

21. Reti neurali

Corso di Python per il Calcolo Scientifico

Outline

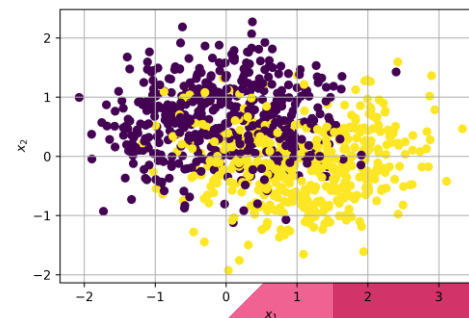
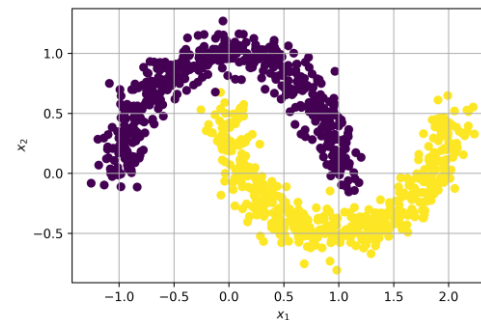
- Problemi lineari e non lineari
- Reti neurali e problemi non lineari
- Funzioni di attivazione

Problemi non lineari

- Non tutti i problemi sono lineari
- I problemi non lineari non possono essere risolti usando dei risolutori del tipo

$$y = ax_1 + bx_2 + c$$

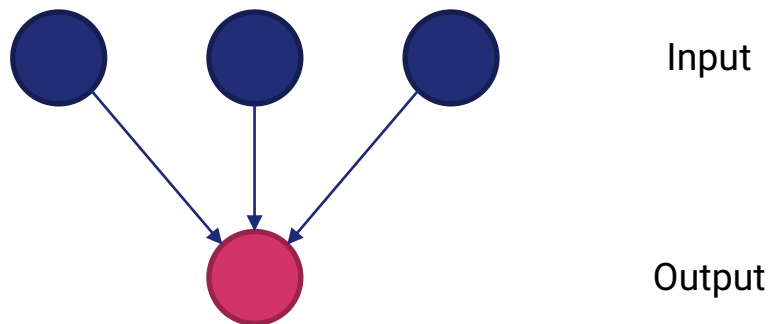
- Ovviamente, la precedente vale soltanto se consideriamo esclusivamente due feature; nel caso di n feature, andranno considerati n termini.



Reti neurali e problemi non lineari

- Immaginiamo un sommatore pesato.
- Ad ogni nodo corrisponde un peso w_i , ed alla rete un bias b .
- Questo tipo di sommatore è in grado di modellare solo problemi lineari:

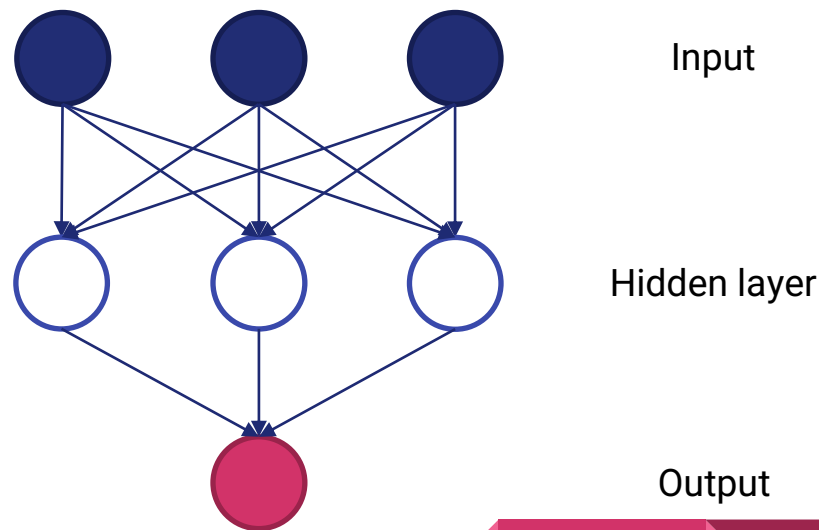
$$y = w_1x_1 + w_2x_2 + w_3x_3 + b$$



Reti neurali e problemi non lineari

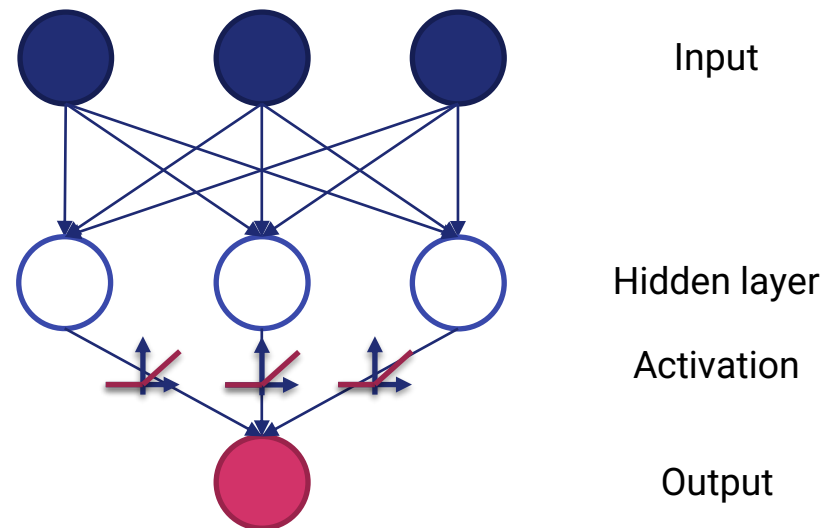
- Inseriamo un altro layer.
- La relazione tra ingresso ed uscita si fa certamente più complessa:
$$x_4 = w_{14}x_1 + w_{24}x_2 + w_{34}x_3$$

...
$$y = w_4x_4 + w_5x_5 + w_6x_6 + b$$
- Tuttavia, **permane una relazione di tipo lineare**, in quanto si tratta sempre di una sommatoria pesata.



Funzioni di attivazione

- Una rete neurale prevede l'uso di una **funzione di attivazione** all'uscita di ciascun neurone.
- La funzione di attivazione ha il ruolo di inserire delle non linearità.
- La più utilizzata attualmente è la **rectified linear unit (ReLU)**.

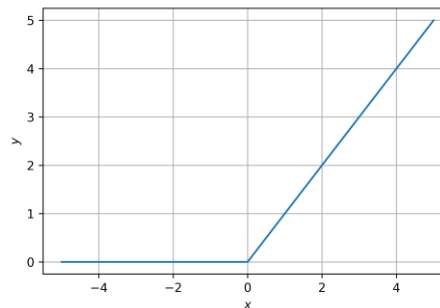


Funzioni di attivazione

- La forma della ReLU è molto semplice, ed è data da:

$$y = \max(0, x)$$

- In pratica, una ReLU porta a 0 tutti i valori in input negativi, lasciando inalterati quelli positivi.



Domande?

42