

# Dal percettrone alle reti multi-strato

Una prima introduzione alle reti neurali



ARTIFICIAL INTELLIGENCE  
& DATA ANALYTICS

[ai.units.it](http://ai.units.it)

# Una breve storia delle reti neurali

**Anno 1958:** Rosenblatt propone il *percettrone* (*perceptron*), che prende ispirazione da precedenti lavori di McCulloch-Pitt.



Frank Rosenblatt  
(1928-1971)



Marvin Minsky  
(1927-2016)



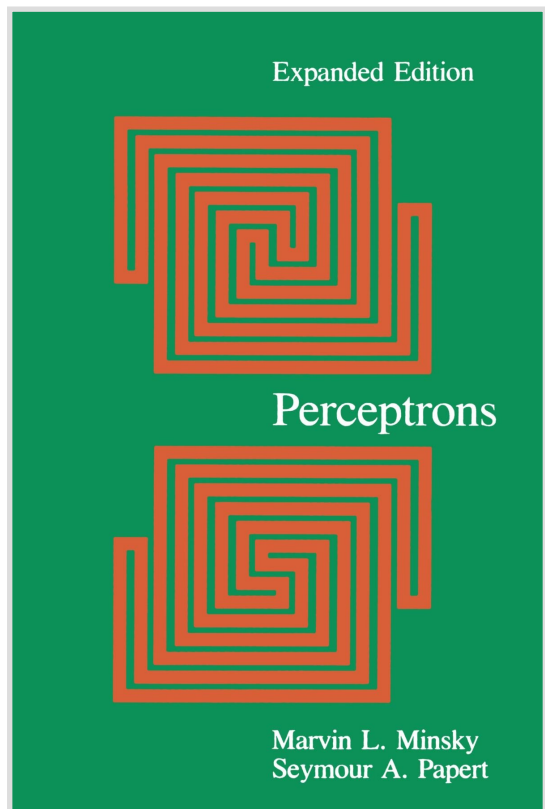
Seymour Papert  
(1928-2016)

Negli anni successivi Minsky e Papert analizzano, formalizzano e studiano il modello del percettrone



# Una breve storia delle reti neurali

Nel 1969 il libro *Perceptrons* di Minsky e Papert mostra i limiti del percettrone



Le studio delle reti neurali proseguirà  
in sordina fino agli anni '80, quando  
avrà una rinascita

Vedremo assieme il percettrone,  
Il suo funzionamento e le sue limitazioni

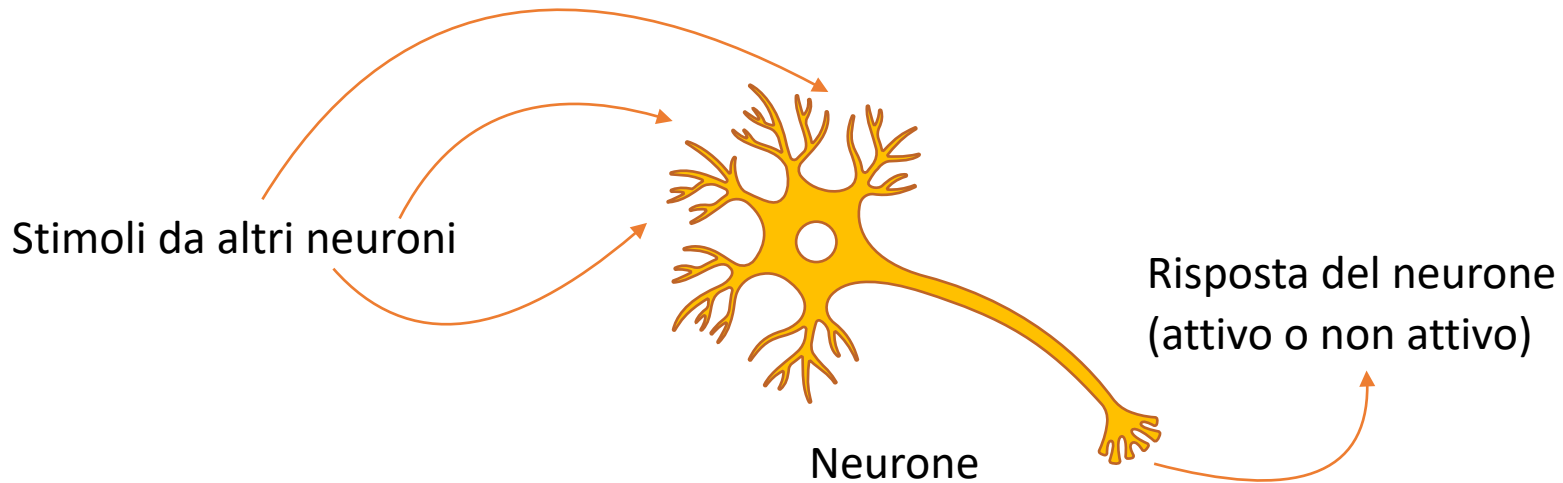


# Il percettore

Partiamo dall'idea di un neurone e sviluppiamo assieme l'idea del percettore

In termini astratti un neurone può essere visto come un oggetto che:

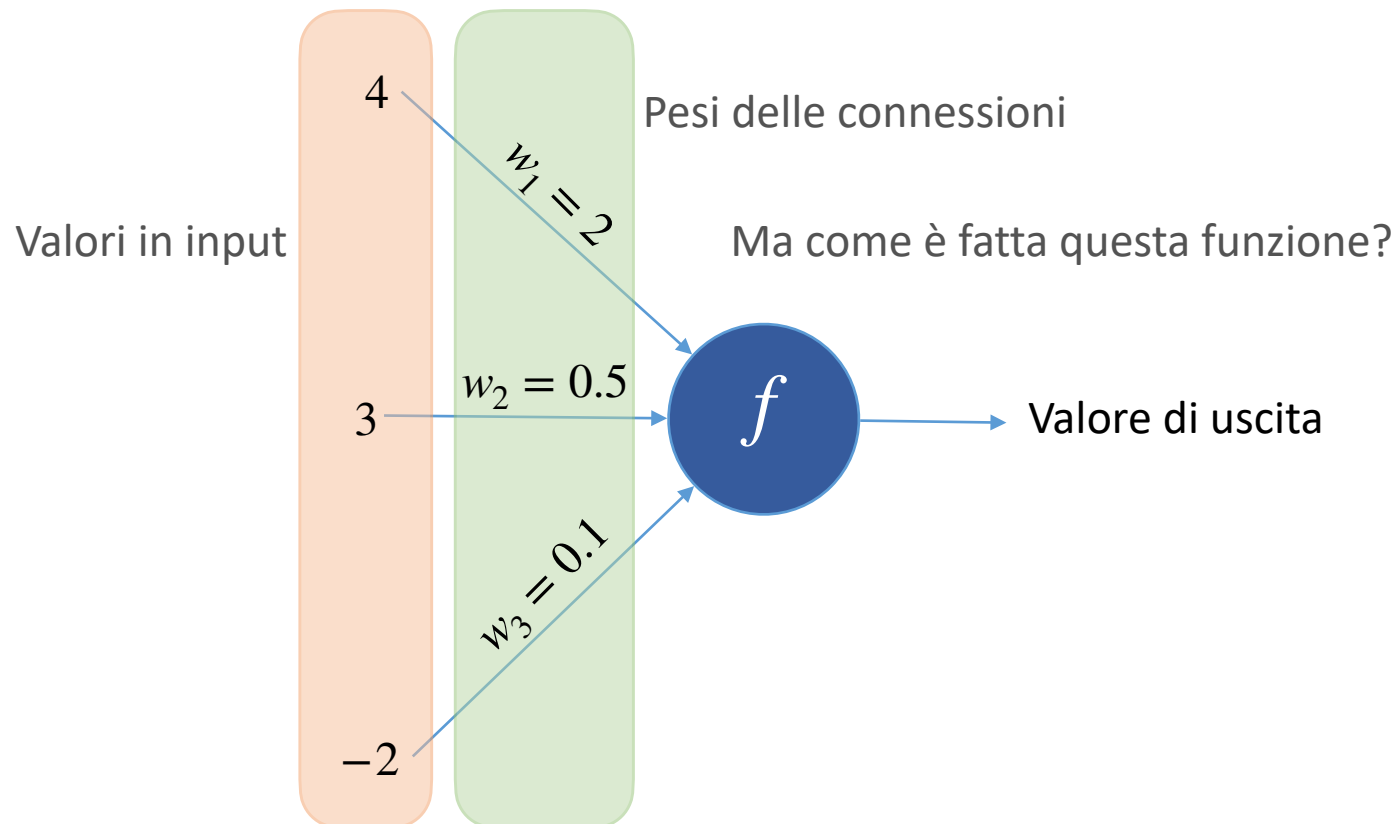
- Riceve degli stimoli da altri neuroni
- A seconda di chi lo invia lo stimolo può essere più o meno importante
- Se i segnali che riceve superano una soglia allora si attiva
- Un neurone può essere attivo o non attivo



# Il percettrone

Semplifichiamo ora il modello all'essenziale

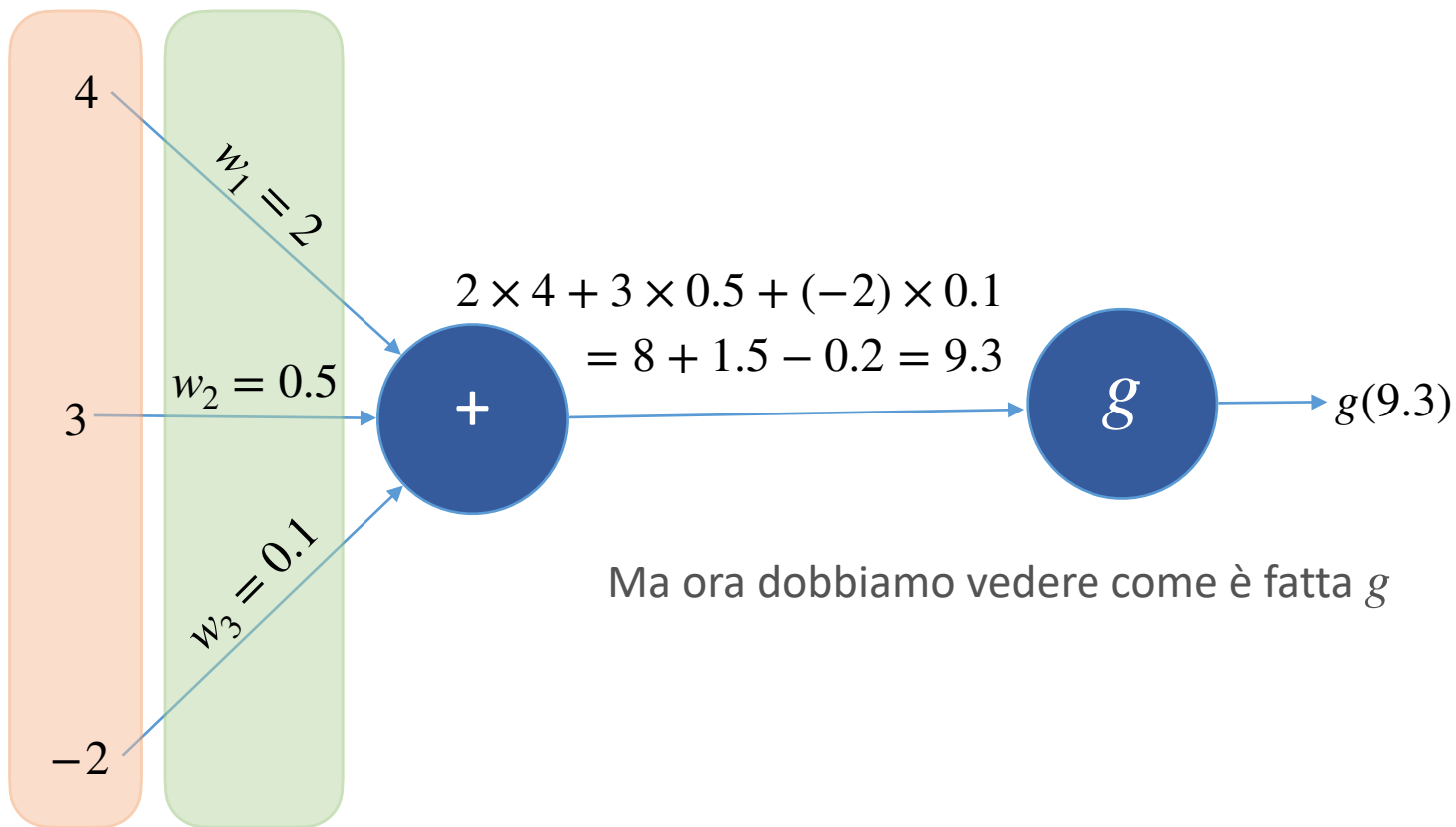
- Stimoli: valori numerici
- Importanza dello stimolo: peso che moltiplica lo stimolo
- Attivazione: una qualche funzione che, presi tutti gli input, genera un output



# Il percettrone

Possiamo separare la funzione  $f$  in due parti:

- Combinazione degli input (solitamente la **somma** di tutti gli input pesati)
- Una funzione di attivazione  $g$  che riceve solo gli input combinati e produce l'output

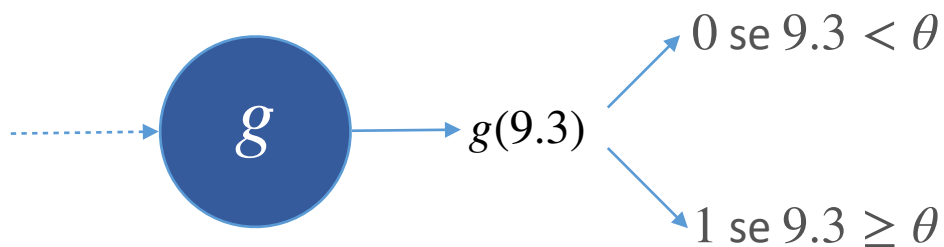
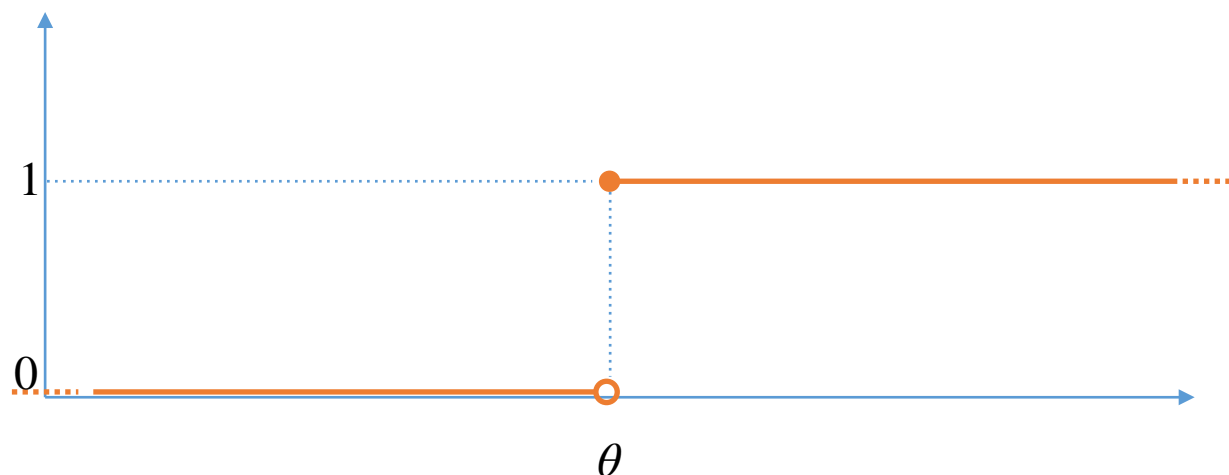


Ma ora dobbiamo vedere come è fatta  $g$



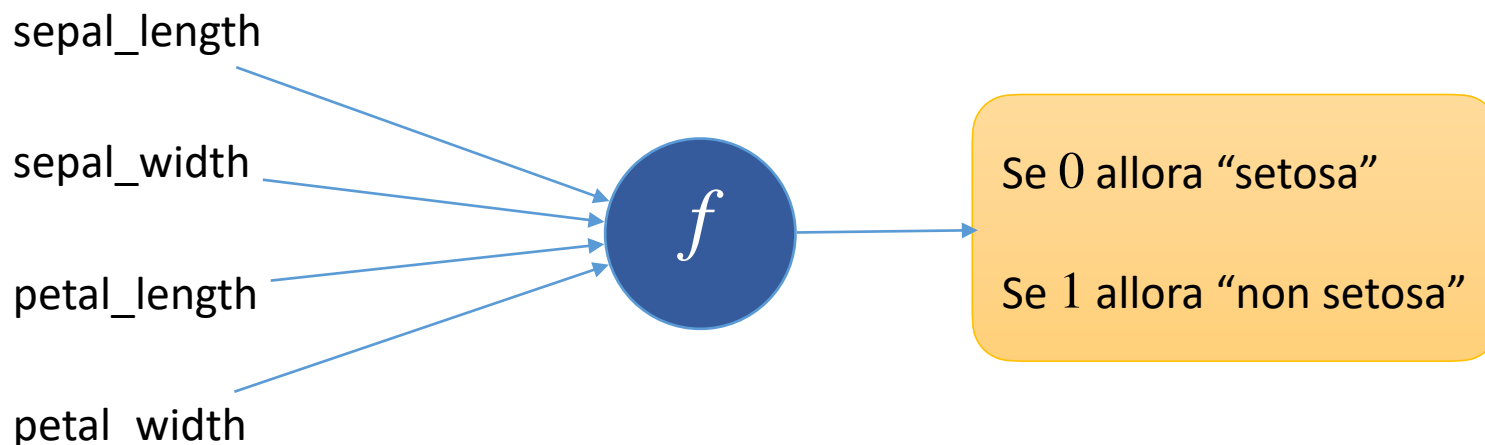
# Funzione di attivazione a soglia

La funzione di attivazione è 0 per tutti i valori sotto una soglia e 1 per tutti i valori sopra la soglia



# Percettrone come classificatore binario

Possiamo usare il percettrone come classificatore binario, ovvero che distingue tra due classi



Notate che, dato che abbiamo solo **due** possibili output non possiamo distinguere tra più di due classi





# Percettrone come classificatore binario

Iniziamo con un problema più semplice con due classi.

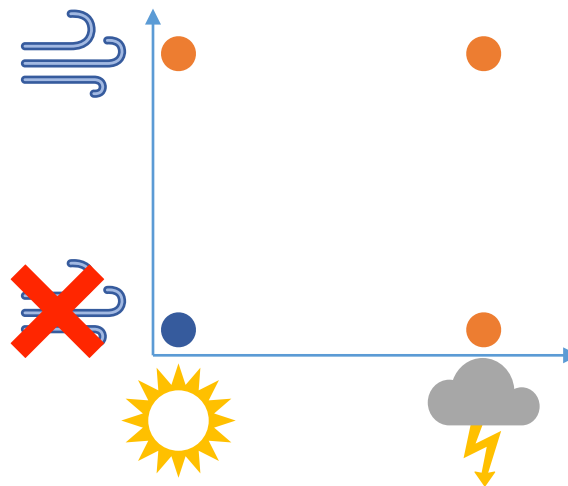
Vogliamo sapere se andare fuori casa a correre in base al tempo.

Abbiamo due variabili in ingresso:

- Pioggia (0 = non piove, 1 = piove)
- Vento (0 = niente o poco vento, 1 = molto vento)

E abbiamo due classi

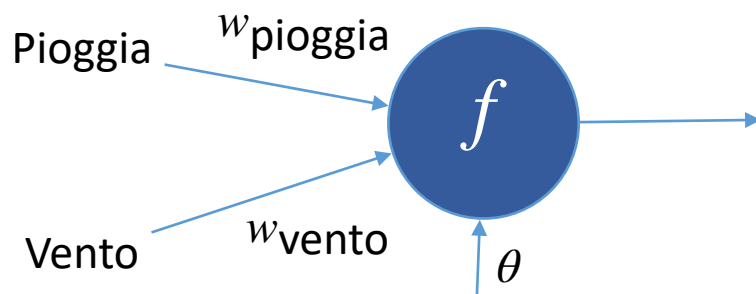
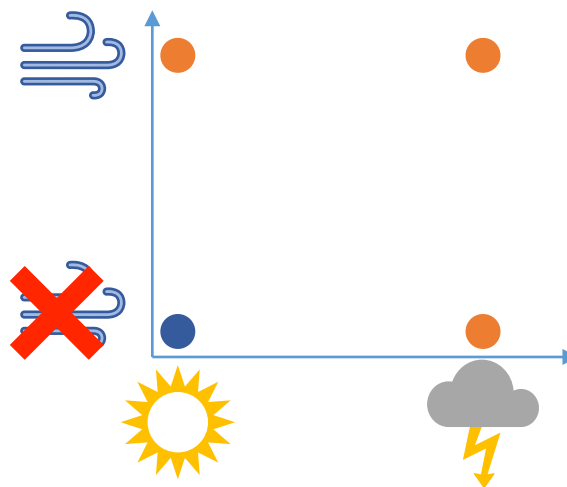
- **Non si corre (0)**
- **Si corre (1)**



# Percettrone come classificatore binario

Proviamo a vedere come possiamo definire un percettrone che ci permetta di classificare correttamente.

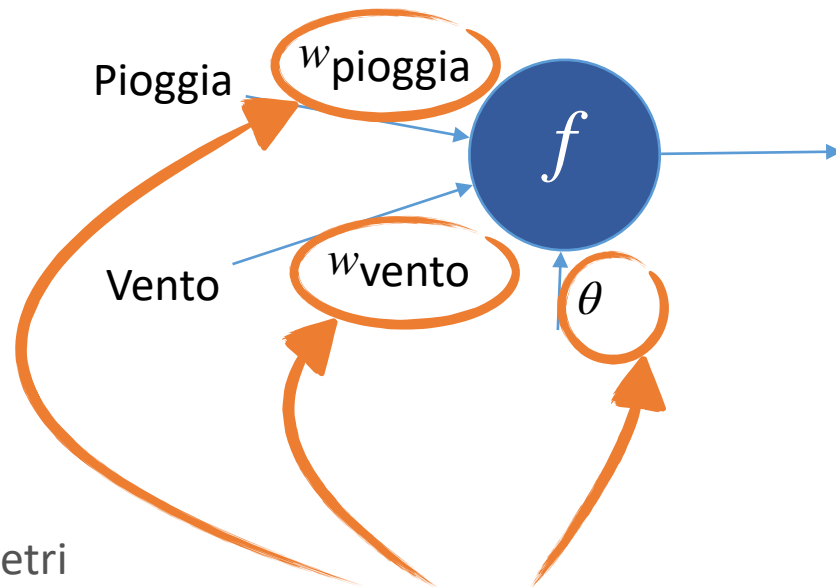
|       |   | Pioggia |   |
|-------|---|---------|---|
|       |   | 0       | 1 |
| Vento | 0 | 1       | 0 |
|       | 1 | 0       | 0 |



# Percettrone come classificatore binario

Vogliamo che  $w_{\text{pioggia}} \times \text{pioggia} + w_{\text{vento}} \times \text{vento} \geq \theta$   
sia vero **solo** quando pioggia = 0 e vento = 0

|       |   | Pioggia |   |
|-------|---|---------|---|
|       |   | 0       | 1 |
| Vento | 0 | 1       | 0 |
|       | 1 | 0       | 0 |



Andiamo a scegliere questi tre parametri in modo da ottenere un percettrone che classifica correttamente

Parametri che possiamo controllare



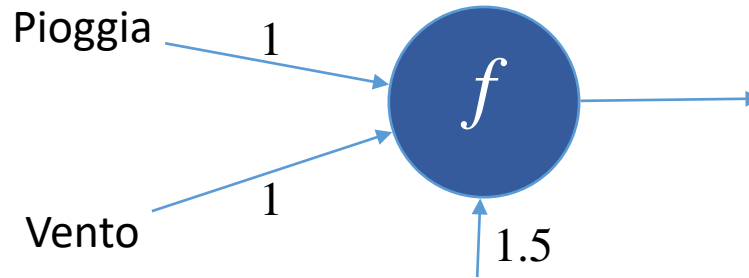
# Percettrone come classificatore binario

Proviamo con

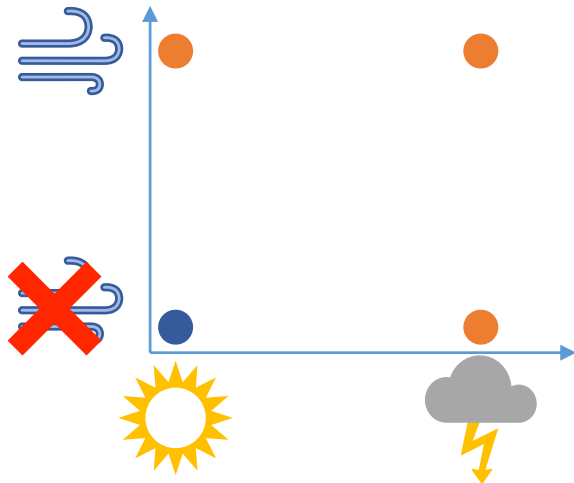
$$w_{\text{pioggia}} = 1$$

$$w_{\text{vento}} = 1$$

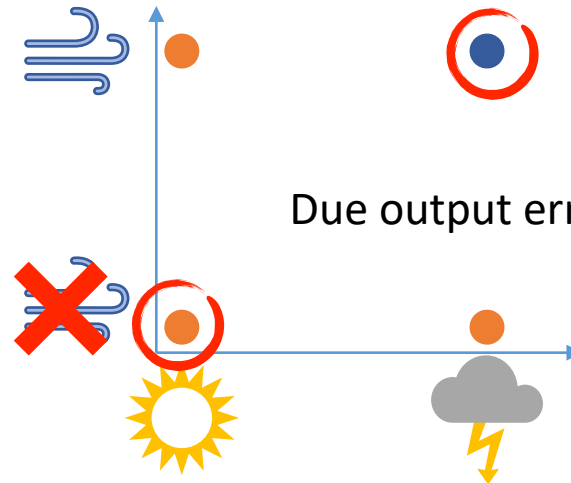
$$\theta = 1.5$$



Dati reali



Risultati ottenuti

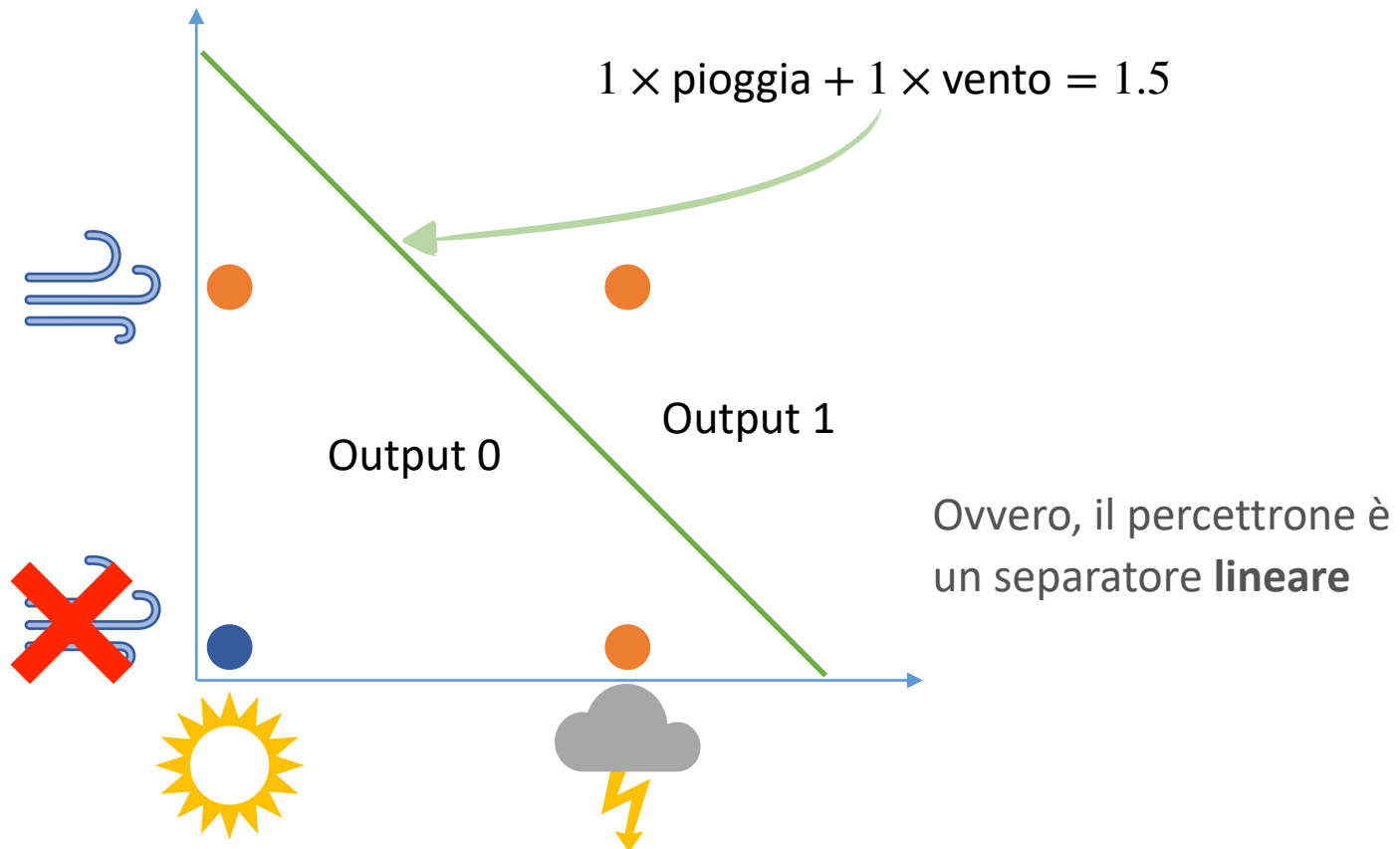


Due output errati

# Percettrone: significato geometrico

Che intuizione abbiamo per impostare i pesi del percettrone?

Un percettrone utilizza una retta\* per separare i risultati 0 da quelli 1

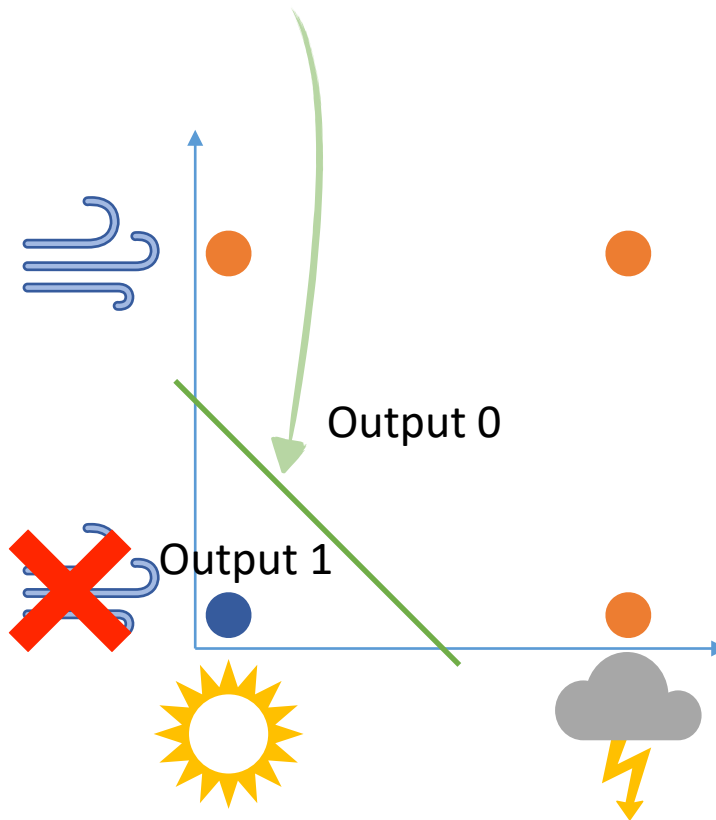


\*in tre dimensioni è un piano, in generale è un iperpiano

# Percettrone: significato geometrico

Questo è quello che vogliamo:

$$w_{\text{pioggia}} \times \text{pioggia} + w_{\text{vento}} \times \text{vento} = \theta$$



Un possibile insieme di parametri potrebbe essere:

$$w_{\text{pioggia}} = -1$$

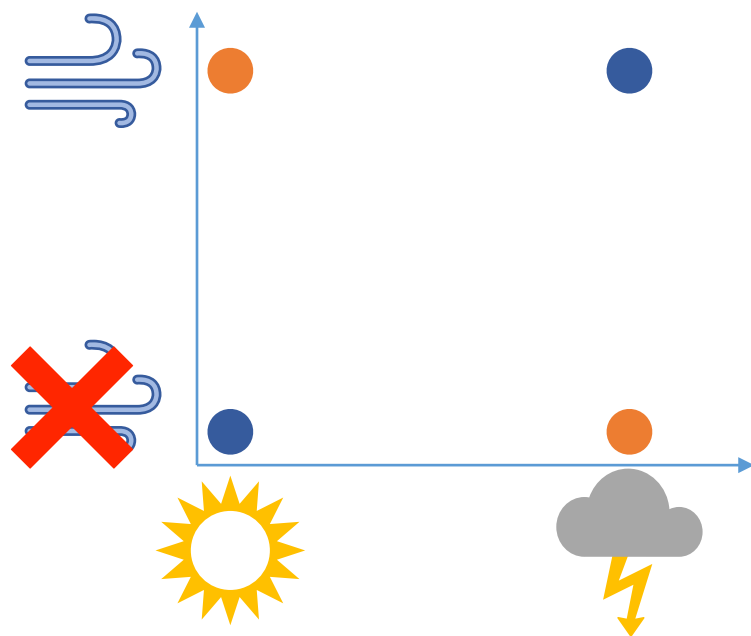
$$w_{\text{vento}} = -1$$

$$\theta = -0.5$$

# Percettrone: limitazioni

Possiamo sempre separare due classi con un percettrone?

**No**, solo classi che sono *linearmente separabili*



**XOR**

Problema **non** linearmente separabile,  
non esiste modo di dividere le due classi  
usando una retta



# Percettrone: apprendimento

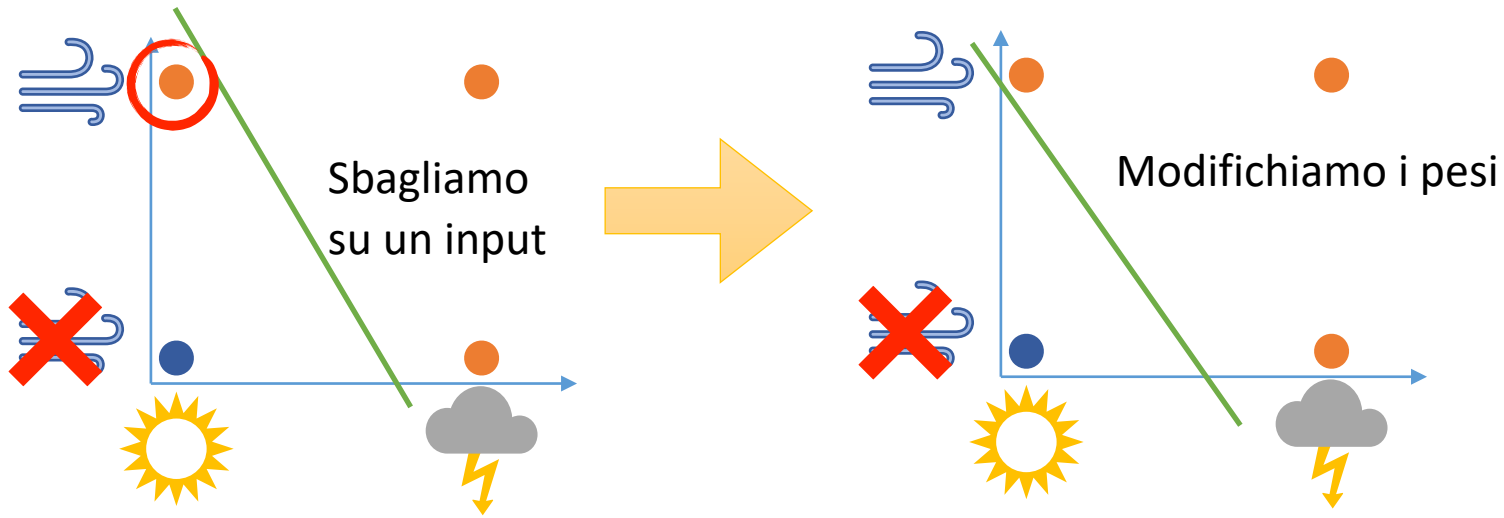
Ma dobbiamo scegliere ogni volta noi i pesi e le soglie?

**No**, esiste un algoritmo di apprendimento

Sommando o sottraendo

## Idea di base:

Se sbagliamo su un input muoviamo i pesi nella direzione che migliora la nostra predizione

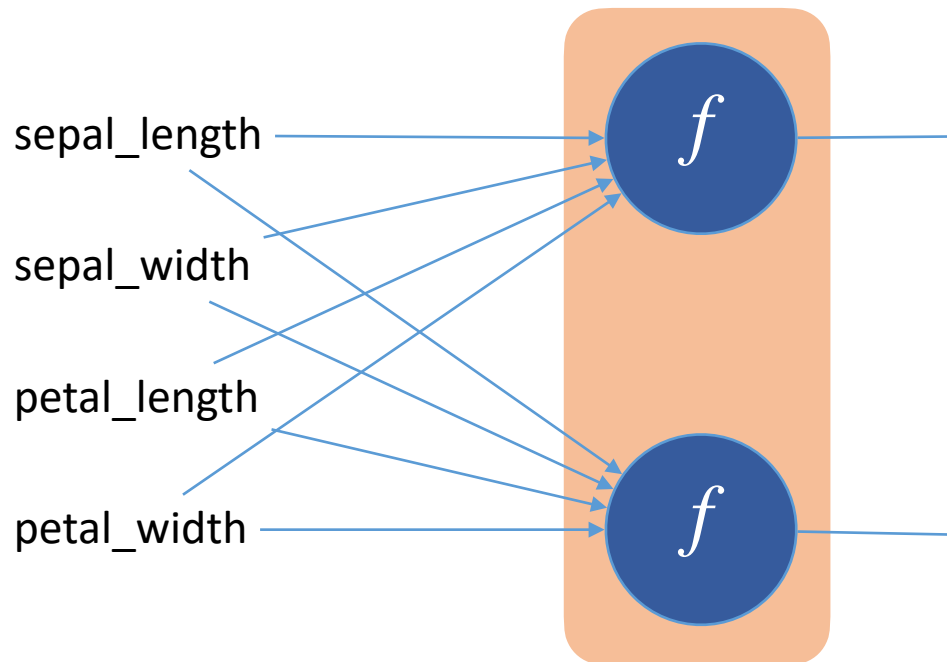




# Percettrone: output multipli

Possiamo usare più percettroni per ottenere più output

La combinazione degli output ci darà la classe di appartenenza  
(è come usare più rette per separare le classi)

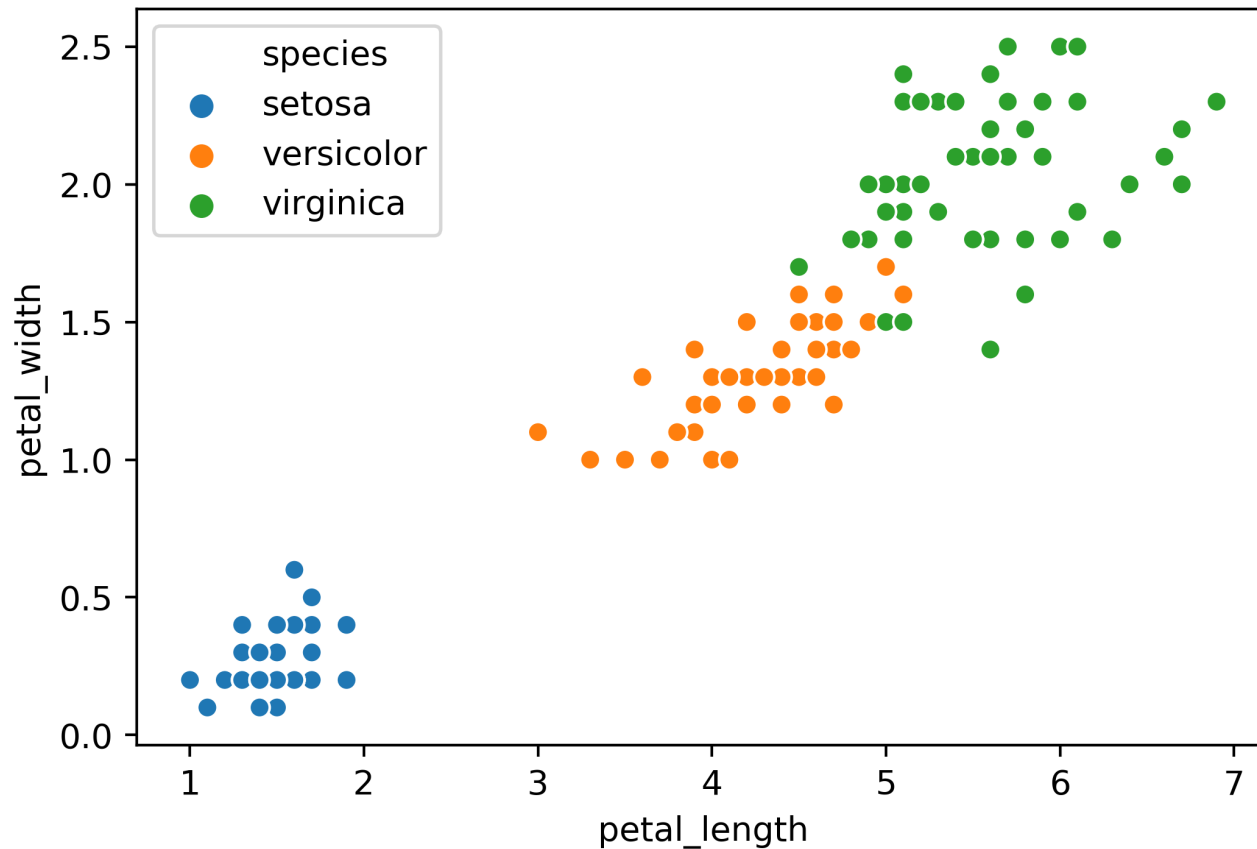


Strato composto da due neuroni



# Iris ed il percettrone

Possiamo ora vedere come una rete neurale composta da un solo strato di percettroni divide le tre classi



# Iris ed il percettrone

La divisione è fatta utilizzando delle rette. Notate come due delle classi non siano linearmente separabili (sebbene l'errore sia basso)

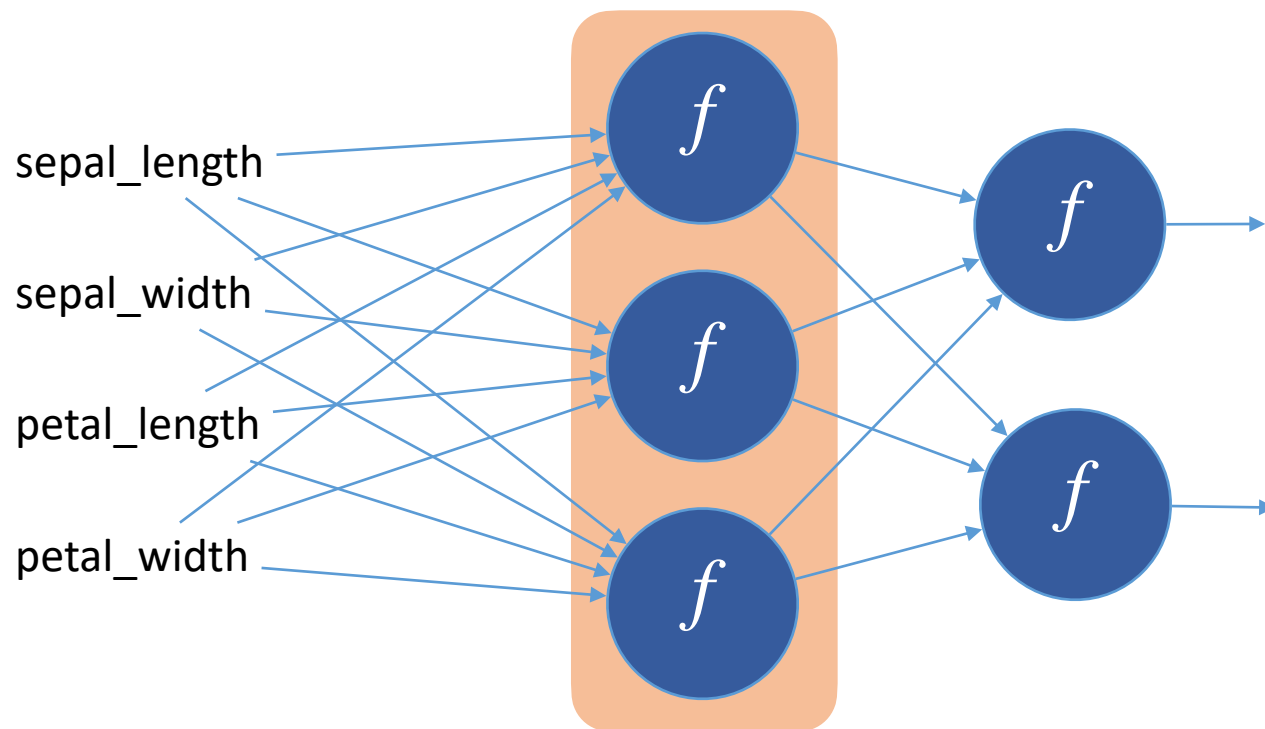


# Oltre le rette: le reti multi-strato

E se vogliamo trattare problemi **non** linearmente separabili?

Aggiungere più neuroni/percettroni sullo stesso livello non risolve il problema...

...quindi si aggiungono più strati di neuroni



Strato **nascosto** composto da tre neuroni



# Teoremi di approssimazione universale

Riusciamo ad avere superfici di separazione abbastanza complesse usando reti con più strati?

Teoremi di approssimazione universale:

- Primi risultati nel 1989
- Successivi miglioramenti nel 1991, 1993 e 1999

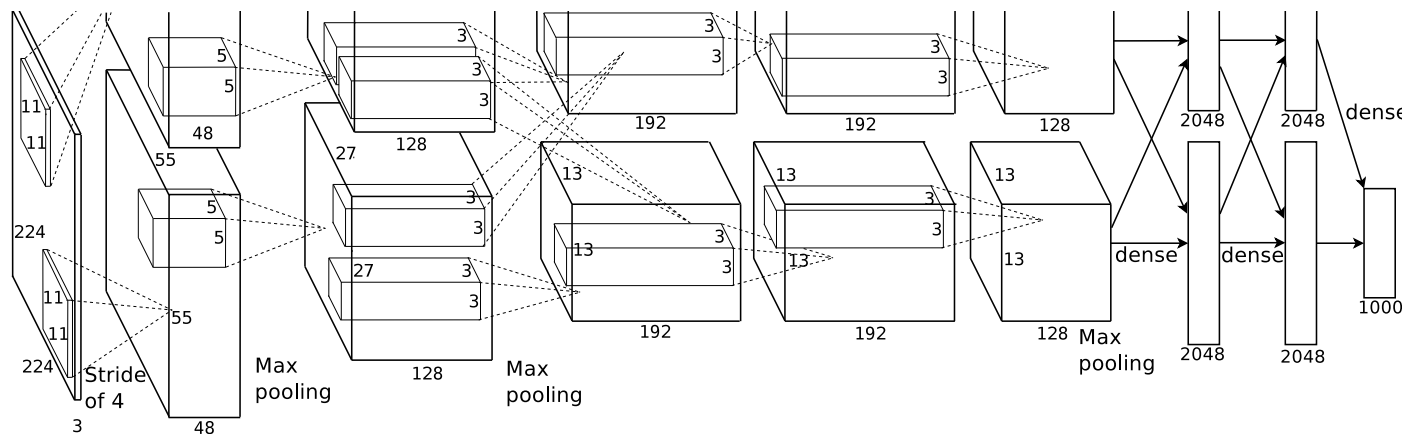
Ma il numero di strati nascosti?  
La dimensione degli strati nascosti?  
La funzione di attivazione?  
etc.

Sono tutti parametri che devono essere scelti in base al problema



# Un esempio di rete moderna

AlexNet, 2012



Dimensione dei livelli nascosti:

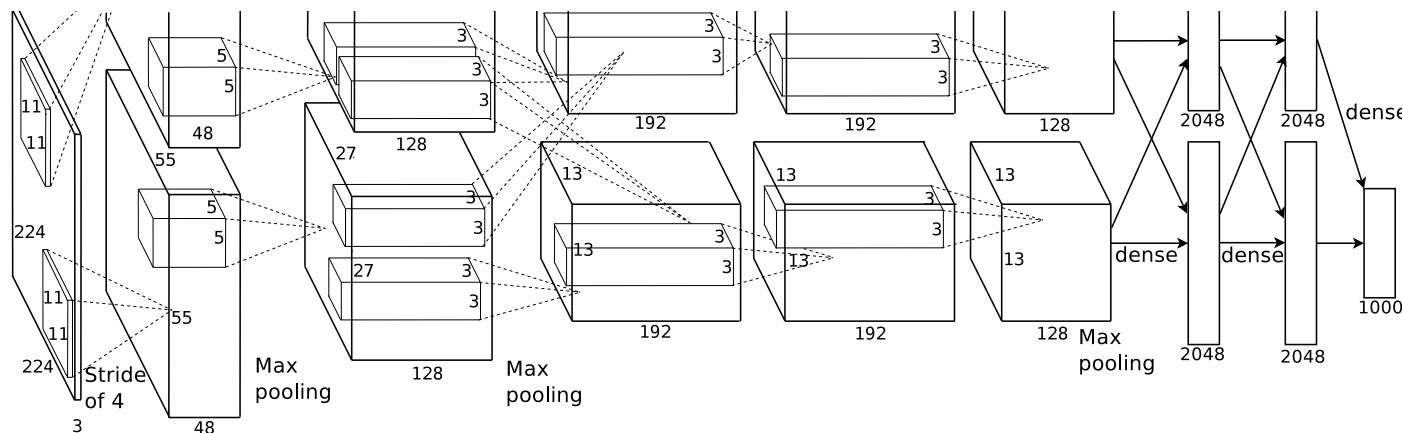
253440 – 186624 – 64896 – 64896 – 43264 – 4096 – 4096 – 1000

Circa 60 milioni di parametri (i pesi della rete)



# Un esempio di rete moderna

AlexNet, 2012



Dimensione dei livelli nascosti:

253440 – 186624 – 64896 – 64896 – 43264 – 4096 – 4096 – 1000

Circa 60 milioni di parametri.

Ma le reti moderne arrivano a miliardi di parametri!

