

# Machine Learning

Le reti neurali e l'apprendimento automatico

*“Nessuno entri, che non sia geometra.”*  
Platone

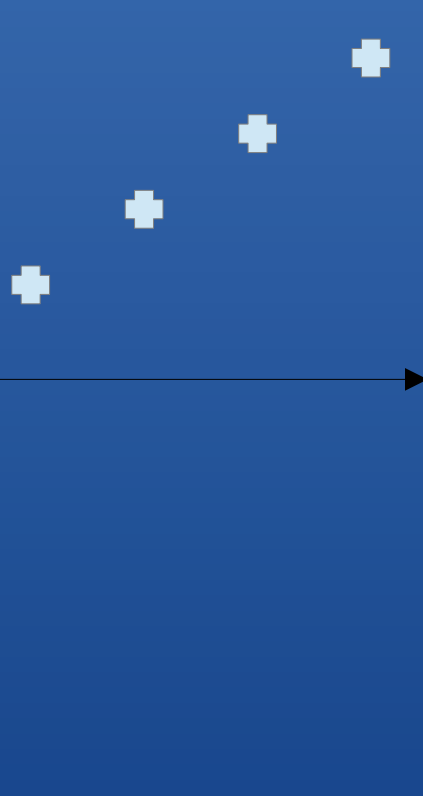
A cura di Pietro Squilla

# Indice

- Introduzione ai modelli matematici
- Modelli ed algoritmi
- Approfondimento sui DATI
- Storia delle reti neurali e funzionamento
- Applicazioni e chiarimenti

# Modelli classici

Settimana Enigmistica



# Modelli classici

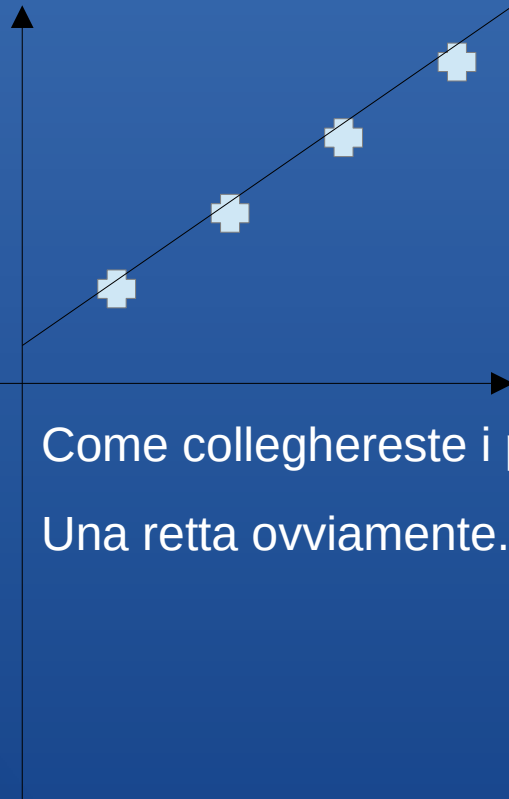
Settimana Enigmistica



Come colleghereste i punti?

# Modelli classici

Settimana Enigmistica

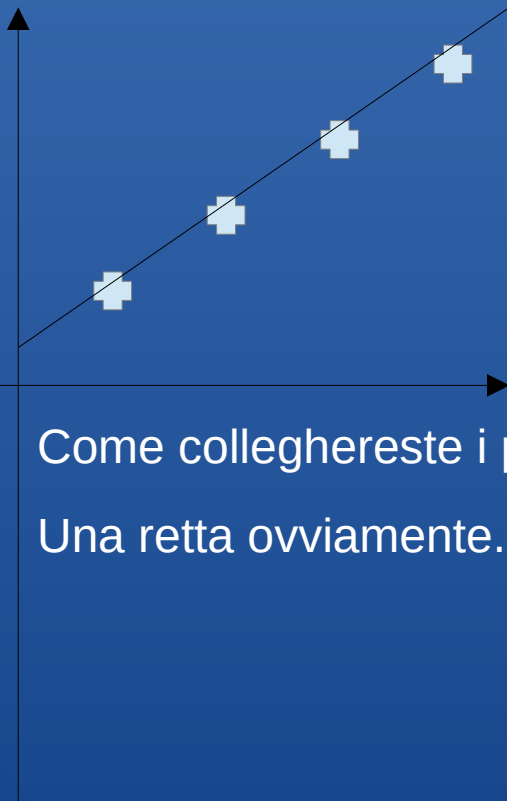


Come colleghereste i punti?

Una retta ovviamente...

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...

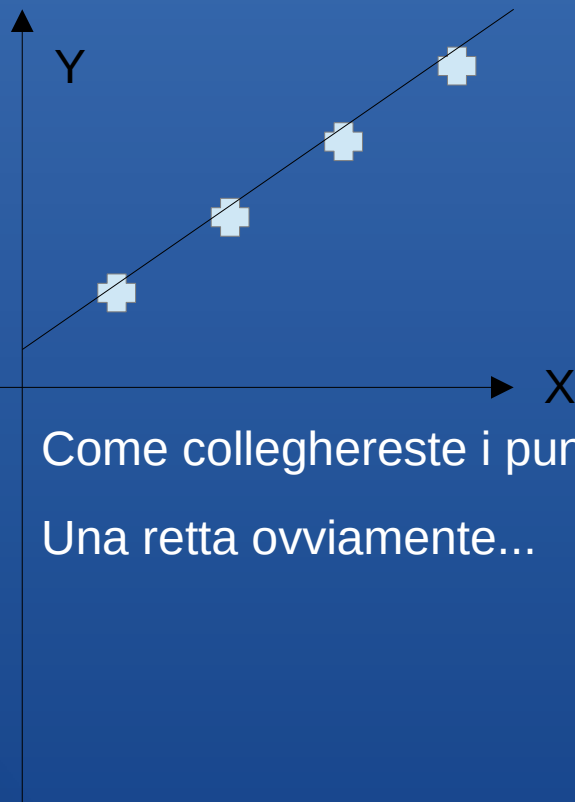


Modello lineare:

$$Y = (a) \cdot X + (b)$$

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

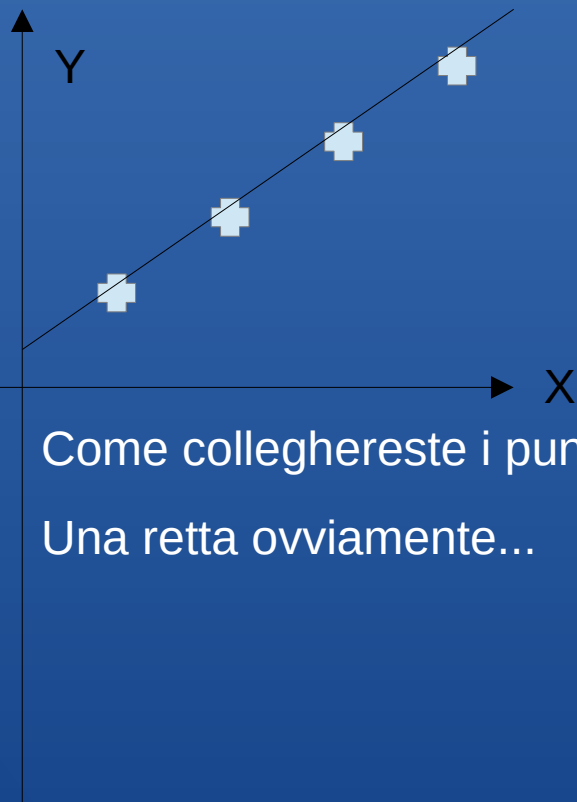
Una retta ovviamente...

Modello lineare:

$$Y = (a) \cdot X + (b)$$

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...



Modello lineare

$$Y = (a) \cdot X + (b)$$

Es: la legge di Hubble

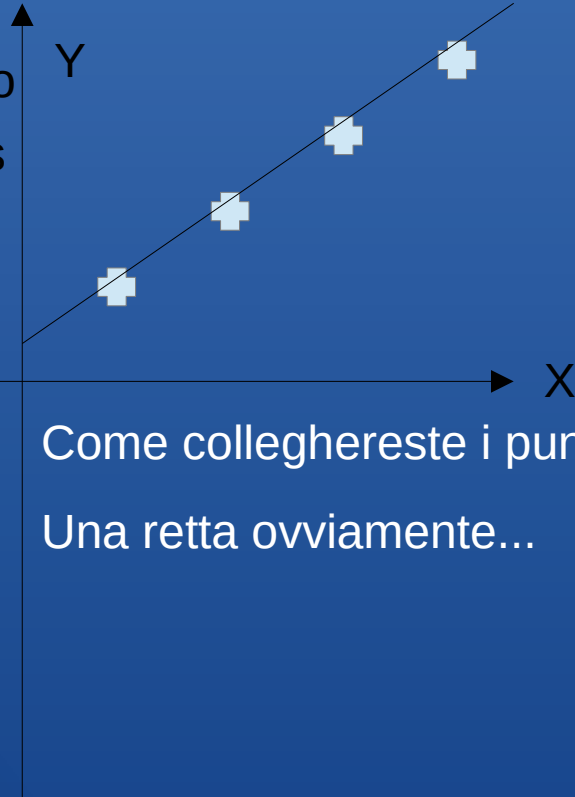
$$Y = H \cdot X$$



# Modelli classici

Settimana Enigmistica

Galassia Sombrero  
1024 Km/s



Modello lineare

$$Y = (a) \cdot X + (b)$$

Es: la legge di Hubble

$$Y = H \cdot X$$

# Modelli classici

Settimana Enigmistica

Galassia Sombrero

1024 Km/s



Come colleghereste i punti?

Una retta ovviamente...

Modello lineare

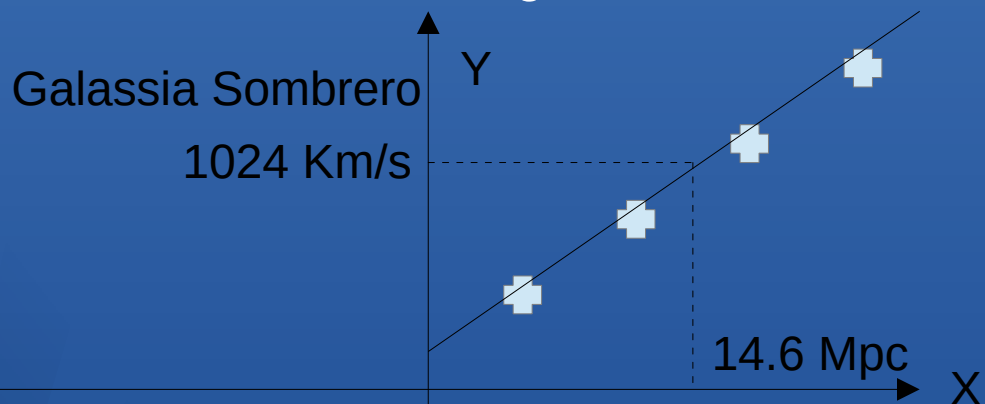
$$Y = (a) \cdot X + (b)$$

Es: la legge di Hubble

$$Y = H \cdot X$$

# Modelli classici

Settimana Enigmistica



Modello lineare

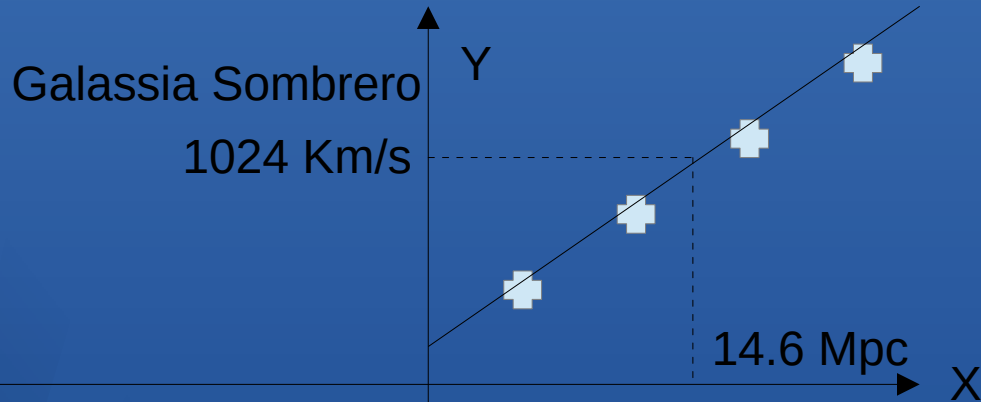
$$Y = (a) \cdot X + (b)$$

Es: la legge di Hubble

$$Y = H \cdot X$$

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...



Modello lineare

$$Y = (a) \cdot X + (b)$$

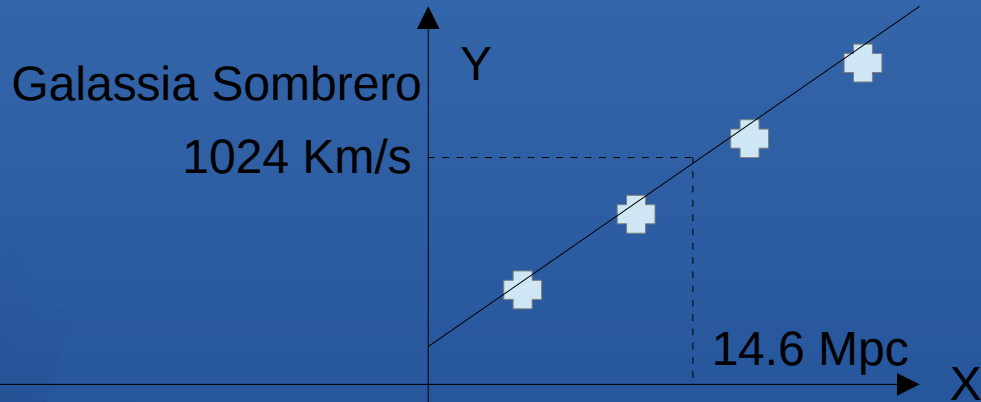
Es: la legge di Hubble

$$Y = H \cdot X$$

$$1024 = H \cdot 14.6$$

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...



Modello lineare

$$Y = (a) \cdot X + (b)$$

Es: la legge di Hubble

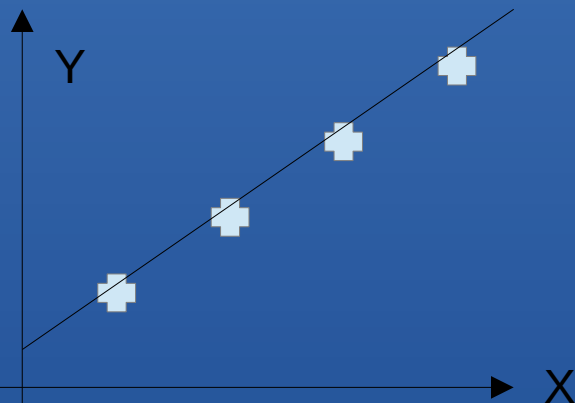
$$Y = H \cdot X$$

$$1024 = H \cdot 14.6$$

$$H = 72 \pm 8 \text{ km/s/Mpc}$$

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...

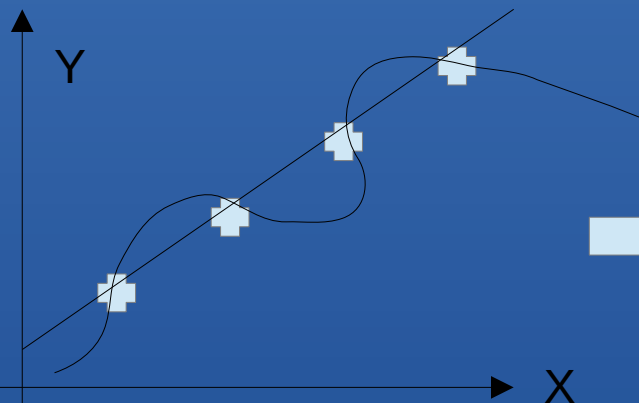
...e se fosse più complicato?

Modello lineare

$$Y = (a) \cdot X + (b)$$

# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...

...e se fosse più complicato?

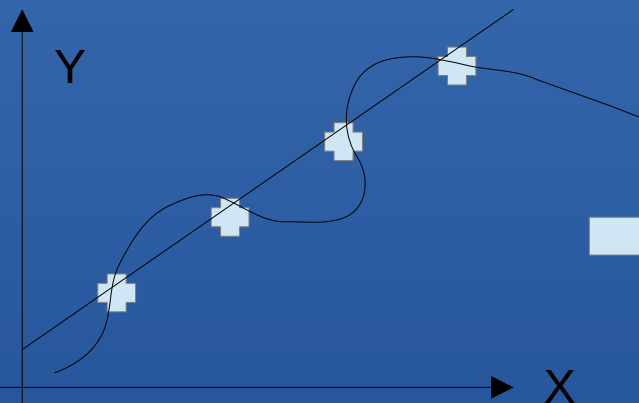
Modello lineare

$$Y = (a) \cdot X + (b)$$



# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...

...e se fosse più complicato?



Modello lineare

$$Y = (a) \cdot X + (b)$$

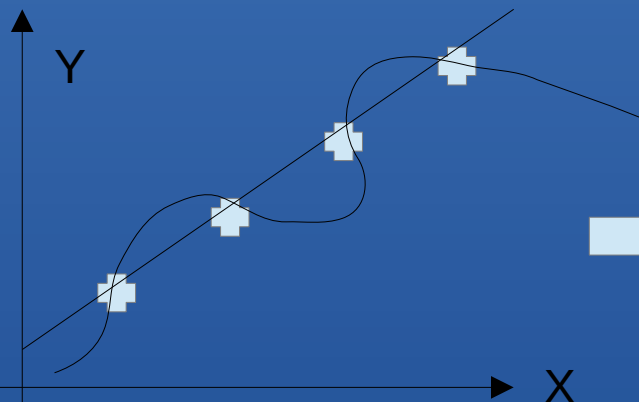
Modello non lineare

$$Y = X ?$$



# Modelli classici

Settimana Enigmistica



Come colleghereste i punti?

Una retta ovviamente...

...e se fosse più complicato?

Generalizziamo e chiamiamo modello una funzione  $Y=f(X)$

Modello lineare

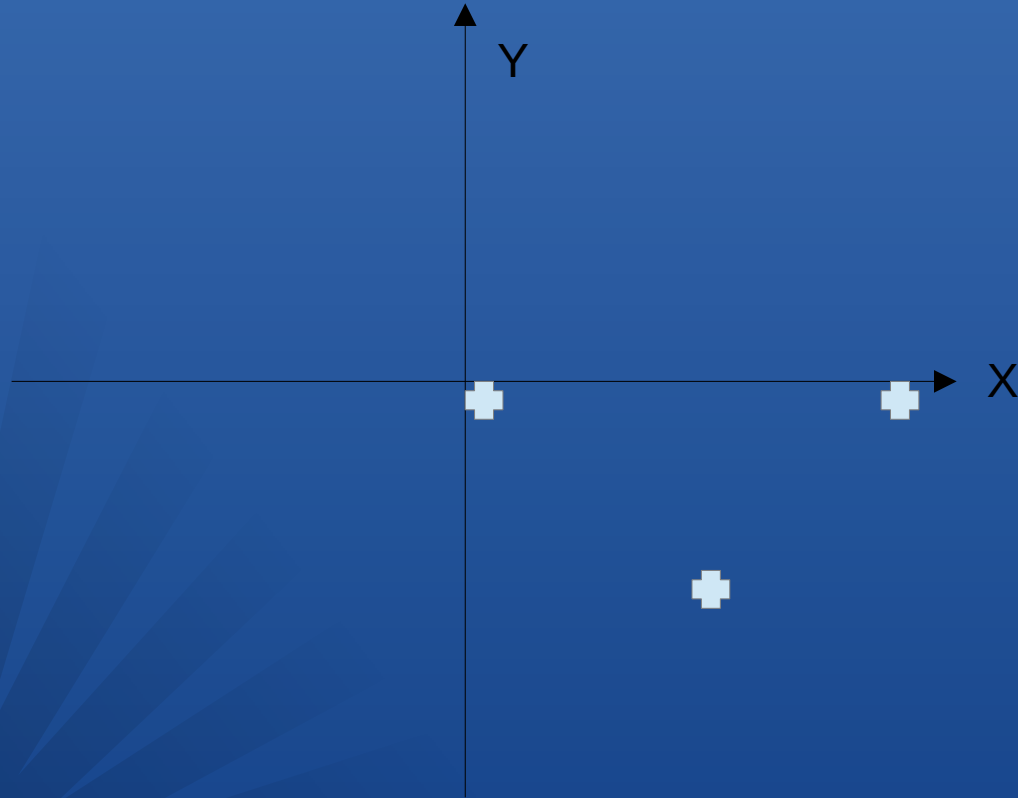
$$Y=(a)\cdot X+(b)$$

Modello non lineare

$$Y=X?$$

# Modelli classici

Esempio non lineare

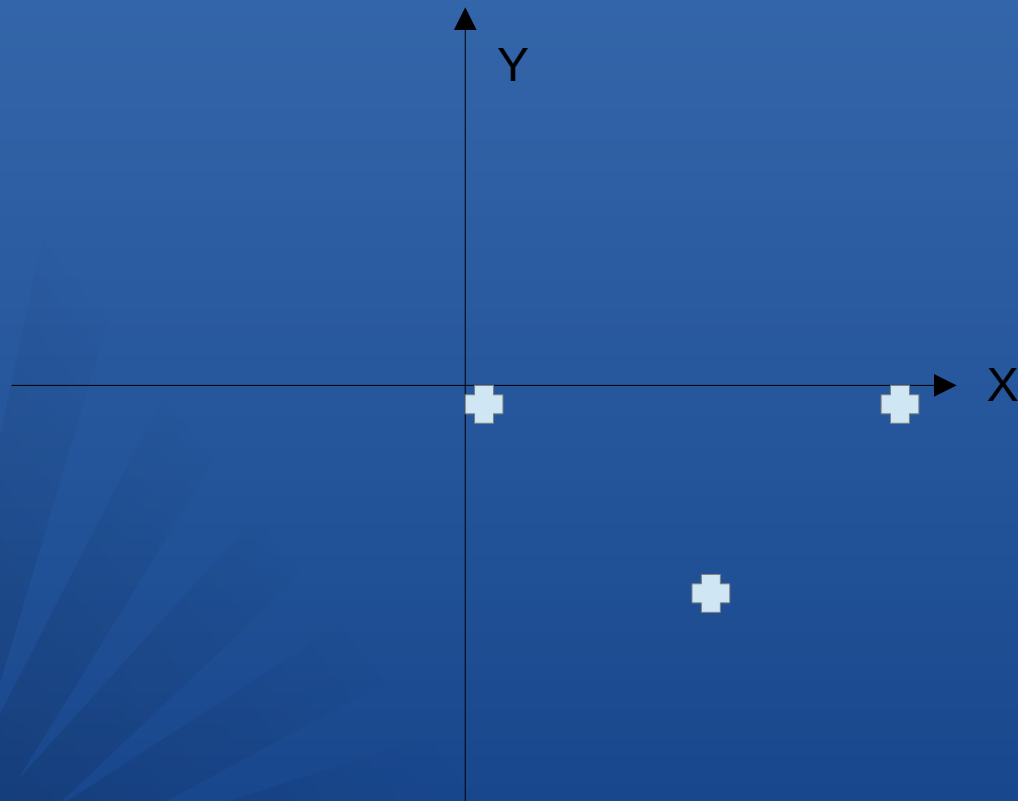


Modello non lineare semplice  
(quadratico)

$$Y = (a) \cdot X^2 + (b) \cdot X + (c)$$

# Modelli classici

Esempio non lineare



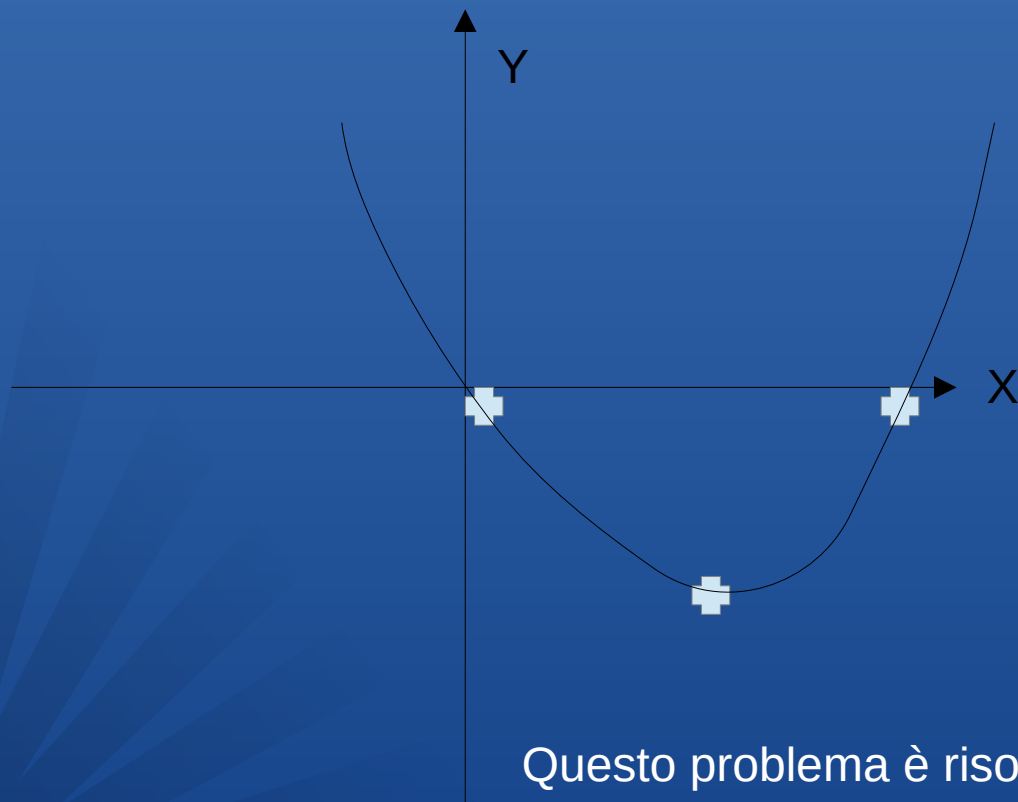
Modello non lineare semplice  
(quadratico)

$$Y = (a) \cdot X^2 + (b) \cdot X + (c)$$

scegliendo opportunamente  
questi numeri possiamo trovare  
una curva che passa per i punti

# Modelli classici

Esempio non lineare



Modello non lineare semplice  
(quadratico)

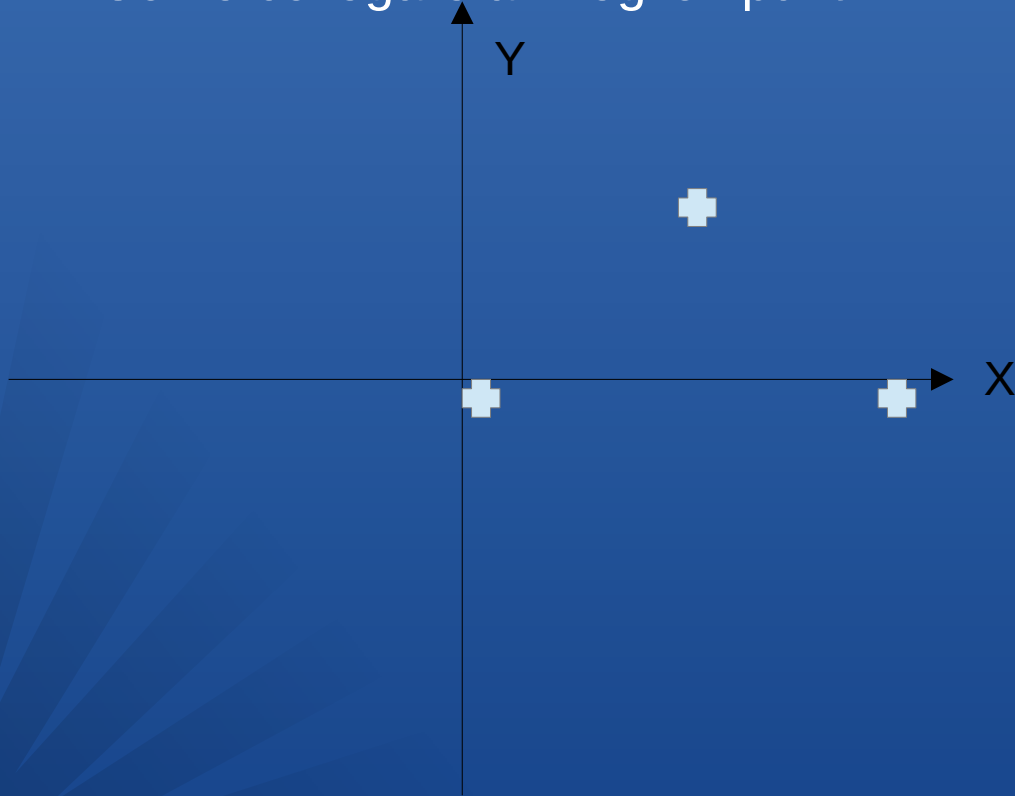
$$Y = (a) \cdot X^2 + (b) \cdot X + (c)$$

scegliendo opportunamente  
questi numeri possiamo trovare  
una curva che passa per i punti

Questo problema è risolto dai ragazzi di III superiore

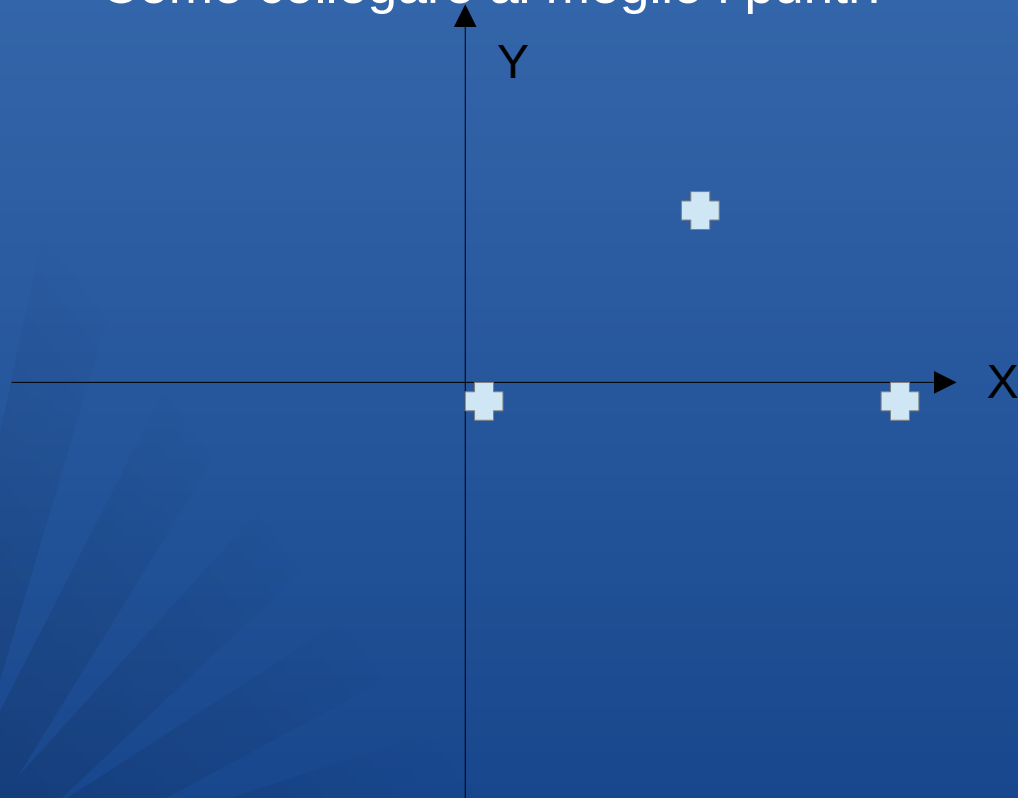
# Modelli classici

Ma se il modello è ignoto?  
Come collegare al meglio i punti?



# Modelli classici

Ma se il modello è ignoto?  
Come collegare al meglio i punti?

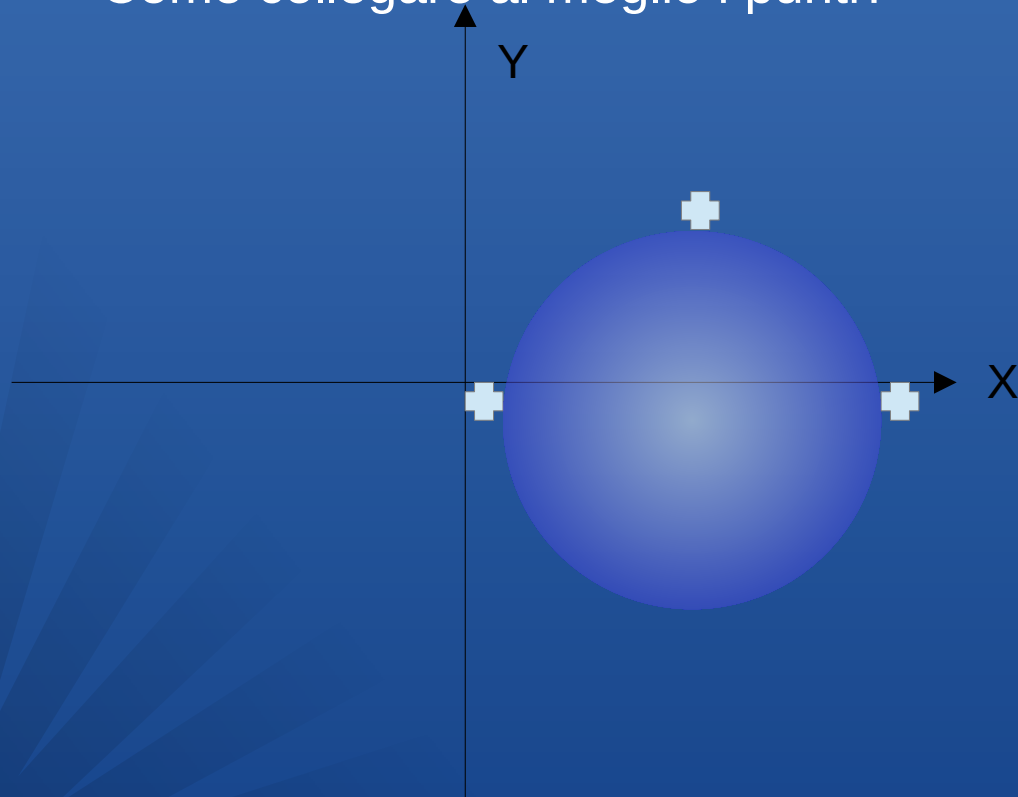


...dei candidati sarebbero:

- Parabola come prima
- Circonferenza
- Altro

# Modelli classici

Ma se il modello è ignoto?  
Come collegare al meglio i punti?

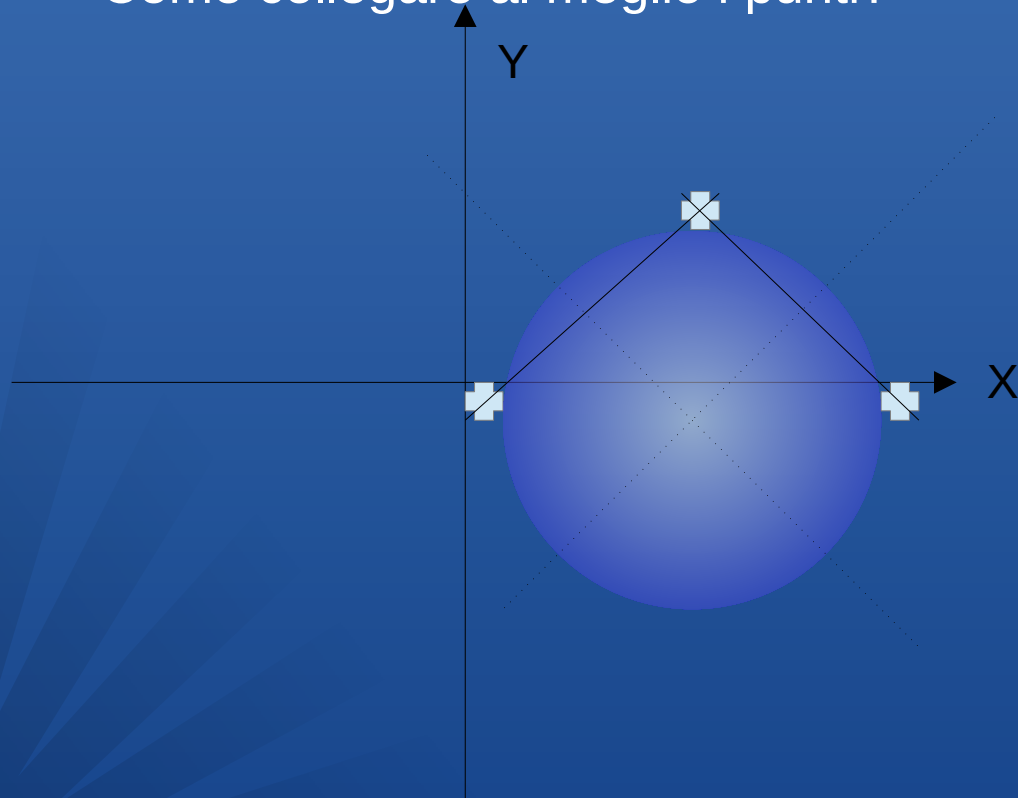


...dei candidati sarebbero:

- Parabola come prima
- Circonferenza
- Altro

# Modelli classici

Ma se il modello è ignoto?  
Come collegare al meglio i punti?



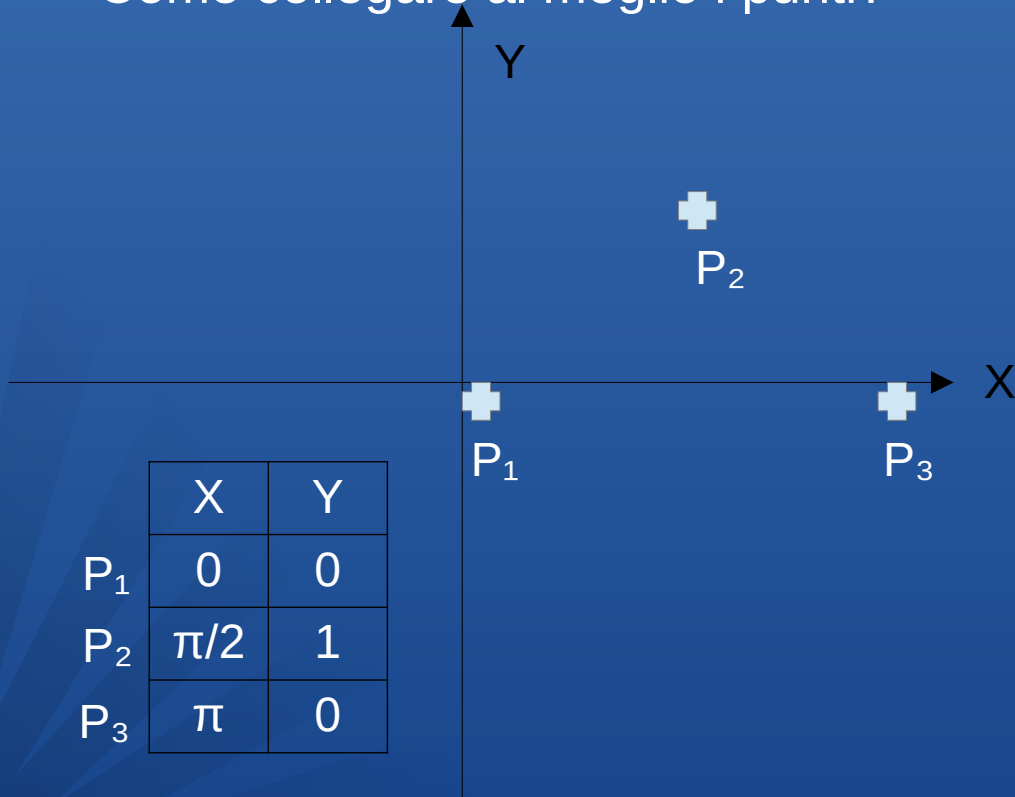
...dei candidati sarebbero:

- Parabola come prima
- Circonferenza
- Altro?



# Modelli classici

Ma se il modello è ignoto?  
Come collegare al meglio i punti?



...dei candidati sarebbero:

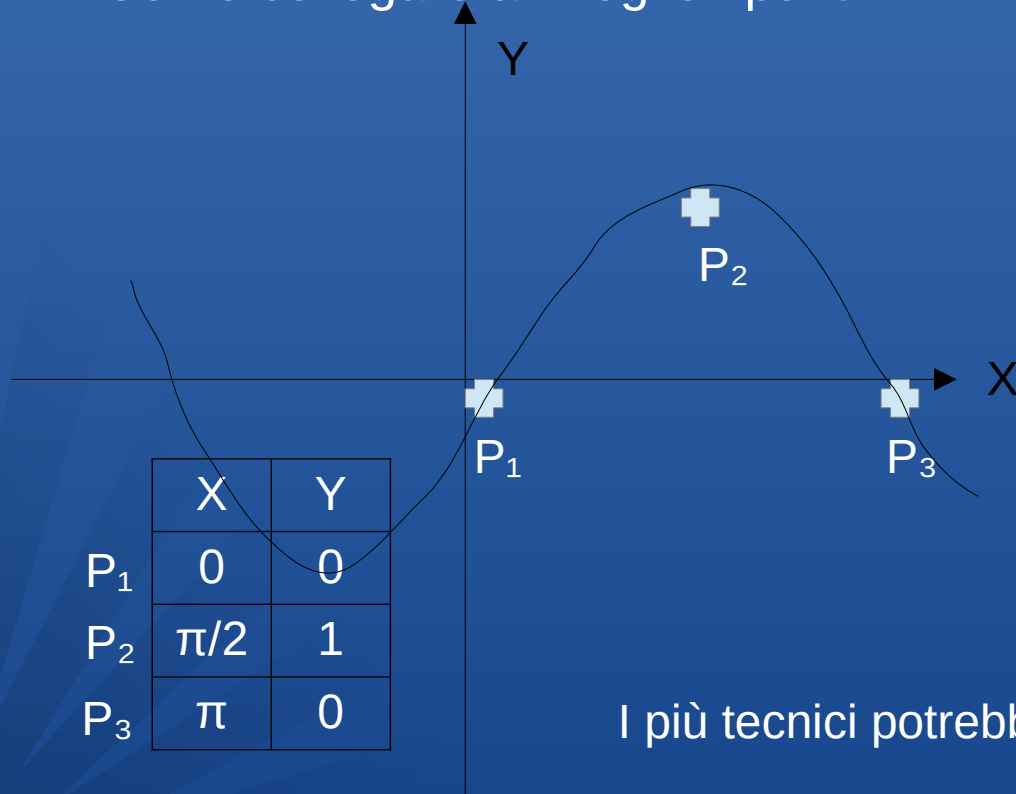
- Parabola come prima
- Circonferenza
- Altro?

Con informazioni specifiche sul dataset?

Idee?

# Modelli classici

Ma se il modello è ignoto?  
Come collegare al meglio i punti?



...dei candidati sarebbero:

- Parabola come prima
- Circonferenza
- Altro?

Con informazioni specifiche sul dataset?

Idee?

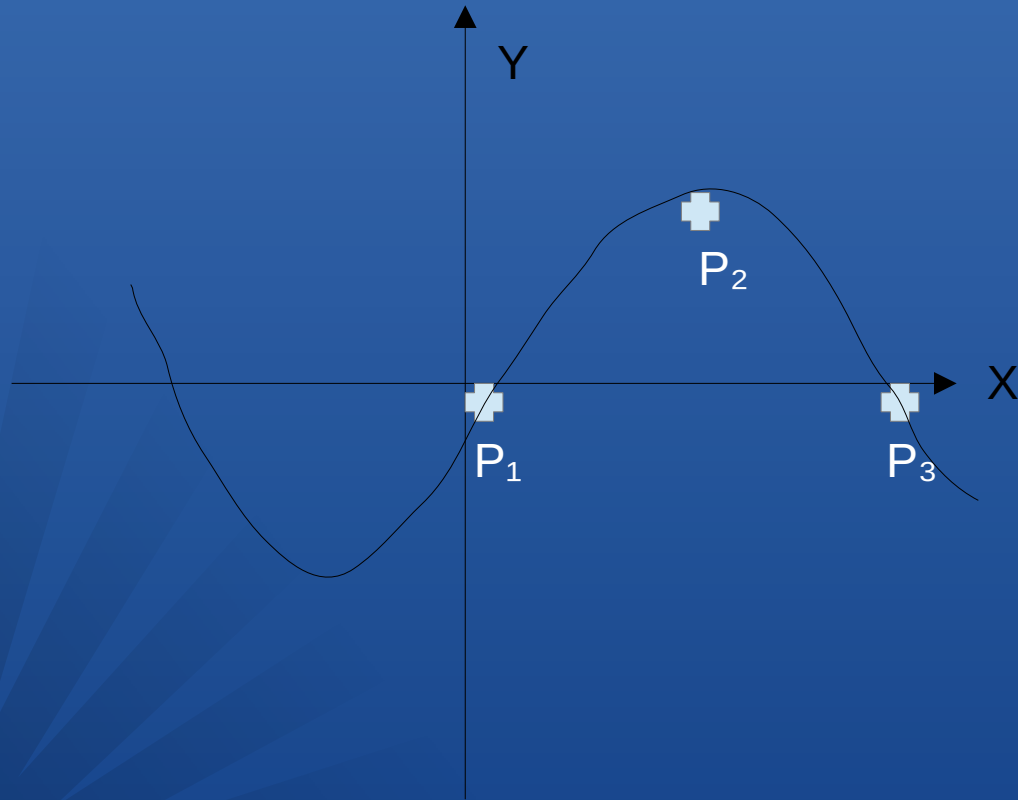
I più tecnici potrebbero azzardare  $Y = \sin(X)$  !

# Algoritmi classici

- *Non è sempre facile sapere quale modello descrive quale fenomeno! Questa è la differenza tra modello e realtà.*
- *Possiamo “automatizzare” il processo creando algoritmi che approssimano sempre meglio i nostri dati.*
- *Esempio: interpolazione polinomiale di Lagrange.*

# Modelli classici

## Polinomi di Lagrange

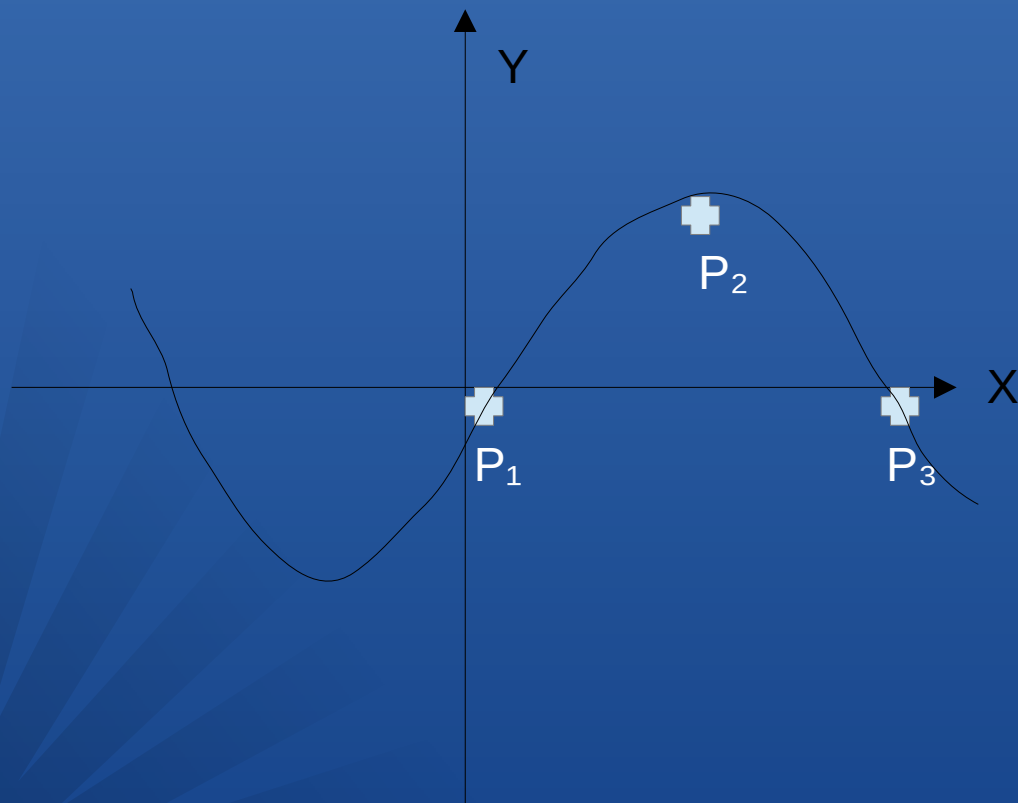


Il modello reale:  $Y = \sin(X)$

	$X$	$Y$
$P_1$	0	0
$P_2$	$\pi/2$	1
$P_3$	$\pi$	0

# Modelli classici

## Polinomi di Lagrange



Il modello reale:  $Y = \sin(X)$

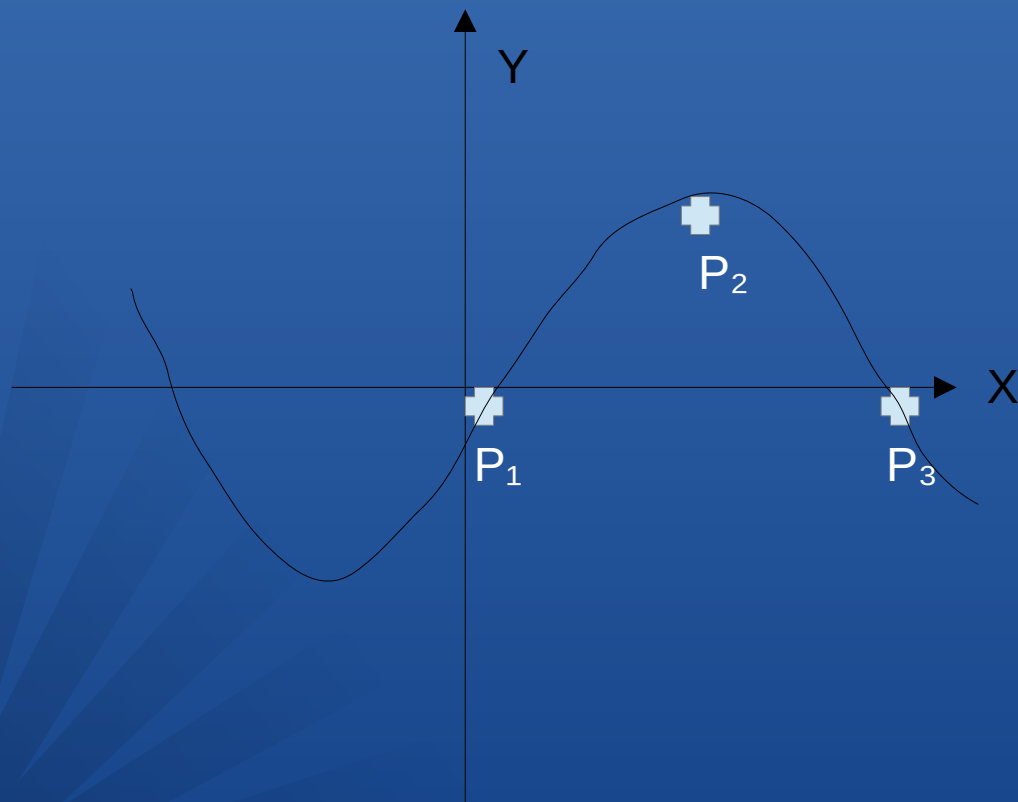
	X	Y
P <sub>1</sub>	0	0
P <sub>2</sub>	$\pi/2$	1
P <sub>3</sub>	$\pi$	0

Modello lagrangiano

$$P(X) = y_0 \cdot L_0(X) + \dots + y_{n-1} \cdot L_{n-1}(X)$$

# Modelli classici

## Polinomi di Lagrange



Il modello reale:  $Y = \sin(X)$

	X	Y
P <sub>1</sub>	0	0
P <sub>2</sub>	π/2	1
P <sub>3</sub>	π	0

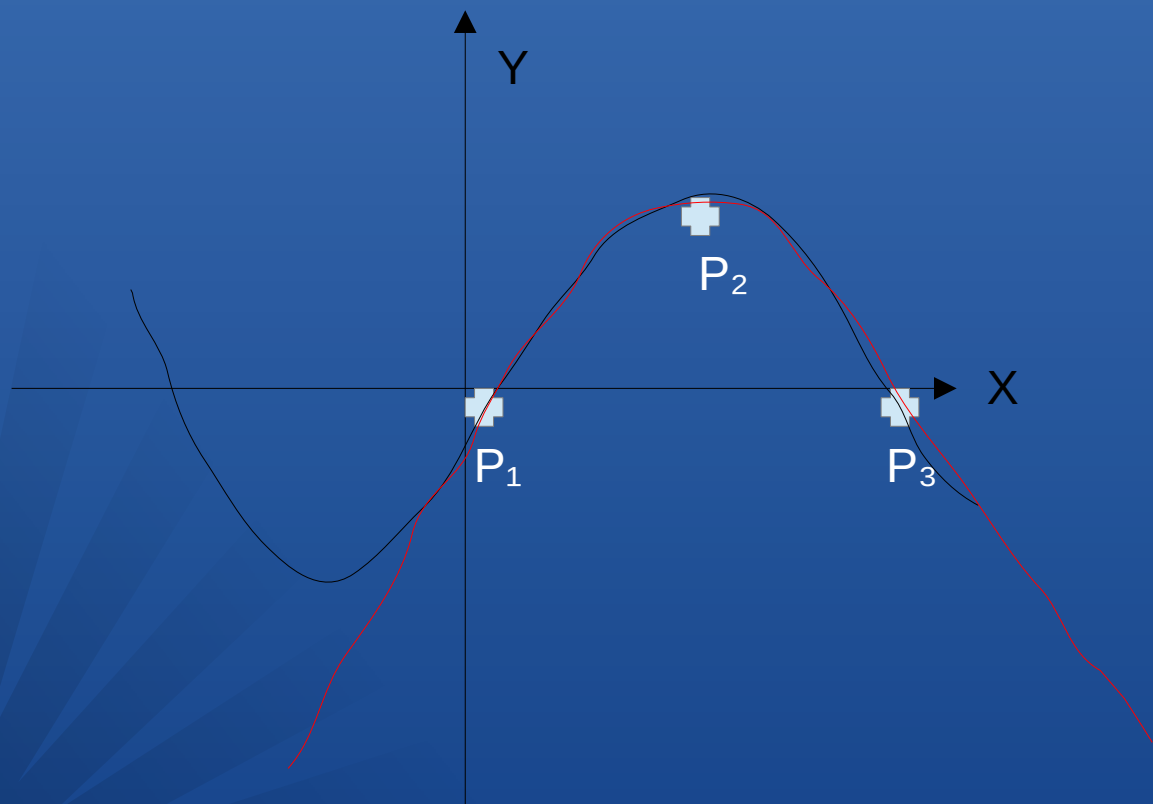
Modello lagrangiano

$$P(X) = y_0 \cdot L_0(X) + \dots + y_{n-1} \cdot L_{n-1}(X)$$

$$L_n = \prod_{j=0, j \neq i}^n \frac{(X - x_j)}{x_i - x_j}$$

# Modelli classici

## Polinomi di Lagrange



Il modello reale:  $Y = \sin(X)$

	X	Y
$P_1$	0	0
$P_2$	$\pi/2$	1
$P_3$	$\pi$	0

Modello lagrangiano

$$P(X) = y_0 \cdot L_0(X) + \dots + y_{n-1} \cdot L_{n-1}(X)$$

$$L_n = \prod_{j=0, j \neq i}^n \frac{(X - x_j)}{x_i - x_j}$$

# Algoritmi classici

Per quanto flessibili i modelli classici hanno dei limiti imposti dalla matematica alla loro base.

Esempio: con Lagrange il modello produrrà dei pattern in risposta a dati nuovi che ereditano tutte le caratteristiche dei polinomi!



# Algoritmi classici

Limiti dei polinomi nel calcolo delle maree:

Fourier cercò un modello che descrivesse le maree a fini scientifici e militari (conoscere esattamente le correnti). Non bastò interpolare i dati, serviva riconoscere componenti periodiche multiple.

Servì approssimare il comportamento globale, non solo puntuale.

# Algoritmi classici

Ma come creare un modello che non abbia questo “bias” o preconcezzo matematico?

Senza una struttura matematica fissa per lo specifico scopo?

Un modello che sia un “guscio vuoto” che non limiti la complessità del modello target  $f(x)$  ignoto.

# Algoritmi classici

Modelli del genere si chiamano approssimatori universali.

Degli esempi sono:

**Modelli deterministici:** polinomi ortogonali (Fourier, ecc).

Sono interpretabili, non necessitano di una potenza di calcolo esagerata, non dipendono dalla mole di dati ma più dalle ipotesi fatte a monte dall'uomo (nel caso di Fourier dalle ipotesi del teorema).

**Modelli statistici:** alberi decisionali, reti neurali, altri...

Dipendono dalla quantità di dati, creano una statistica da inferire, necessitano di potenza di calcolo per essere utili.

# Algoritmi classici

Modelli del genere si chiamano approssimatori universali.

Degli esempi sono:

**Modelli deterministici:** polinomi ortogonali (Fourier, ecc).

Sono interpretabili, non necessitano di una potenza di calcolo esagerata, non dipendono dalla mole di dati ma più dalle ipotesi fatte a monte dall'uomo (nel caso di Fourier dalle ipotesi del teorema).

**Modelli statistici:** alberi decisionali, reti neurali, altri...

Dipendono dalla quantità di dati, creano una statistica da inferire, necessitano di potenza di calcolo per essere utili.

# Algoritmi

Ci avviciniamo alle reti neurali...

# Algoritmi

Ci avviciniamo alle reti neurali...

...ma prima un approfondimento sui **DATI**!

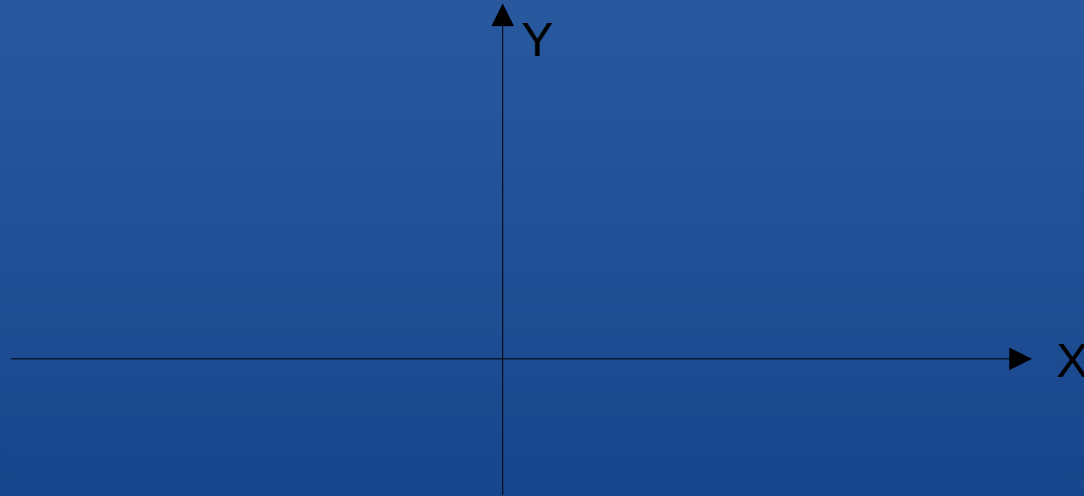
# Dati

La rappresentazione geometrica delle relazioni tra dati (qualsiasi) avviene definendo uno spazio in cui i dati possono “vivere”, in cui hanno un senso.

# Dati

Se vogliamo esplicitare la relazione di un dato  $P$  composto da due grandezze, avremo che  $X \rightarrow Y$ .

Geometricamente le informazioni avranno senso in uno spazio 2D

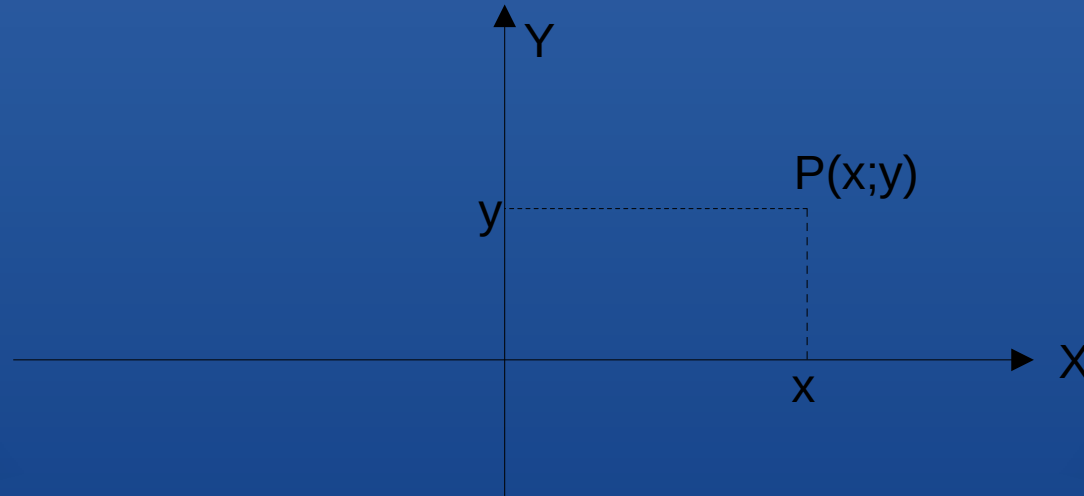




# Dati

Se vogliamo esplicitare la relazione di un dato  $P$  composto da due grandezze, avremo che  $X \rightarrow Y$ .

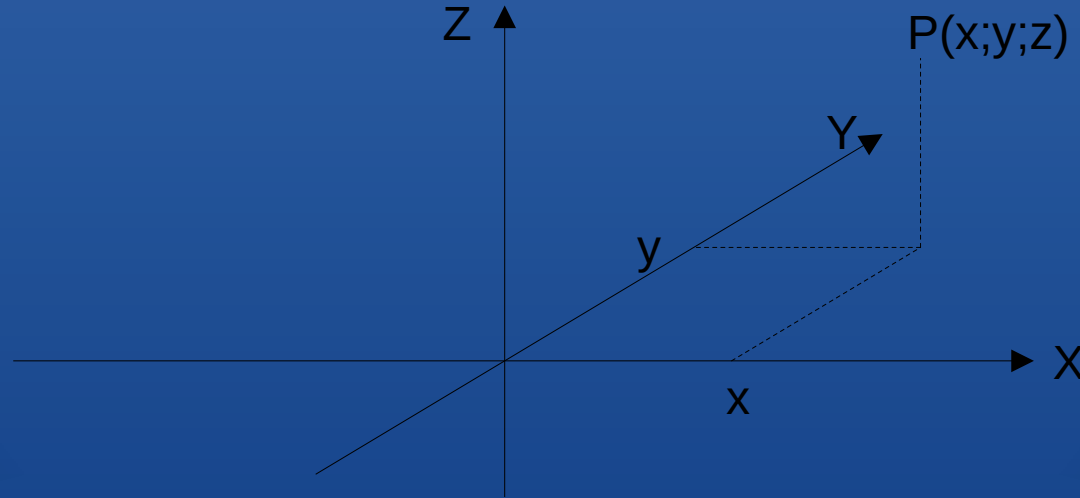
Geometricamente le informazioni avranno senso in uno spazio 2D



# Dati

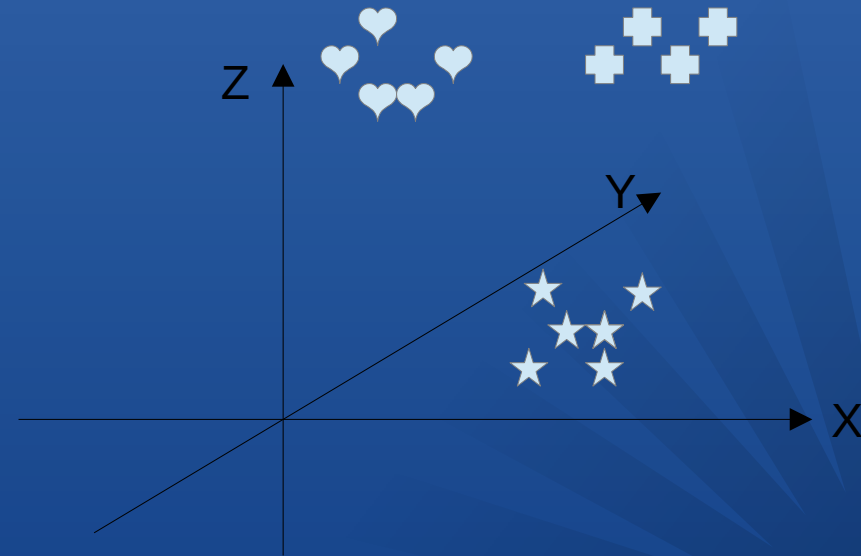
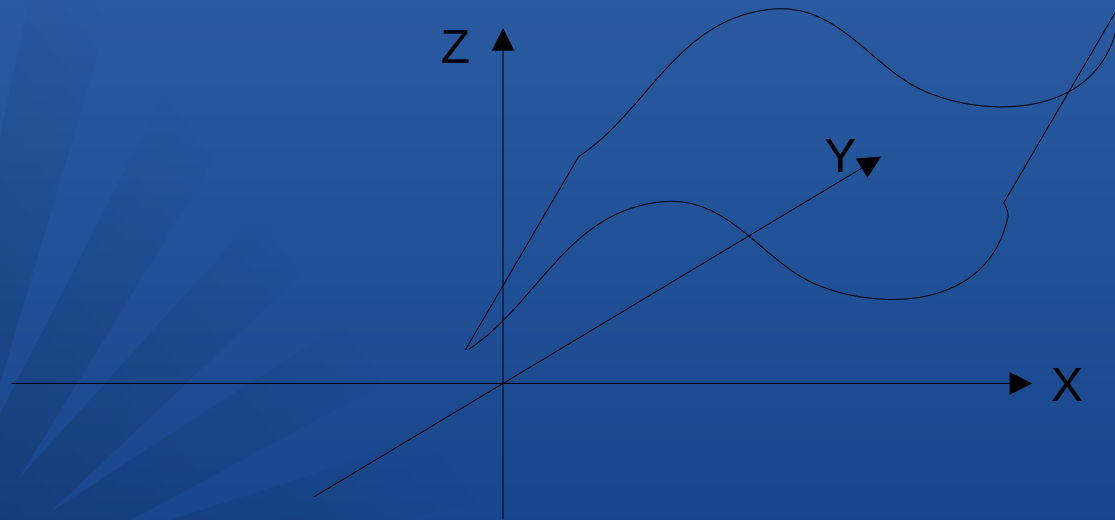
Se vogliamo esplicitare la relazione tra più grandezze, per esempio  $X, Y \rightarrow Z$

Geometricamente le informazioni avranno senso in uno spazio 3D



# Dati

Un modello può quindi descrivere dei dati  
disposti con varie geometrie (superfici, cluster)



# Dati

Chiamiamo questo oggetto matematico “spazio vettoriale” e può essere definito con  $N$  dimensioni!

La dimensionalità dello spazio dipende dai dati, conferisce espressività geometrica oltre che completezza.

Umanamente è difficile visualizzare spazi di dimensioni superiori a 3, ciò non significa che non esistano o non possano avere senso.

# Dati

Gli algoritmi di ML servono proprio a questo, per cercare pattern, proprietà che non vediamo, definire modelli.

Esempio: dati generati con RANDU

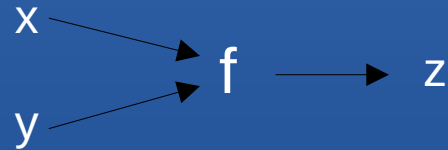
# Dati

Ora possiamo capire anche due diverse applicazioni, alla luce dei modelli che descrivono dati “ambientati”:

# Dati

Ora possiamo capire anche due diverse applicazioni, alla luce dei modelli che descrivono dati “ambientati”:

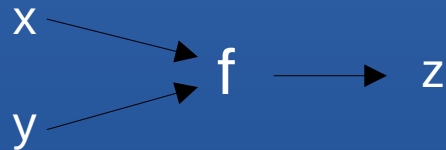
Modelli di regressione



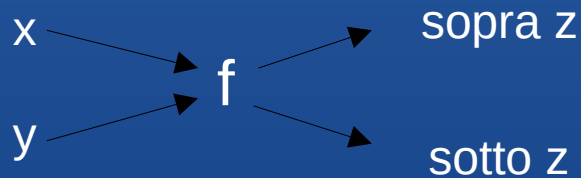
# Dati

Ora possiamo capire anche due diverse applicazioni, alla luce dei modelli che descrivono dati “ambientati”:

Modelli di regressione



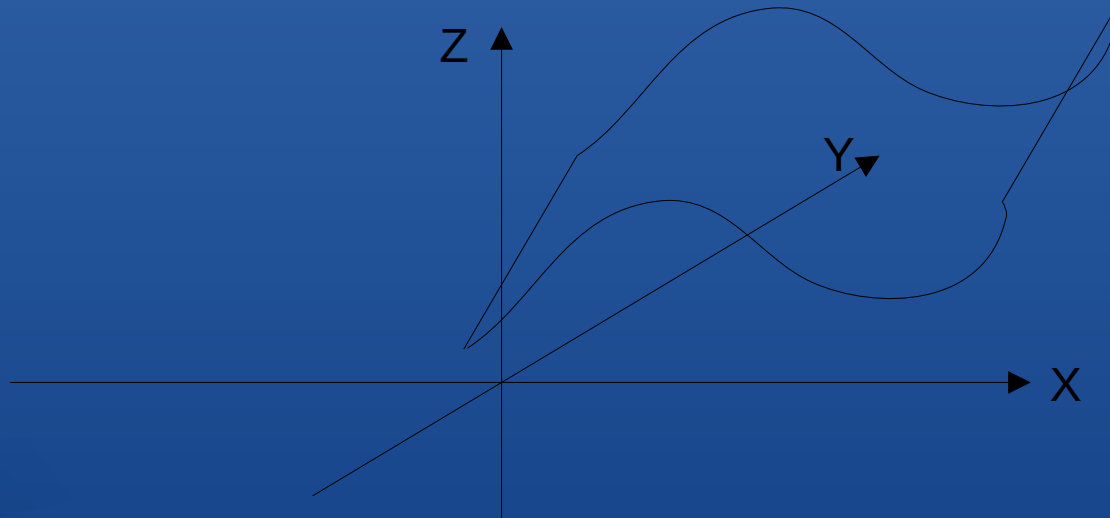
Modelli di classificazione





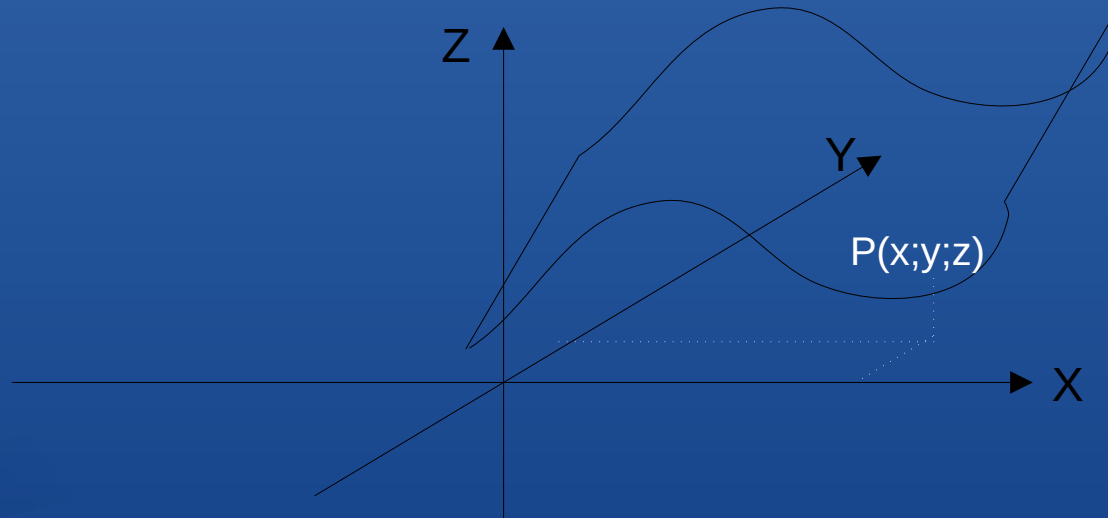
# Modello di regressione

- Prevede quale  $z$  è associabile ai dati  $x, y$ .
- Trova quindi dei punti di una superficie.
- È quello che fanno i modelli visti finora.



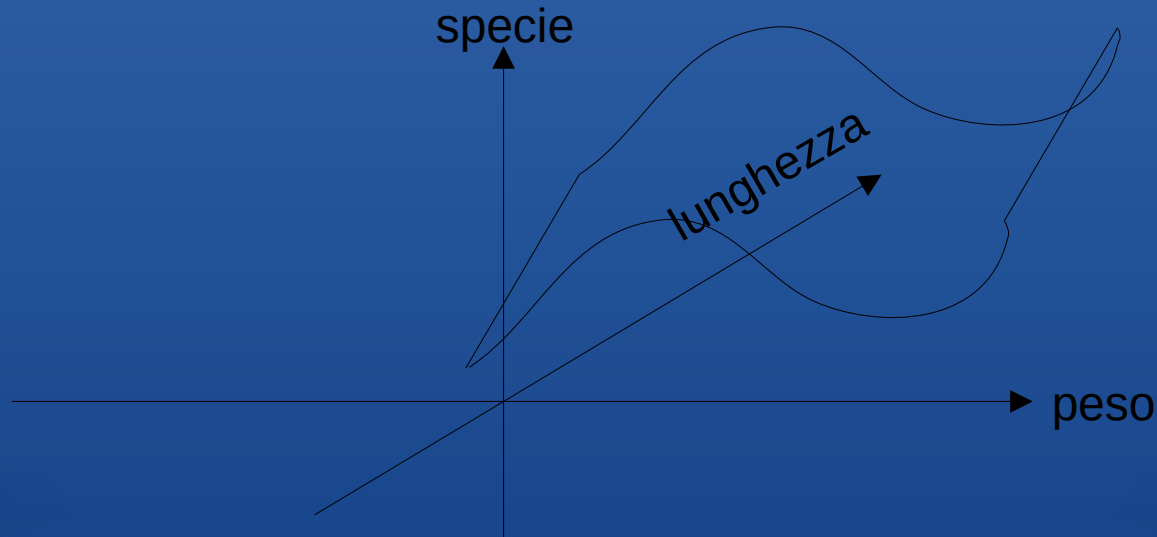
# Modello di regressione

- Prevede quale  $z$  è associabile ai dati  $x, y$ .
- Trova quindi dei punti di una superficie.
- È quello che fanno i modelli visti finora.



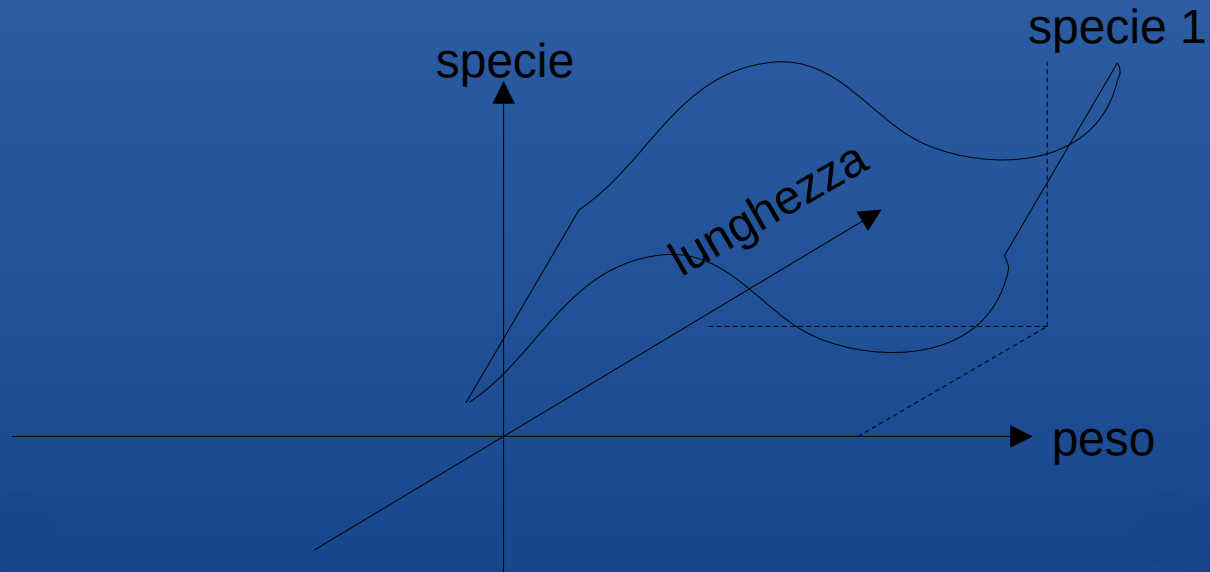
# Modello di classificazione

- Seziona lo spazio in regioni.
- Dati in regioni diverse sono categorie diverse
- Esempio: classificazione dei pesci



# Modello di classificazione

- Seziona lo spazio in regioni.
- Dati in regioni diverse sono categorie diverse
- Esempio: classificazione dei pesci

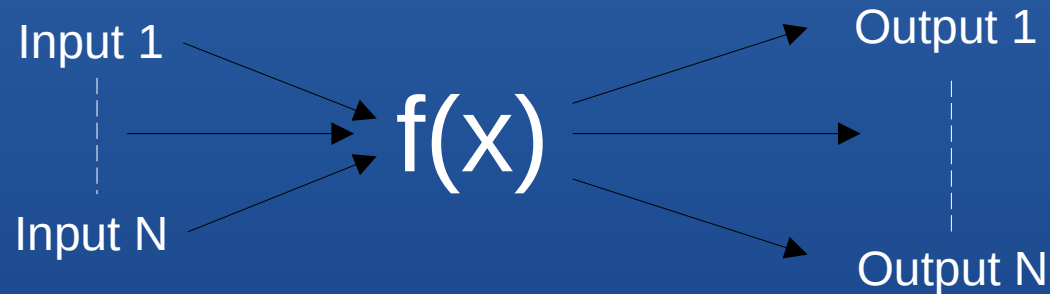


# Dati

Avete notato qualcosa?  
Una struttura particolare che il nostro modello  
 $f(x)$  possiede anche se ignoto?

# Dati

Avete notato qualcosa?  
Una struttura particolare che il nostro modello  
 $f(x)$  possiede anche se ignoto?

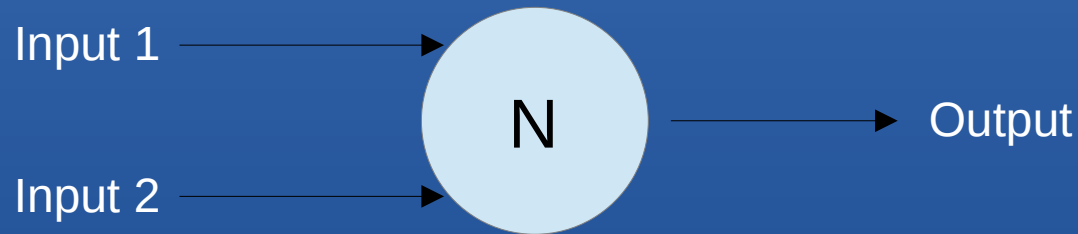


# Reti neurali

Da dove vengono fuori?

# Reti neurali

- **1943:** McCulloch e Pitts, un neurologo ed un logico matematico, vogliono modellizzare il cervello in modo formale. Sviluppano l'idea di neurone come porta logica a soglia.



A differenza delle porte logiche, il neurone ha un output solo se il segnale che lo raggiunge supera una certa soglia, in analogia con il comportamento dei neuroni biologici.



# Reti neurali

- **1949:** Hebb (neuroscienziato) propone un principio di apprendimento:  
**“neuroni che si attivano insieme si rafforzano insieme”.**

É fonte di ispirazione per i lavori successivi.

- **1958:** Rosenblatt (informatico e psicologo) costruisce il primo percettrone, una “rete” ad un solo strato in grado di trovare i coefficienti di una funzione che fa fit lineari (la miglior regressione).

Lo fa però con il concetto di neurone sviluppato dai colleghi precedenti, dando una forma al principio di Hebb.

# Reti neurali

- Il perceptrone è un modello di regressione, come l'equazione della retta, ma fatto in uno spazio non necessariamente 2D come quello del piano cartesiano a cui siamo abituati.
- I coefficienti diventano “pesi” delle connessioni, cioè i numeri trovati a partire dai dati; si possono pensare come i pesi di una media:

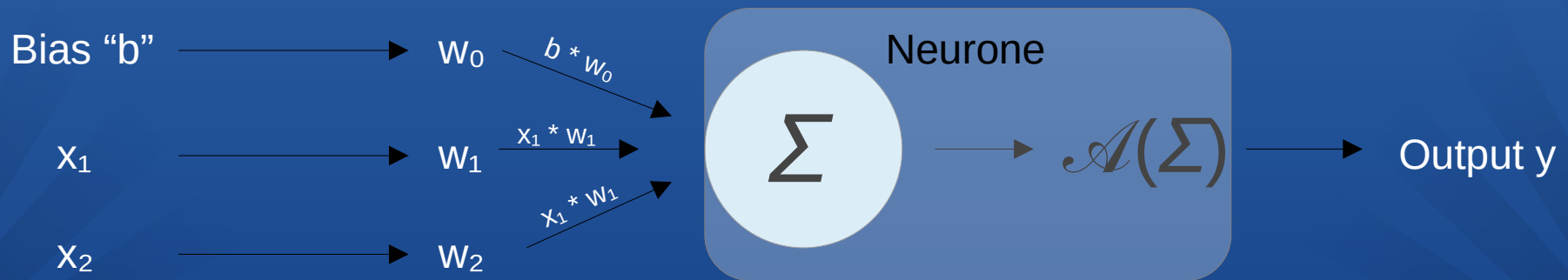
# Reti neurali

- Il perceptrone è un modello di regressione, come l'equazione della retta, ma fatto in uno spazio non necessariamente 2D come quello del piano cartesiano a cui siamo abituati.
- I coefficienti diventano “pesi” delle connessioni, cioè i numeri trovati a partire dai dati; si possono pensare come i pesi di una media:



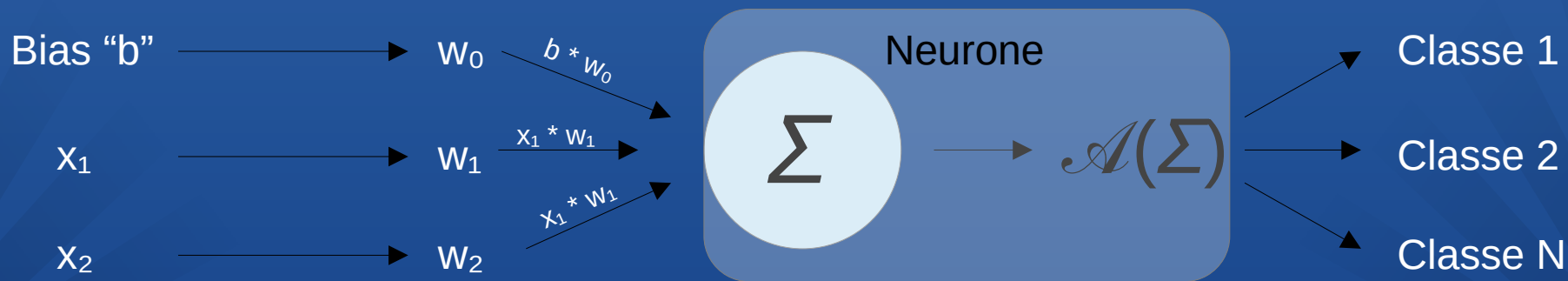
# Reti neurali

- Il perceptrone è un modello di regressione, come l'equazione della retta, ma fatto in uno spazio non necessariamente 2D come quello del piano cartesiano a cui siamo abituati.
- I coefficienti diventano “pesi” delle connessioni, cioè i numeri trovati a partire dai dati; si possono pensare come i pesi di una media:



# Reti neurali

- NB: combinazioni di operazioni lineari producono risultati lineari.
- Altro modo pittorico di rappresentare il percettrone (in versione di classificatore stavolta):



# Reti neurali

- Il problema adesso è: come trovo i pesi?
- Una analogia utile per capire il meccanismo alla base è l'operazione di taratura di una bilancia di precisione (esempio).
- L'apprendimento perciò si chiama supervisionato (perché mostriamo degli esempi di abbinamenti input/output ed aggiustiamo i pesi intermedi per far tornare il risultato giusto).
- Esistono molti metodi, tutti basati sulla potenza di calcolo. Nel caso del perceptrone veniva solo modificato il peso in base al calcolo dell'errore commesso rispetto all'output giusto.

# Limiti e problemi

- Il perceptrone esegue il calcolo  $f(x) = (w * x + b)$  con funzione di attivazione di tipo soglia nei classificatori o sigmoide per i regressori.
- In ogni caso il suo nucleo operativo è una combinazione lineare degli input, quindi risponde solo linearmente!
- Non riesce a separare o prevedere dati non lineari!

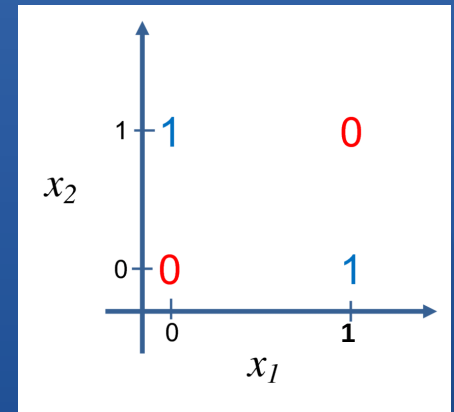
# Reti neurali

- **1969:** Minsky e Papert (due matematici) pubblicano una trattazione sul limite del perceptrone con problemi non lineari; dimostrano che non riesce ad emulare il comportamento della porta logica XOR.
- **Anni '70:** sono gli anni chiamati “AI winter”

Il campo di studi viene screditato

Le reti neurali sembrano poco promettenti.

Non si dispone della potenza di calcolo sufficiente per fare cose utili (perché in fondo è il modo più stupido per fare dei fit).





# Reti neurali

- **1982:** Hopfield (fisico premio Nobel 2024) riprende in mano le reti. Non per fare fit ma per studiare la meccanica statistica.

Usa le reti per modellare i problemi sui vetri di spin.

Era interessato alla capacità dei sistemi di possedere una memoria associativa; nel suo caso introdusse tutti gli strumenti matematici utili per trattare i problemi fisici ma con reti di oggetti interagenti.

Introdusse le fully-connected e grazie alla funzione energia da minimizzare riusciva a far riconoscere dei pattern anche in caso di input corrotti o assenti, attraverso l'evoluzione verso gli attrattori.

# Reti neurali

- **1983-1986:** Hinton (psicologo e premio Nobel) nel corso di questi anni collabora con altri accademici riprendendo il lavoro di Hopfield. Pubblica molti lavori dove infine supera i limiti precedenti.

Rilancia il metodo usato nel percettrone di backpropagation, ma questa volta per reti multistrato.

(il metodo di propagazione nel percettrone non è in grado di risalire all'origine dell'errore in eventuali livelli intermedi)

Si introduce formalmente la funzione di costo e la tecnica della “discesa del gradiente”.

# Addestramento (tecnico)

- La funzione di costo fornisce la stima dell'errore in output.
- Viene usata la regola di derivazione della catena tra funzione di costo e funzioni di attivazione per capire quali neuroni sbagliano di più.
- Si modificano così tutti i pesi e si valuta il dato successivo; quale errore viene commesso dopo l'aggiornamento dei pesi.
- Viene ripetuto il processo su tutto il dataset (più volte) fino ad ottenere un errore considerato statisticamente accettabile.
- Si ottiene un minimo di errore e la rete risulta addestrata.

# Addestramento (analogia)

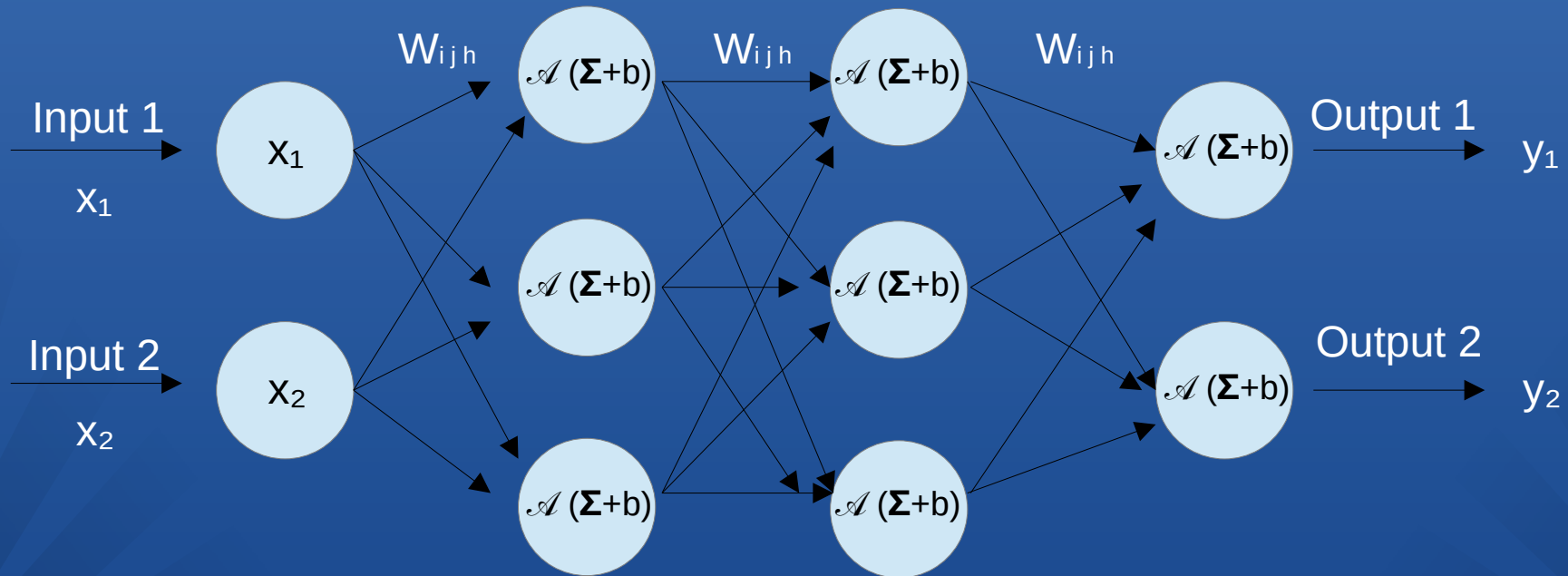
- La funzione di costo è un regista che valuta la bontà degli attori alla fine di una scena.
- Gli attori sono i neuroni che seguono un copione sempre uguale (che è la funzione di attivazione) ma per scene diverse (i dati).
- Se la scena è insoddisfacente (errore alto) il regista fornisce istruzioni di correzione a seconda dell'attore più rilevante nella scena; ne stempera l'atteggiamento alla luce del copione.
- Si ripete la scena addestrando gli attori ad assumere l'atteggiamento giusto su ogni scena del film (cioè sull'intero dataset).

# Architettura

Ora possiamo concepire una ANN (Artificial Neural Network):

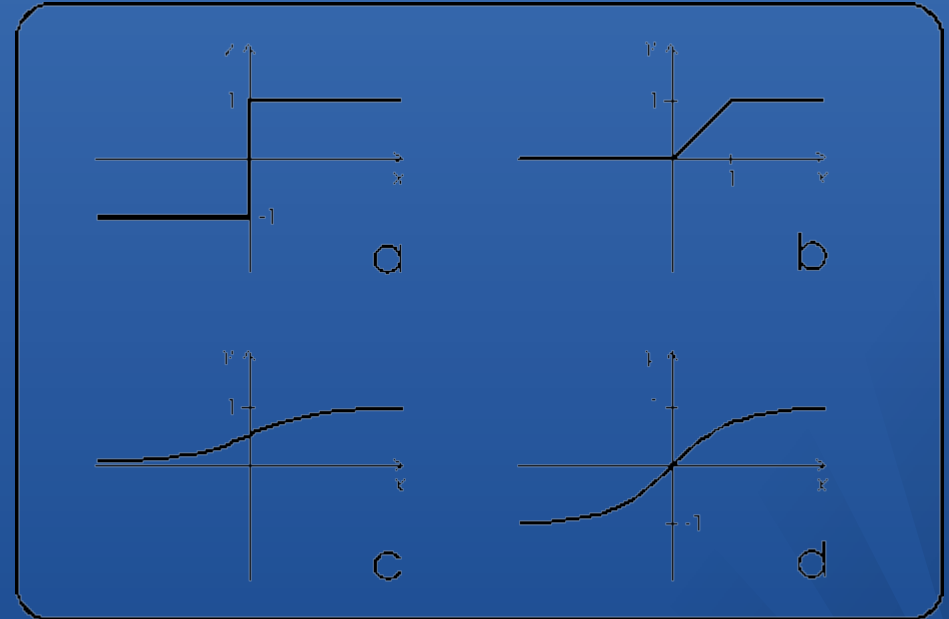
# Architettura

Ora possiamo concepire una ANN (Artificial Neural Network):



# Funzioni di attivazione

- a) funzione Segno ( $\approx$ gradino)
- b) funzione ReLu con saturazione
- c) funzione Sigmoide
- d) funzione Tangente iperbolica

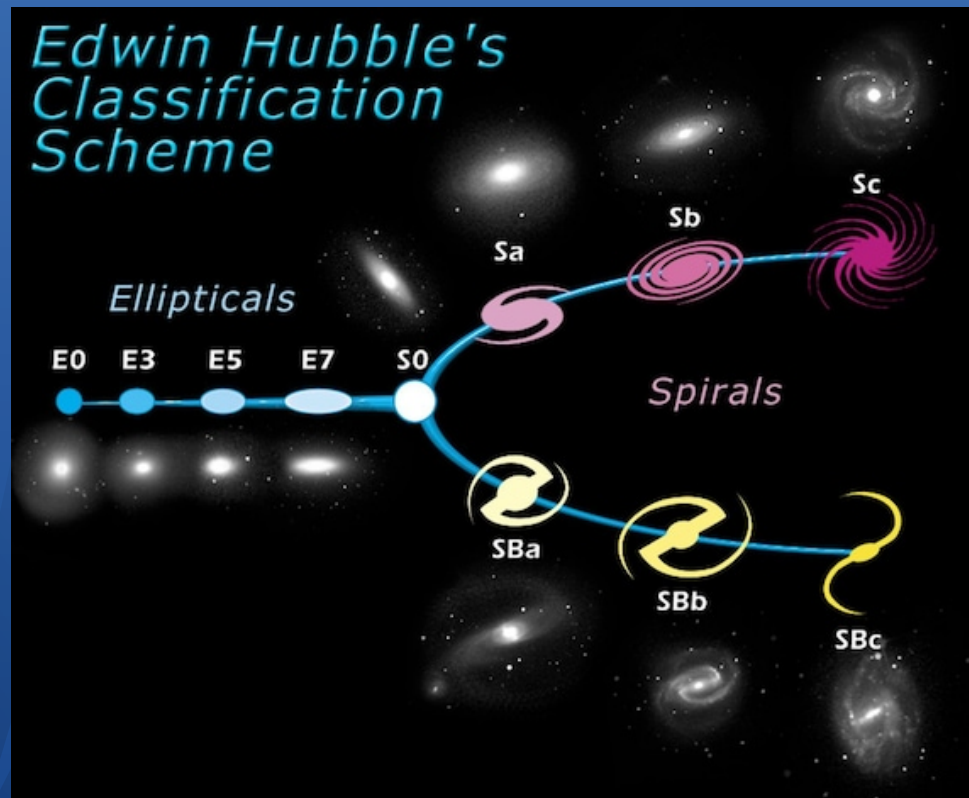


# Applicazioni

- Computer Vision (architettura CNN)
- Previsioni di serie temporali (architettura RNN)
- Large Language Model (architettura Transformer)
- GAN (Generative Adversarial Network)
- Data Mining



# Applicazioni



Credits: ESA

Possiamo addestrare una CNN a riconoscere e classificare le galassie.

Le reti convoluzionali hanno un filtro convoluzionale con pesi modificabili che consentono di ridurre la dimensionalità delle ANN.

# Applicazioni

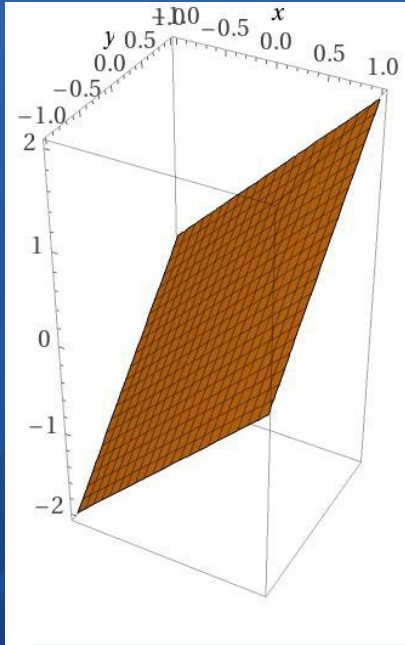
Insegniamo ad una rete neurale a sommare due numeri;  
il modello è:

$$z = x + y$$

# Applicazioni

Insegniamo ad una rete neurale a sommare due numeri;  
il modello è:

$$z = x + y$$



Ci accorgiamo che è una funzione lineare dal grafico. Una rete neurale con due strati è già esagerata, anche un percettrone commetterebbe poco errore.

Si ha un errore basso con dati mai visti ed un errore di estrapolazione basso con numeri lontani da quelli di riferimento del dataset.

# Applicazioni

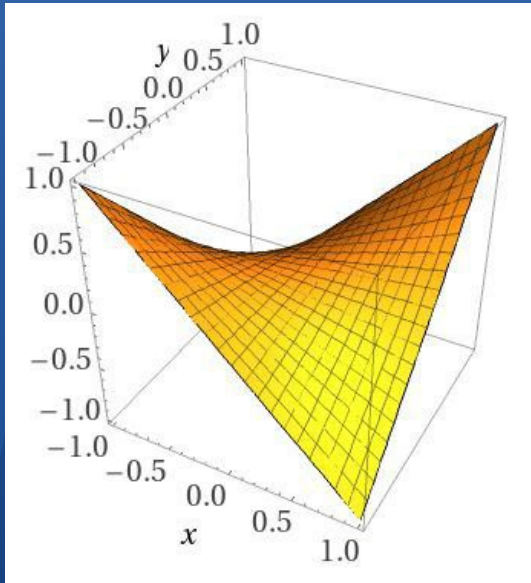
Ora insegniamo a moltiplicarli. Il modello diventa adesso:

$$z = x \cdot y$$

# Applicazioni

Ora insegniamo a moltiplicarli. Il modello diventa adesso:

$$z = x \cdot y$$



La funzione non è più lineare, forma una sella. Una rete neurale con due strati e 3 neuroni funziona, ma un percettrone avrebbe problemi.

Si ha un errore basso con dati mai visti, un errore di estrapolazione alto con numeri lontani da quelli di riferimento del dataset.



COSA NON FA

# COSA NON FA

L'AI pensa come un essere umano?

No.

Classifica, predice, ottimizza.

Nessuna coscienza.

L'unica coscienza che ha è la somma algebrica.

# COSA NON FA

L'AI sa tutto?

No.

Sa solo quello che vede nei dati di addestramento  
(certe volte non sa neanche quelli).

Esempio: quanto pesa in media un gatto?  
Risposta: “dipende dal colore”



# COSA NON FA

L'AI impara da sola?

No.

C'è sempre il controllo umano su:  
dataset, architettura, epoch (tempo di addestramento).

Ma soprattutto c'è il teorema “No Free Lunch”.  
Siamo noi che dobbiamo trovare il miglior modello.

# COSA NON FA

Le reti neurali sono come un cervello?

No.

Le prime sono ispirate al secondo.

Paragonare le reti neurali al cervello è come dire che un forno è come uno chef perché cuoce il cibo.

# COSA NON FA

L'AI è oggettiva?

No.

È più di quanto soggettivo esista.

È un riflesso dei dati scelti per creare il dataset.

# COSA NON FA

L'AI capisce il contesto?

Si.

Solo statisticamente però.

# COSA NON FA

L'AI sostituirà l'uomo?

Forse.

In alcune mansioni sì, ma neanche in quelle realmente.

Esempio: AIEA (Agenzia Internazionale per l'Energia Atomica)

# COSA NON FA

Basta dargli tanti dati e funziona?

No.

Non è solo questione di quantità, ma anche di qualità.  
Di preprocessing, di prove ed errori con l'architettura.

# COSA NON FA

Il deep learning è la soluzione a tutto?

No.

È potente, ma non sempre adatto.  
Per la maggiore è come sparare ad una mosca con un bazooka.

# COSA NON FA

L'AI è neutra perché è matematica?

No.

Le scelte sui dati sono umane.  
L'AI eredita i nostri bias umani.



# COSA NON FA

L'AI è artistica?

No.

Non genera con intenzione.

Nessuna poetica.

È un'artista come una fotocopiatrice ispirata.

Gli artisti più colpiti sono i più mediocri paradossalmente.

# COSA NON FA

Un'opera fatta dall'AI è libera dal copyright?

Non automaticamente.

Se è stata addestrata su materiale coperto da diritti,  
anche il suo output ne eredita le caratteristiche.

Può violare il diritto d'autore anche se l'output sembra nuovo.

Esempio: “ti faccio copiare... ma cambia un po il testo”

# COSA NON FA

“Ma ChatGPT parla!”

# COSA NON FA

“Ma ChatGPT parla!”



“The ability to speak does not make you intelligent,  
now get out of here.”

Fine

Grazie per l'attenzione.