggermente diverso di osservazioni, suddividendo i nodi in ciascun albero considerando un numero limitato di caratteristiche.



Applicazioni



1

Bancario: Fraoud Management

2

Medicina: ad esempio valutare gli effetti di medicinali assunti con altri medicinali

3

Mercato Azionario: previsioni azionarie ed andamenti del mercato

4

E-commerce: suggerimento prodotti

5

Telco: tutto quello che vi abbiamo raccontato finore



FIFA WORLD CUP
RUSSIA 2018

Let's Study

Reti Neurali

Un viaggio nella mente artificiale

1

Intro e Breve Storia

2

Principio di funzionamento

3

Applicazioni



Reti Neurali



1

è un modello matematico composto da neuroni artificiali di ispirazione alle reti neurali biologiche

2

modelli costituiti da almeno due strati, uno strato di ingresso e uno di uscita, e di solito anche da ulteriori strati intermedi

3

Più complesso è il problema da risolvere con la rete neurale artificiale, più strati sono necessari

Tipi di Reti Neurali



1

PERCETTRONE

2

FEED
FORWARD

3

RETI RICORRENTI

4

RETI NEURALI CONVOLUZIONALI

Timeline

START

lo strato di input prevedeva dati binari multipli in entrata mentre per l'output era previsto un singolo dato binario in uscita; per comporre la rete neurale bastava concettualmente combinare un numero opportuno di questi elementi



1943

McCulloch & Pitts

l'apprendimento hebbiano che si basa sul "peso", si basa sul semplice principio che se due neuroni si attivano contemporaneamente, la loro interconnessione deve essere rafforzata



1949

Hebb

Perceptron e indicava una rete con uno strato di ingresso ed uno di uscita ed una regola di apprendimento intermedia, basata sull'algoritmo, 'error back-propagation'



1958

Rosenblatt

Timeline

In Progress

dimostra i forti limiti del perceptrone: risultava essere infatti una rete neurale poco potente, incapace di calcolare la funzione "or esclusivo" (XOR).

1969

Marvin

retropropagazione dell'errore (error backpropagation) il quale modifica sistematicamente i pesi delle connessioni tra i nodi, così che la risposta della rete si avvicini sempre di più a quella desiderata

1986

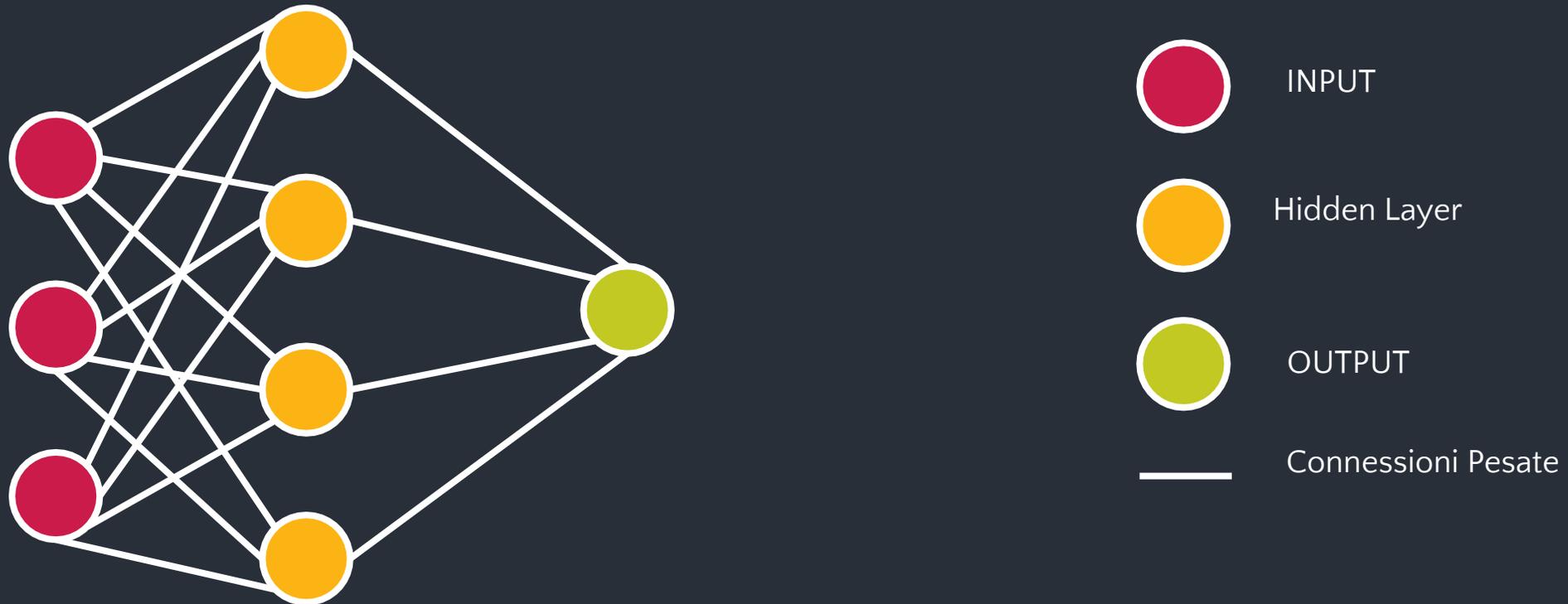
Rumelhart

Chip Neuromorfici: si stanno sviluppando chip e processori in grado di imitare il funzionamento del cervello

20..

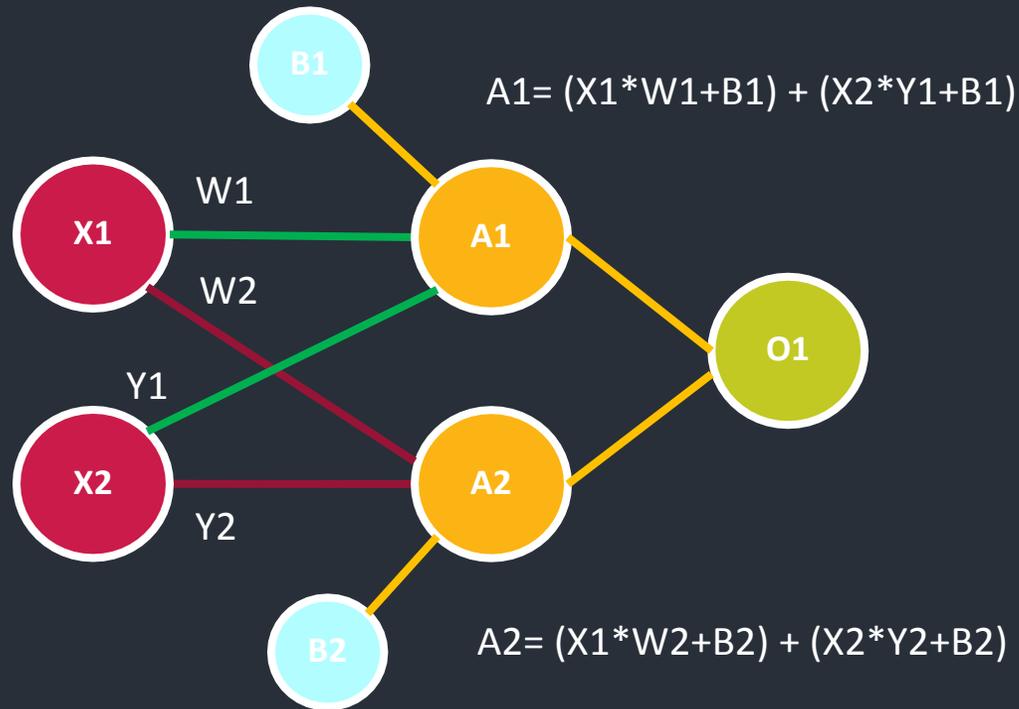
Reti Neurali

Come funziona



Reti Neurali

Come funziona



A1 ed A2 vengono passate ad una funzione di attivazione, che serve per pesare meglio le approssimazioni.

“ LO SCOPO DELLE RETI NEURALI È DI ESSERE UN UNIVERSAL **FUNCTION APPROXIMATOR**, OVVERO ESSERE IN GRADO DI APPROSSIMARE QUALSIASI FUNZIONE, E PER FARE QUESTO È NECESSARIO INTRODURRE UN FATTORE DI NON LINEARITÀ, DA QUI LA FUNZIONE DI ATTIVAZIONE. ”



Funzioni di Attivazione



1

Step Function: funzione a gradino

2

Funzione Sigmoide

3

Funzione RELU



Il **gradient descent** è una tecnica che ha lo scopo di minimizzare quanto possibile la cost function. Immaginando la cost function come funzione di sole due variabili (per semplificare), lo scopo del nostro gradient descent è quello di trovare il minimo globale della funzione, ovvero il punto più basso. In questo caso semplificato il minimo sembra abbastanza ovvio, ma nella maggior parte dei casi le funzioni sono molto più complesse, e bisogna arrivarci per approssimazioni successive.

