

# **Metodi Numerici per L'intelligenza Artificiale.**

Andrea Cecchini

28 febbraio 2023

# Indice

<b>1</b>	<b>Introduzione all'Analisi Numerica.</b>	<b>3</b>
1.1	Analisi Numerica. . . . .	3
1.1.1	Fasi della risoluzione di un problema numerico. . . . .	3
1.1.2	Errori nel risolvere un problema numerico. . . . .	3
1.2	Classificazione dei problemi numerici . . . . .	4
1.2.1	Problema numerico . . . . .	4
1.2.2	Classificazione dei problemi numerici. . . . .	4
<b>2</b>	<b>Introduzione all'Intelligenza artificiale</b>	<b>5</b>
2.1	Cambio di paradigma di programmazione . . . . .	6
2.1.1	Paradigma classico . . . . .	6
2.1.2	Paradigma del Machine Learning . . . . .	6
2.2	I dati . . . . .	8
2.2.1	Acquisizione dei dati . . . . .	8
2.2.2	Annotazione dei dati . . . . .	8
2.2.3	Organizzazione dei dati . . . . .	8
2.3	Complessità dei dati . . . . .	9
2.3.1	Feature extraction . . . . .	9
2.4	Machine Learning tasks . . . . .	11
2.4.1	Classificazione . . . . .	11
2.4.2	Regressione . . . . .	11
2.4.3	Clustering . . . . .	12
2.5	Loss Function . . . . .	12
2.6	Tipi di apprendimento . . . . .	12
2.6.1	Apprendimento supervisionato . . . . .	12
2.6.2	Apprendimento non supervisionato . . . . .	12
2.6.3	Apprendimento con rinforzo . . . . .	12
<b>3</b>	<b>Artificial Neural Networks</b>	<b>13</b>
3.1	Ispirazione dal modello umano . . . . .	13
3.2	Neurone artificiale . . . . .	13
3.3	Funzione di attivazione . . . . .	15

# 1 Introduzione all'Analisi Numerica.

## 1.1 Analisi Numerica.

Introduciamo nel definire il compito dell'analisi numerica.

**Analisi Numerica.**

L'Analisi Numerica è la parte di matematica che si occupa di dare una **risposta numerica** ad un problema matematico che modella un problema reale.

### 1.1.1 Fasi della risoluzione di un problema numerico.

Al fine di raggiungere tale problema, ci avvaliamo delle seguenti fasi:

- **Tradurre** il problema reale in un insieme di equazioni matematiche in grado di descriverlo
- **Trasformare** il problema matematico nel continuo in un problema numerico discreto che sia risolubile.
- **Trasportare** il problema discreto in un calcolatore mediante l'applicazione di algoritmi numerici capaci di determinare la soluzione in un tempo ottimale.
- **Interpretare** la soluzione numerica nei termini della situazione reale e **verificare** così sia l'adeguatezza del modello matematico sia l'efficienza dell'algoritmo risolutivo.

### 1.1.2 Errori nel risolvere un problema numerico.

Nel percorso appena descritto vi possono essere numerosi errori, le quali sorgenti sono:

- **Errori nel modello matematico** Nascono da una cattiva traduzione del problema reale a quello matematico, per esempio si considerano alcune cose come trascurabili quando non lo sono.
- **Errori nel modello numerico-computazionale** Vengono descritti come errori di *discretizzazione* o *troncamento*.
- **Errori presenti nei dati** Nati da uno strumento di misurazione fallace o da misurazioni che possono essere influenzate da errori sistematici.
- **Errori di arrotondamento nei dati e nei calcoli** Sono gli errori introdotti nella rappresentazione dei numeri sul calcolatore.

## 1.2 Classificazione dei problemi numerici

### 1.2.1 Problema numerico

**Problema Numerico.**

Per **problema numerico** intendiamo una descrizione chiara di una **relazione funzionale** tra i dati (**input**) e i risultati (**output**).

In particolare, in un problema numerico abbiamo i seguenti elementi:

- **F** rappresenta la relazione funzionale tra input ed output.
- **x** rappresenta il dato di input della relazione funzionale.
- **y** rappresenta l'output della funzione di un determinato input

### 1.2.2 Classificazione dei problemi numerici.

Descritti questi 3 elementi, è possibile classificare il problema numerico in base a cosa stiamo cercando:

- **Problema diretto** **F** e **x** sono dati, bisogna **trovare y**.
- **Problema inverso** **F** e **y** sono dati, bisogna **trovare x**.
- **Problema di identificazione** **x** e **y** sono noti, bisogna trovare **F**.

Quest'ultimo problema è quello che interesserà di più durante il corso, perchè è proprio il problema numerico che l'intelligenza artificiale cerca di risolvere.

#### Summary

Abbiamo introdotto la materia dell'**analisi numerica** e quello che si prefissa di risolvere. Successivamente abbiamo definito il concetto di **problema numerico** ed abbiamo elencato i diversi tipi, quali **problema diretto**, **problema inverso** e **problema di identificazione**.

## 2 Introduzione all'Intelligenza artificiale

**Intelligenza Artificiale.**

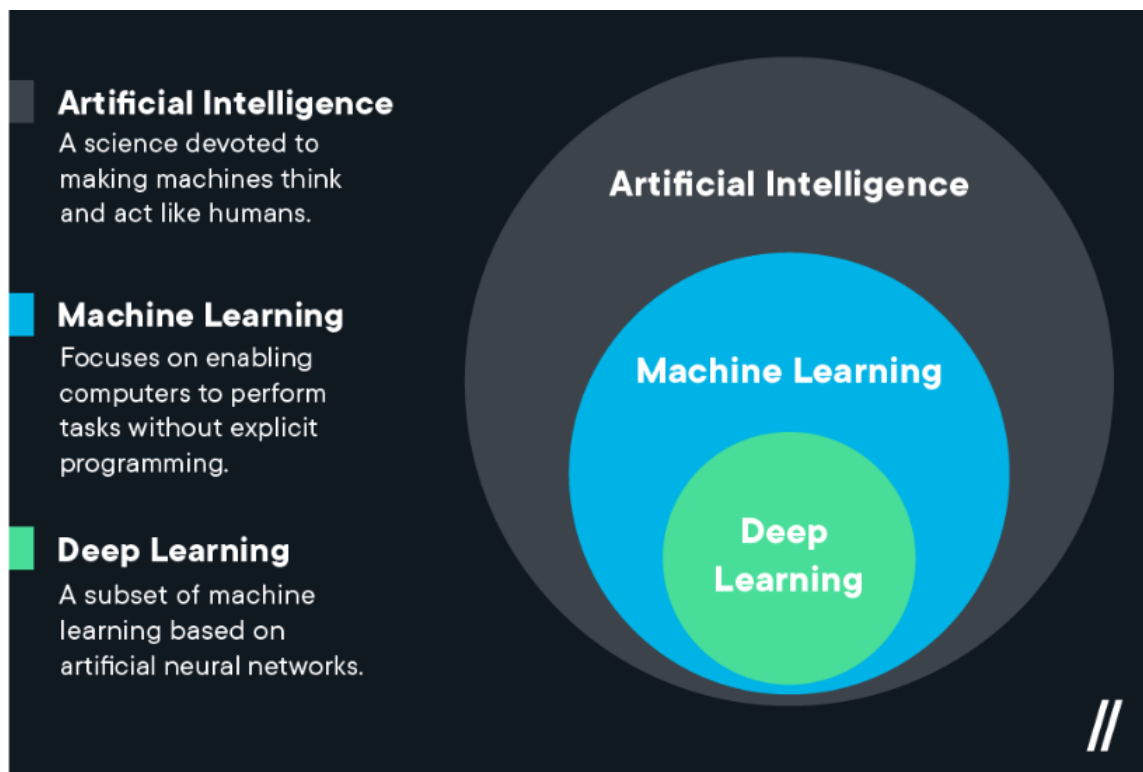
Per **intelligenza artificiale** si intende una parziale riproduzione dell'attività intellettuale propria dell'uomo.

Esistono due tipi di intelligenze artificiali, basate sul loro dominio applicativo:

- **Intelligenza artificiale “debole”**: Sono dei sistemi basati per risolvere **problemi specifici**.
- **Intelligenza artificiale “forte”**: Sono dei sistemi in grado di replicare **tutte le funzioni cognitive** dell'essere umano. Spesso per riferirci a questa categoria useremo il termine **“Intelligenza Generalista”**.

Molti tendono a confondere e a non capire il legame tra **intelligenza artificiale**, **machine learning** e **deep learning**.

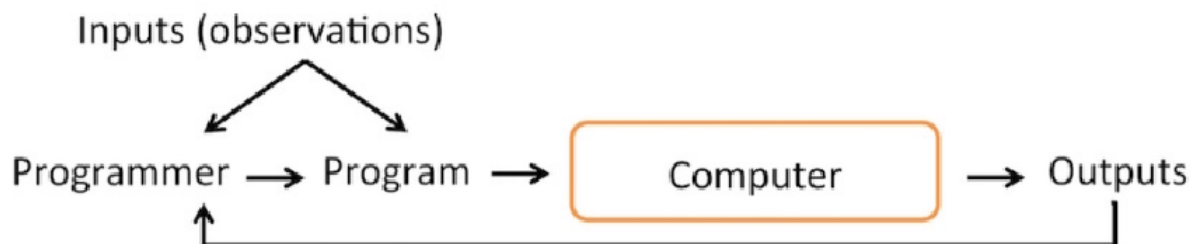
Rappresenteremo il loro legame attraverso il seguente diagramma di Venn:



Detto ciò, capiamo cosa cambia grazie all'utilizzo di questa tecnologia.

## 2.1 Cambio di paradigma di programmazione

### 2.1.1 Paradigma classico

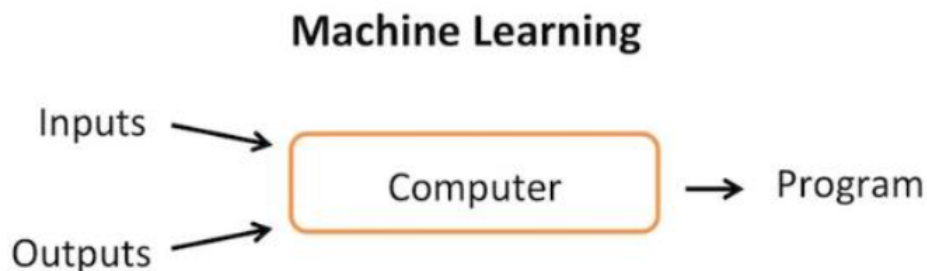


Il paradigma sul quale noi siamo abituati a creare programmi è il seguente:

- Il **programmatore** elabora e crea un **algoritmo** (programma).
- All'algoritmo vengono forniti dei dati come **input**.
- L'elaboratore computa l'input sulla base dell'algoritmo e fornisce un output.

Questo modo di agire prevede una forte presenza dell'essere umano, il quale in veste di programmatore, crea l'algoritmo voluto.

### 2.1.2 Paradigma del Machine Learning



In questo paradigma il ruolo dell'uomo viene "sostituito" dalla tecnica del **Machine Learning**.

Machine Learning.

Sistema in grado di apprendere automaticamente da esempi specifici (**training data**) e di generalizzare la conoscenza su nuovi campioni (**test data**) dello stesso dominio

Difatti, il machine learning dato input ed output, procede nella risoluzione del problema di identificazione.

Andiamo ad analizzare le diverse fasi di questo paradigma:



- **Acquisizione dati** I dati sono l'elemento base di tutte le applicazioni di M.L.. E' molto importante quindi sapere come acquisire questi dati.
- **Data processing** I dati raccolti nella fase precedente vengono processati al fine di attarli al meglio al modello M.L. che intendiamo sviluppare.
- **Modello** Insieme di tecniche matematiche e statistiche in grado di apprendere da una certa distribuzione di dati.
- **Predizione** Una volta ottenuto il modello è possibile "predire" l'output correlato ad un certo input non presentato nel training data.

## 2.2 I dati

### 2.2.1 Acquisizione dei dati

E' possibile ottenere i dati in due modi:

- Usare set di dati pubblici: sono presenti molte piattaforme, come kaggle.
- Acquisito un nuovo set di dati.

E' molto comune nel mondo della ricerca di fornire questi set di dati pubblici, in modo altruista. Visto ciò, non usarli sarebbe un peccato.

### 2.2.2 Annotazione dei dati

Etichetta.

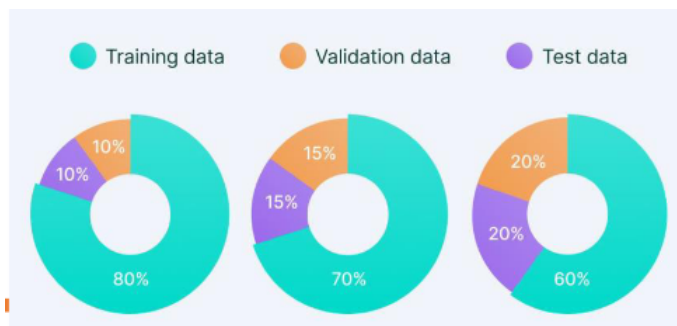
Annotare i dati vuol dire assegnare un **etichetta** (output) ad una determinata istanza di input. L'etichetta rappresenta il contenuto semantico dei dati.

Diremo quindi che un dato è **annotato** se associato ad una **etichetta**. I dati non annotati sono spesso **inutili**. Tuttavia, grazie alla tecnica di apprendimento **non supervisionata** (vedremo dopo) è possibile comunque estrarre conoscenza da essi.

### 2.2.3 Organizzazione dei dati

Bisogna **organizzare** i dati come segue:

- **Training set** sono i dati sui quali il modello apprende automaticamente durante la fase di apprendimento.
- **Validation set** sottoinsieme del training set, sono i dati con il quali si informa il sistema della validazione del suo apprendimento.
- **Testing set** dati con il quali si testa il modello. Questa fase verifica l'efficacia del modello, anche attraverso misure numeriche qualitative e quantitative.



Nell'immagine qui sopra si elenca differenti proporzioni in cui si dovrebbe suddividere il set di dati che abbiamo nei subset descritti precedentemente.



## 2.3 Complessità dei dati

### Dimensionalità.

La **dimensionalità** di un dato rappresenta la **densità** di quest'ultimo, ovvero la quantità.

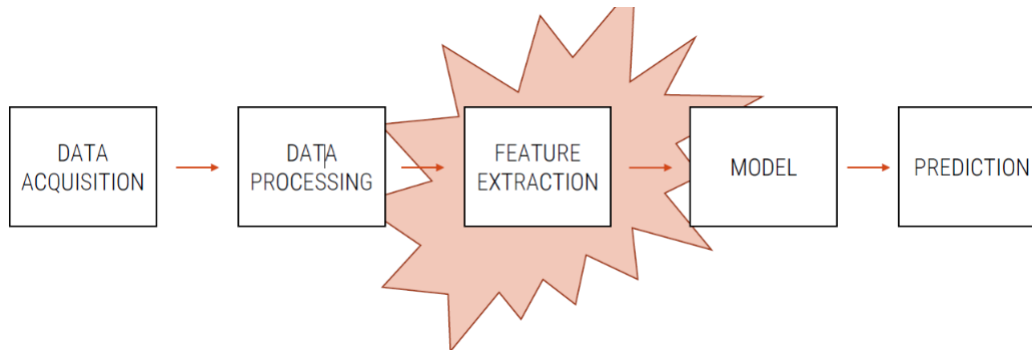
### Complessità.

Diremo che un dato è complesso se presenta una **alta dimensionalità**.

Dare al sistema di M.L. una mole sproporzionata di dati, come tutti i pixel di un'immagine, non è una buona cosa.

Se stessimo lavorando su un classificatore di immagini, sarebbe un errore grave dargli dati ad alta dimensionalità, in quanto non riuscirebbe ad apprendere da così tanti dati **inutili**.

La soluzione a questo problema si chiama **feature extraction**.



### 2.3.1 Feature extraction

#### Feature.

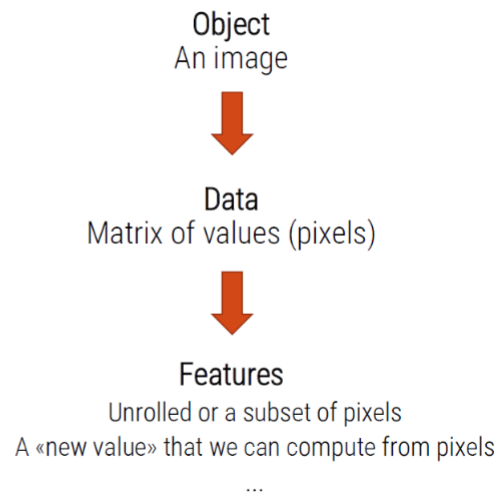
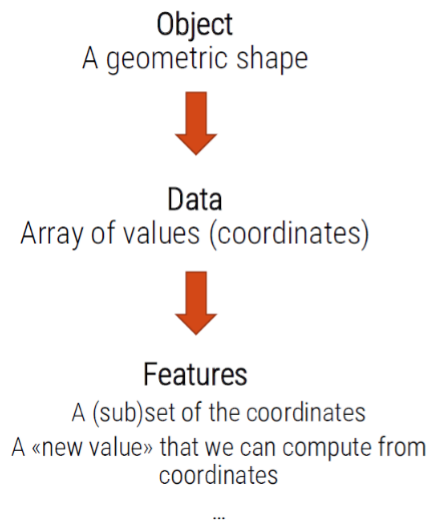
La **feature** rappresenta la parte più utile del dato grezzo.

#### Feature Expansion.

Rappresenta l'operazione di estrazione di features dal dato grezzo.  
E' un modo per creare un nuovo e più piccolo insieme di dati che cattura la maggiore parte dell'informazione dei dati grezzi.

#### Feature Descriptor.

Un **feature descriptor** rappresenta un vettore  $n$  dimensionale di feature numeriche che rappresentano qualche oggetto.



## 2.4 Machine Learning tasks

Il machine learning offre diversi **task** a seconda dell'output che vogliamo

- **Classificazione**
- **Regressione**
- **Clustering**

### 2.4.1 Classificazione

**Classe.**

Una classe è un **set di dati con proprietà comuni**.

Il concetto di classe è correlato al concetto di "etichetta".

**Classificazione.**

Dato un input specifico, il modello (**classificatore**) emette una **classe**.

- Se ci sono 2 classi, chiamiamo il problema come **problema di classificazione binaria**
- Se ci sono  $n$  classi con  $n > 2$ , chiamiamo il problema come **problema di classificazione multiclasse**.

### 2.4.2 Regressione

**Regressione.**

La **relazione** viene utilizzata per **modellare la relazione** tra le variabili indipendenti e le variabili dipendenti.

Quindi la regressione si occupa di risolvere un **problema di identificazione**.

**Regressione Multi-Variata.**

La **regressione multivariata** prevede l'impiego di più variabili in gioco rispetto alla classica regressione lineare.

### 2.4.3 Clustering

#### Clustering.

Il **clustering** permette di identificare dei gruppi di dati in classi, senza saper a priori le classi.

Il clustering è spesso applicato, infatti, in un ambiente di apprendimento non supervisionato, in cui le classi del problema non sono noti a priori.

## 2.5 Loss Function

Facciamo una piccolissima diramazione di poche righe per spiegare il concetto di **loss function**.

#### Loss Function.

La **loss function** (funzione di perdita) è un indicatore che permette di descrivere la qualità dell'apprendimento del sistema.

Al fine di aver un sistema che appreso in maniera esaustiva, bisogna far tendere verso il basso il valore della **loss function**.

## 2.6 Tipi di apprendimento

### 2.6.1 Apprendimento supervisionato

### 2.6.2 Apprendimento non supervisionato

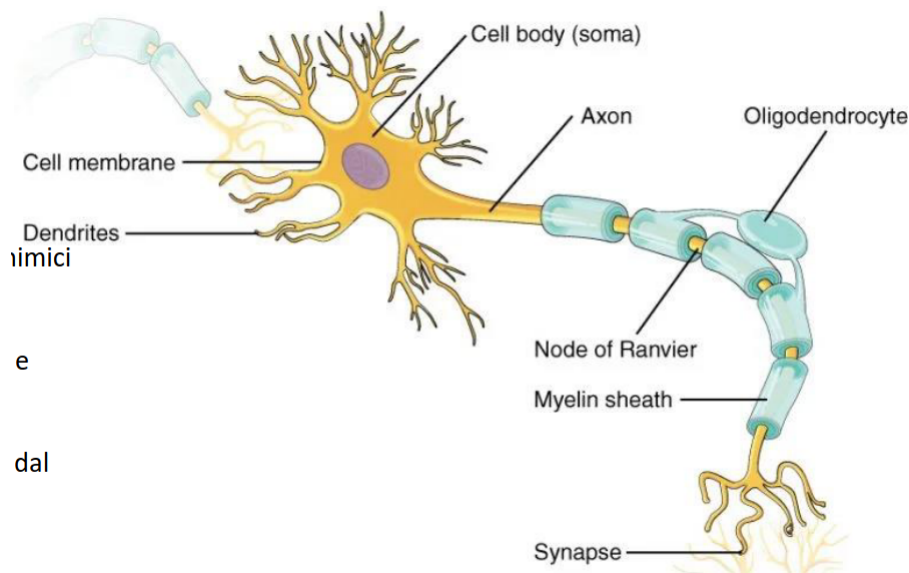
### 2.6.3 Apprendimento con rinforzo

## 3 Artificial Neural Networks

### 3.1 Ispirazione dal modello umano

Il paradigma delle reti neurali artificiali e' ispirato al modo in cui il sistema nervoso biologico elabora le informazioni.

L'unita' elementare del nostro sistema nervoso si chiama **Neurone**.



Le parti piu' importanti di un neurone sono:

- **Dentriti** sono i canali di ricezione dell'input proveniente da altri neuroni.
- **Soma** E la parte "core" del neurone.  
Essa elabora l'input nel tempo e converte il valore elaborato in output.
- **Assone** Rappresenta il canale di trasferimento dell'output.  
L'assone trasferisce l'informazione elaborato dal soma ad altri neuroni.
- **Sinapsi** sono le estremita' dell'assone, le quali funzionano come dei ganci verso i dentriti di altri neuroni.

Quando il soma riceve segnali in ingresso, e se quest'ultimi superano una certa **soglia**, viene prodotto un segnale di uscita sull'assone che diventera' un segnale di input per altri neuroni. **Il valore di questa soglia e l'efficienza chimica di trasmissione delle sinapsi sono strettamente legate ai processi di apprendimento.** Nella testa l'indicatore **wtf-per-minute** sara' andato alle stelle con questa lezione di biologia, ma non vi preoccupate, rispetto a dopo e' ancora basso.

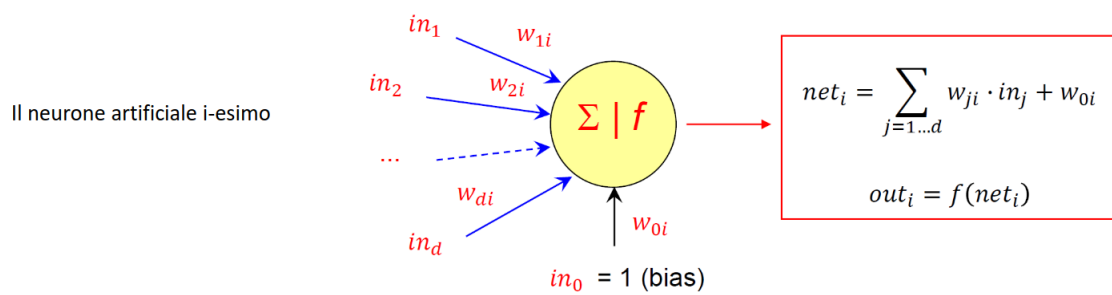
### 3.2 Neurone artificiale

Le reti neurali sono un tentativo dell'uomo di creare una rappresentazione informatica del nostro sistema nervoso. L'elemento base di esso, ovvero neuroni, vengono usati come caso di studio e base comportamentale per la creazione dell'elemento base delle reti neurali: il **Neurone artificiale**.

La prima rivisitazione rappresenta le basi di rappresentazione dei segnali di ingresso e uscita.

Tali segnali verranno rappresentati in binario.

Il primo modello di neurone artificiale, creato da McCulloch e Pitts, e' in grado di eseguire computazioni logiche dato dei segnali di input.



L'indicatore wtf-per-minute si e' alzato, almeno nella mia testa, quindi cerchiamo di svelare il mistero nel geroffico.

Per prima cosa andiamo ad analizzare i vari attore presenti nel disegno superiore:

- **i** rappresenta l'**identificativo del neurone corrente**.
- **in<sub>j</sub>** con  $j \leq d$  rappresenta il **segnale di ingresso j-esimo**.
- **w<sub>ji</sub>** rappresenta il **peso** (approssimiamolo con il termine "**importanza**" per ora) di un certo segnali di input.  
Vale a dire che  $w_{1i}$  sara' il peso assegnato al primo segnale di ingresso del neurone i.
- **in<sub>0</sub>** e **w<sub>0i</sub>** vengono denominato come **bias**, ovvero del **peso** collegato ad un **input fittizio**  $in_0 = 1$ .  
La sua utilita' sta nel **calibrare** il punto ottimale di lavoro di un neurone.
- **net<sub>i</sub>** e' il **livello di eccitazione** del neurone i-esimo.  
Avremo che  $net_i = \sum_{j=0}^d w_{ji} * in_j + w_{0i}$  dove  $w_{ji} * in_j$  rappresenta il segnale di ingresso nella sua totalita' (considerando cosi' il suo peso).  
Nella sommatoria troviamo anche la somma  $w_{0i}$ , ovvero il bias utile per tarare.
- **f()** e' la **funzine di attivazione** che determina il comportamento di un neurone.  
Esamineremo dopo i dettagli di questa funzione. Avremo che **out<sub>i</sub> = f(net<sub>i</sub>)**.

Visto che risulta essere ancora abbastanza completato capiamo il flusso

Vengono acquisiti i vari segnali.

I vari segnali in ingresso vengono acquisiti e trasportati nella "soma" artificiale.

I segnali vengono "pesati".

Ad ogni segnali in ingresso corrisponde un certo peso.

Si calcola il livello di eccitazione.

Si calcola l'eccitazione del neurone sommando tutti i segnali di ingresso moltiplicati per il loro rispettivo peso, sommati a loro volta al bias.

La funzione di attivazione calcola l'output.

In base al livello di eccitazione del neurone la funzione di attivazione calcola l'output.  
Se  $net_i > soglia$  allora viene prodotto un output, altrimenti ne viene prodotto un altro.

### 3.3 Funzione di attivazione

Abbiamo osservato come la **funzione di attivazione** trasforma il livello di eccitazione del neurone in un plausibile output. Facciamo un esempio allora:

$$F(net_i) = \begin{cases} 1 & \text{se } net_i > soglia \\ -1 & \text{se } net_i < soglia \end{cases}$$