

LEZIONE 8

INTRODUZIONE AL MACHINE LEARNING

Artificial Intelligence

Chat GPT: Cos'è l'Intelligenza Artificiale ?

L'Intelligenza Artificiale (AI) è un campo dell'informatica e dell'ingegneria che si concentra sulla creazione di sistemi e programmi in grado di eseguire attività che **richiedono intelligenza umana**, come la comprensione del linguaggio naturale, la percezione sensoriale, la ragionamento, il problem solving e la capacità di apprendimento.

L'obiettivo dell'intelligenza artificiale è quello di creare macchine che possano replicare le funzioni cerebrali dell'uomo, in modo da poter **risolvere problemi in modo autonomo e adattarsi alle nuove situazioni**. In altre parole, l'AI mira a creare macchine **"intelligenti" in grado di apprendere da sé e migliorare le loro prestazioni nel tempo**.

Machine Learning

Il Machine Learning (ML) è una branca dell'intelligenza artificiale che si concentra sulla creazione di algoritmi e modelli che consentono ai computer di apprendere da dati senza essere esplicitamente programmati per farlo. In altre parole, **il Machine Learning consente ai computer di apprendere in modo autonomo e di migliorare le proprie prestazioni nel tempo**, analizzando grandi quantità di dati e rilevando modelli e relazioni in essi contenuti.

Esempi di applicazioni di ML

- Riconoscimento facciale e di oggetti
- Raccomandazioni di prodotti online
- Previsione di prezzi e mercati finanziari
- Assistenza sanitaria e diagnostica
- Guida autonoma e veicoli aerei senza pilota
- Elaborazione del linguaggio naturale (Natural Language Processing - NLP)
- Monitoraggio e rilevamento di frodi
- Analisi di sentimenti e feedback degli utenti
- Personalizzazione dei contenuti online
- Ottimizzazione della catena di fornitura

Esempi di applicazioni di ML

MACHINE LEARNING DEFINITION

Arthur Samuel (1959): Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed

Tom Mitchell (1998): well-posed Learning Problem: A computer program is said to *learn* from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .

“A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .”

ESERCIZIO: Supponiamo che il tuo programma di posta elettronica controlli quali e-mail contrassegni o non contrassegni come spam e, in base a questo, apprende come filtrare meglio lo spam. Qual è il compito T in questo contesto?

- Classificare le email come spam o non spam.
- Il programma osserva come etichetti le email (spam o non spam).
- Il numero (o frazione) di email correttamente classificate come spam/non spam.
- Nessuno dei precedenti: questo non è un problema di apprendimento automatico.

ALGORITMI DI MACHINE LEARNING

Algoritmi di Machine Learning:

- **Supervised Learning**
- **Unsupervised Learning**

Altri algoritmi: Reinforcement Learning

LEZIONE 8

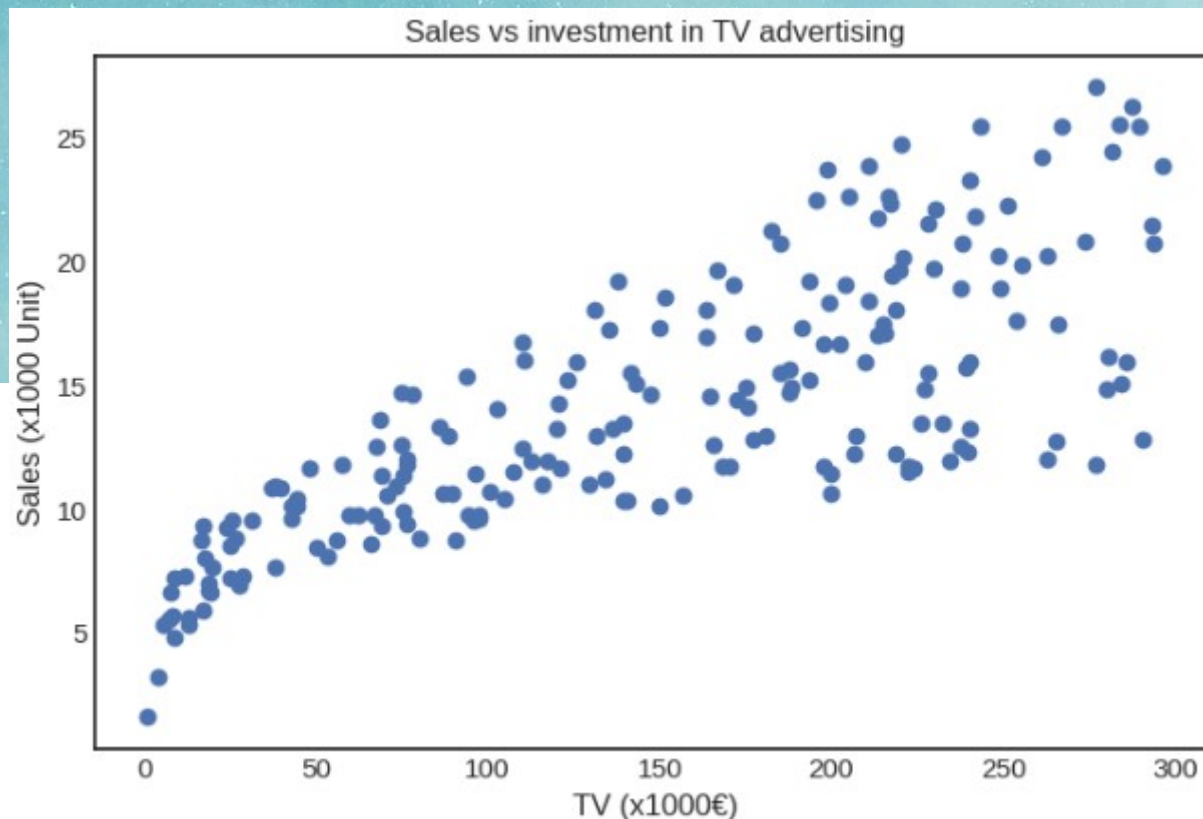
INTRODUZIONE AL SUPERVISED LEARNING

SALES UNITS PREDICTION

DOMANDA?

Quante unità riusciamo a vendere se investiamo 150000€ in pubblicità in TV?

Nel SUPERVISED LEARNING abbiamo già un insieme di dati di partenza con la risposta corretta.



Dato un importo investito in TV advertising, quante unità riesco a vendere?
REGRESSION problem: **predict** continuous value output

IL TRAINING SET

Notazione

m = numero di training examples (200)

x = input variable (input FEATURES)

y = output variable (TARGET variable)

Ogni coppia (TV, Sales) è un TRAINING EXAMPLE.

Per es. il primo training example è la coppia (x^1, y^1) costituita dai numeri (230.1, 22.1), il secondo è (x^2, y^2) che corrisponde a (44.5, 10.4), l' i -esimo è (x^i, y^i)

Sample	TV (x)	Sales (y)
1	230.1	22.1
2	44.5	10.4
3	17.2	9.3
...
199	38.2	7.6
200	232.1	13.4

NOTAZIONE

- Useremo $\mathbf{x}^{(i)}$ per indicare le “input variables” (TV nel nostro esempio), chiamate anche **“INPUT FEATURES”**
- Useremo $\mathbf{y}^{(i)}$ per indicare la “output” variable o **“TARGET VARIABLES”** che stiamo cercando di prevedere (le sales nel nostro esempio).
- Una coppia $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ è chiamato un **“TRAINING EXAMPLE”**
- L’insieme di dati (il “dataset”) che useremo per fare il learning (nel nostro esempio tutte le m righe $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$) è chiamato **“TRAINING SET”**.
- Useremo anche X per indicare lo spazio dei valori di input e Y per indicare lo spazio dei valori di output. Nel nostro esempio, $X = Y = \mathbb{R}$.

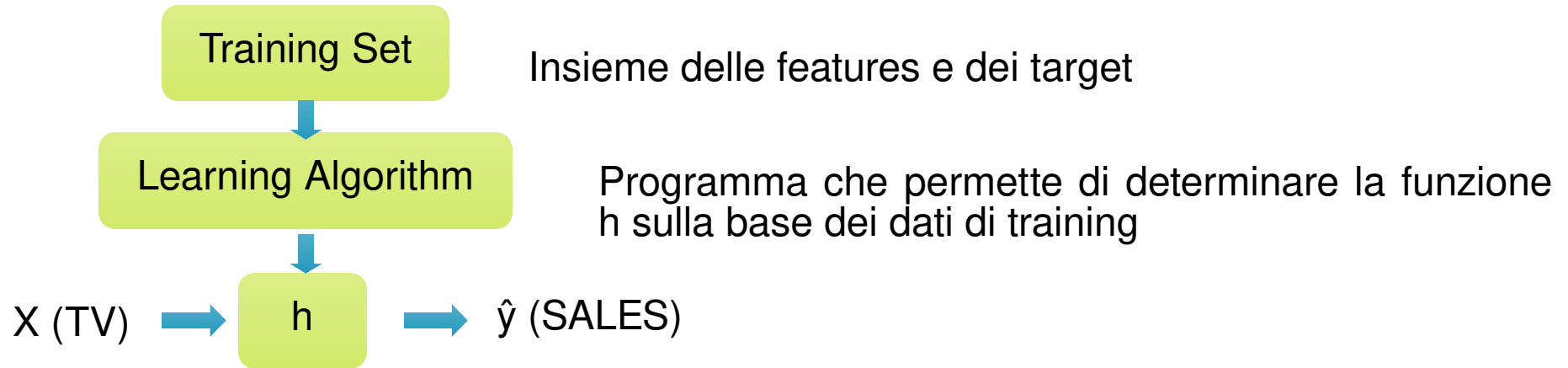
Si noti che “(i)” (superScript nella notazione) è semplicemente un INDICE nel training set e non ha nulla a che fare con l'esponenziazione.

IL MODELLO

Il nostro obiettivo è, dato un **training set**, per mezzo di un **Learning Algorithm**, individuare una **funzione $h: X \rightarrow Y$** in modo che $h(x)$ sia un “**good**” **PREDICTOR**.

Con “good” PREDICTOR intendiamo una funzione che, dato in input la feature x , sia in grado di dare la migliore stima possibile per la risposta y , rispetto ai dati considerati. Chiameremo tale risposta \hat{y} .

Per motivi storici, questa funzione h è chiamata ipotesi (**hypothesis**).



Quando la variabile target che stiamo cercando di prevedere è continua, ad esempio come nel nostro esempio, chiamiamo il learning problem un **regression problem**

IL MODELLO

Abbiamo quindi due fasi:

- 1) **TRAINING**: determinazione della funzione $h: X \rightarrow Y$ per mezzo dei dati a disposizione (in modo che $h(x)$ sia un “**good**” **PREDICTOR**)
- 2) **PREDICTION**: utilizzo della funzione h su un **NUOVO CAMPIONE** per CALCOLARE la migliore previsione possibile, \hat{y}

Consideriamo ora la funzione $\hat{y}=h(x)$.

Scegliamo come primo modello una funzione **LINEARE, PARAMETRICA**.

$$h(x) = \hat{y} = w_0 + w_1 x$$

I numeri w_i sono chiamati **parametri del modello**.

Come scelgo i w_i ? Con questa particolare h , il learning algorithm dovrà essere in grado di calcolare i parametri w_i sulla base del training set.

Come scegliere i parametri

Abbiamo scelto la nostra hypothesis function della forma:

$$h(x) = \hat{y} = w_0 + w_1 x$$

Devo trovare un criterio per scegliere i parametri w_0 e w_1 ?

Supponiamo di avere un training set con n samples: $(x^1, y^1), (x^2, y^2) \dots (x^n, y^n)$

Notiamo che, per **VALORI FISSATI** di w_0 e w_1 , la quantità

$$e^1 = (h(x^1) - y^1).$$

E' l'**ERRORE** che commettiamo nella nostra previsione del primo campione. Ricordiamo che y^i è il valore certo, mentre $h(x^1)$ è il valore calcolato.

Questa quantità può essere negativa o positiva. Per eliminare problemi di segno considero i quadrati delle differenze

$$(e^{(1)})^2 = (h(x^{(1)}) - y^{(1)})^2$$

II MSE

Quindi la quantità $(e^{(i)})^2 = (h(x^{(i)}) - y^{(i)})^2$ è l'errore al quadrato che commetto nel prevedere l'iesimo sample.

Considero ora la **MEDIA** di tutti gli errori del mio sample

$$\text{MSE} = 1/n * ((e^{(1)})^2 + (e^{(2)})^2 + + (e^{(i)})^2 + + (e^{(n)})^2)$$

Chiamerò questo valore **MEAN SQUARE ERROR**

Osserviamo che **MSE DIPENDE DALLA SCELTA DI w_0 , E w_1**

CRITERIO PER LA SCELTA DI w_0 E w_1

Dato un training set con n samples: $(x^1, y^1), (x^2, y^2) \dots (x^n, y^n)$, sceglierò w_0 e w_1 in modo che il valore del **MSE sia il più PICCOLO tra tutti i possibili valori di w_0 e w_1**

Ricapitolando

Siamo partiti dalla nostra hypothesis function della forma:

$$h(x) = \hat{y} = w_0 + w_1 x$$

E abbiamo cercato un criterio per scegliere i parametri w_0 e w_1

Partendo da un training set con n samples: $(x^1, y^1), (x^2, y^2) \dots (x^n, y^n)$

Il criterio adottato è che i parametri w_0 e w_1 devono essere scelti in modo che la quantità

$$\text{MSE} = [(h(x^1) - y^1)^2 + (h(x^2) - y^2)^2 + \dots + (h(x^n) - y^n)^2] / n$$

sia la **più piccola tra tutti i possibili valori di w_0 e w_1**

Osserviamo che la formula non dice altro che: prendiamo la differenza tra il valore calcolato di h e il valore noto di y , facciamo il quadrato e quindi facciamo la media tra tutti questi valori.

La Cost Function

In modo più formale definiamo la funzione COST FUNCTION (Square Error function):

$$J(w_0, w_1) = \frac{1}{2n} \sum_{i=1}^n (h_w(x_i) - y_i)^2$$

E cerchiamo i valori w_0, w_1 che minimizzano tale funzione

$$\min_{(w_0, w_1)} J(w_0, w_1)$$