Misurare l'energia di una particella con una rete neurale

Maurizio Pierini

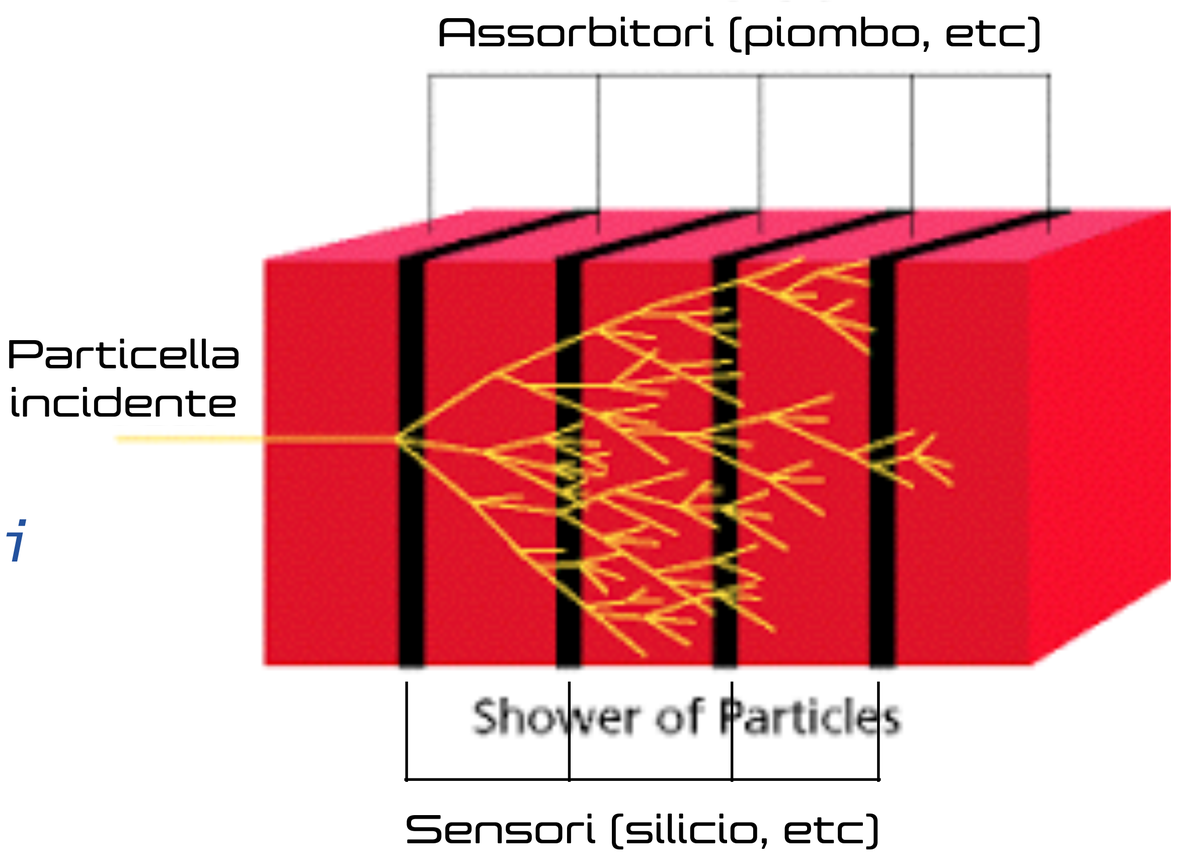
Misura energia

- *Una particella che attraversa un materiale interagisce con gli atomi che incontra, perdendo energia ad ogni urto*

- *Se il materiale e' abbastanza "spesso", la particella rilassata tutta la sua energia e viene assorbita*

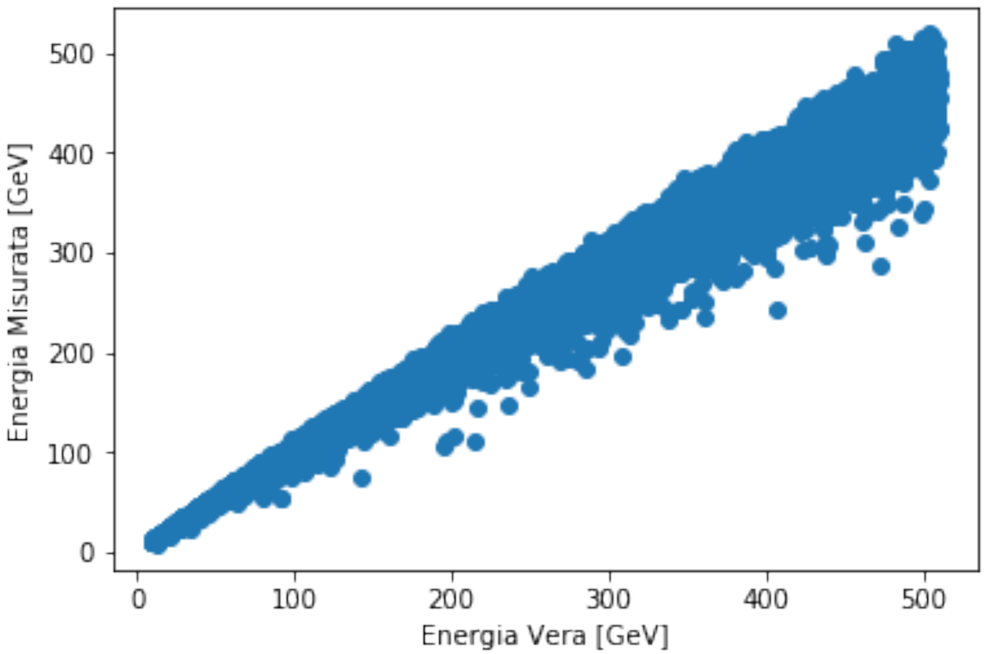
  - *esempio tipico: la luce dei raggi del sole vi scalda (fotoni che cedono energia ai vostri atomi)*

- *In fisica delle particelle, esiste un tipo di rivelatori densi, detti calorimetri, che servono a questo scopo*



Come funziona la misura

- *La particella, nell'interazione, crea particelle secondarie che a loro volta interagiscono con il materiale (sciame)*
- *L'assorbitore cattura la maggior parte della misura, ma non restituisce una misura (materiale passivo)*
- *Il sensore misura una piccola parte dell'energia*
  - *questo perché non esiste un materiale abbastanza denso ed economico da essere usato come assorbitore e sensore*
- *L'energia misurata E_mis viene convertita in E_vera moltiplicandola per una costante di calibrazione misurata a parte*



Nel caso del nostro dataset, la costante e' ~ 50 end e' gia' stat applicata

$$E_{mis} = k \cdot E_{vera}$$

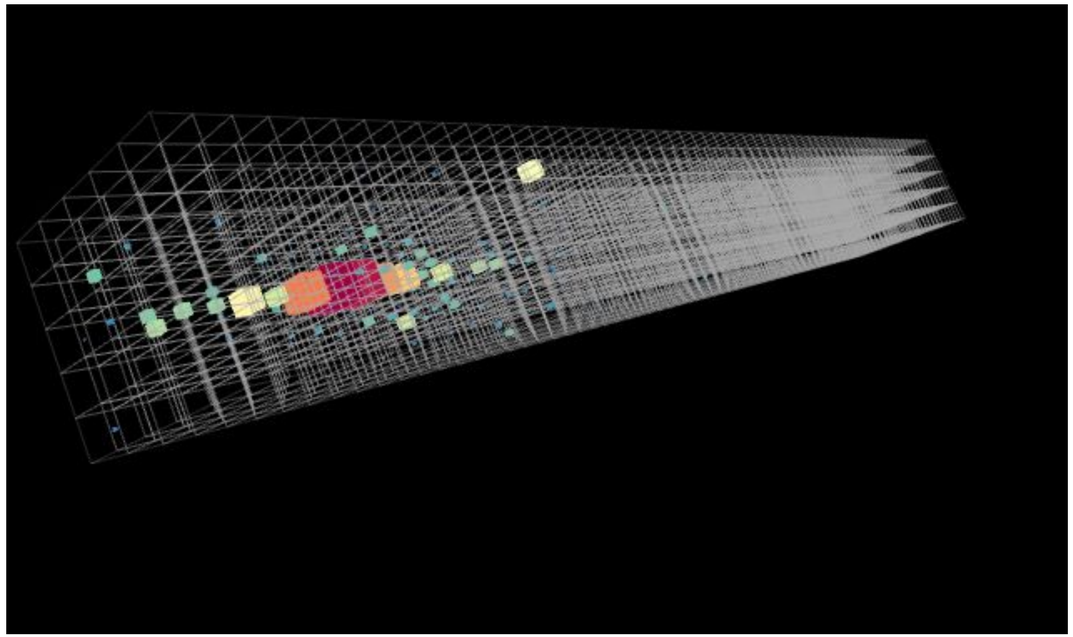
Questo significa che, in media, vediamo 2% dell'energia totale

Si puo' fare meglio?

- *Si, aumentando l'informazione*
  - *Ad esempio, la particella puo' iniziare lo sciame prima o dopo. E quindi k cambia evento per evento, secondo la profondità*
  - *Lo sciame puo' essere piu' o meno largo, secondo il tipo di particella, l'energia, etc*
- *In generale, la forma dello sciame e' legata all'energia vera e determina l'energia misurata (per esempio, quanti sensori sono coinvolti)*
- *Aggiungendo questa informazione, si puo' fare meglio. Questo e' lo scopo dell'esercizio di questa settimana*

Dataset LCD

- *In questo esercizio, useremo un campione di eventi simulati, relativi ad un detector pianificato per un futuro esperimento al CERN*
- *Il detector e' un calorimetro elettromagnetico (ECAL), disegnato per misurare l'energia di elettroni e fotoni*
- *Noi useremo un campione di elettroni con energia tra 10 e 510 GeV (cioè equivalente a 10 volte -> 510 volte la massa del protone)*
- *Per ogni particella, conosciamo l'energia vera (e' una simulazione), l'energia misurata, e una serie di caratteristiche dello sciame*

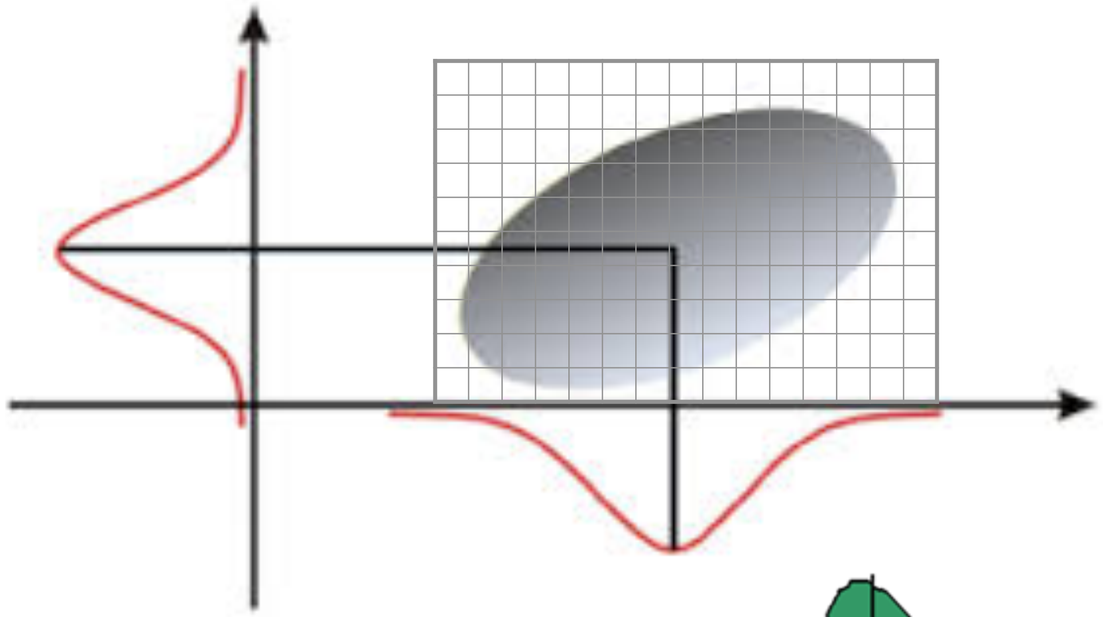


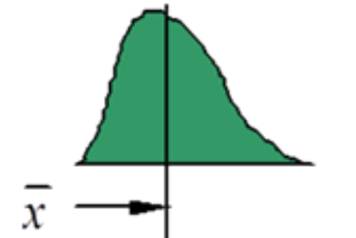
Per ogni particella abbiamo 51x51x25 sensori, che registrano ognuno un valore di energia

Come per la battaglia navale, date le tre coordinate del sensore (i,j,k), possiamo avere l'emergaia di quel sensore E[i,j,k]

Forma dello sciame

- *La forma dello sciame viene quantificata da alcuni numeri*
  - *quanti sensori hanno rivelato un segnale*
  - *energia sul primo strato (layer)/energia totale*
  - *energia sul primo layer / secondo layer*
  - *Momenti 1,2,3 della distribuzione proiettata sull'asse X,Y, e Z*





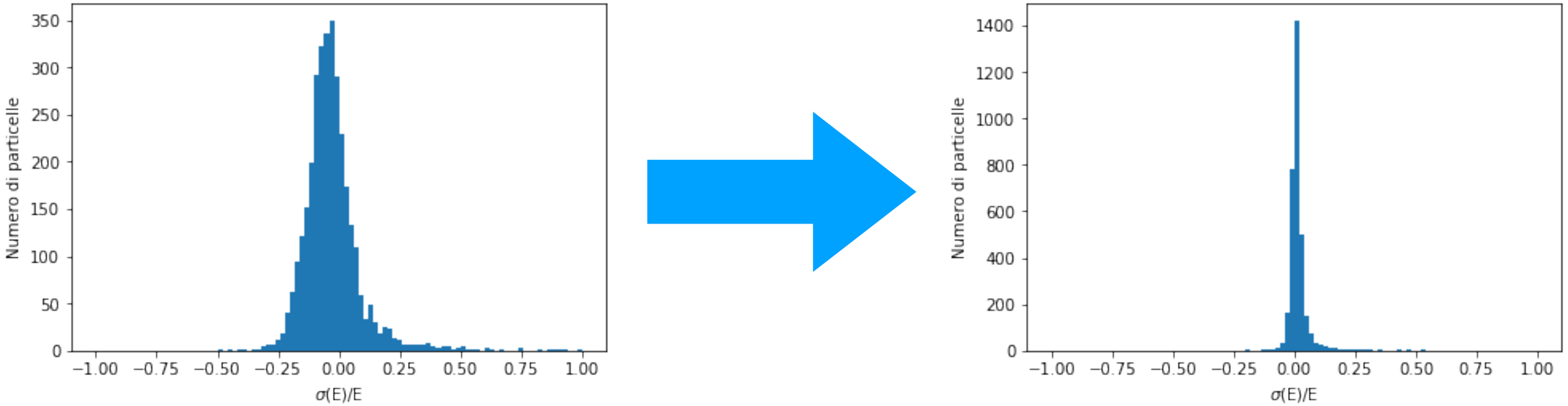
Primo Momento:
misura il valore medio della distribuzione

Primo Momento:
misura la larghezza (la dispersione) della distribuzione

Terzo Momento:
misura la simmetria della distribuzione

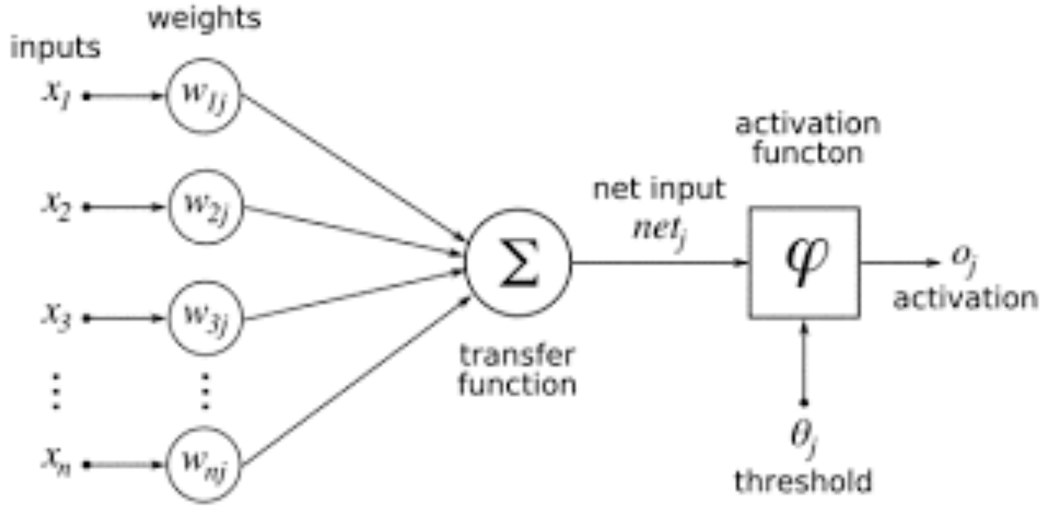
Scopo dell'esercizio

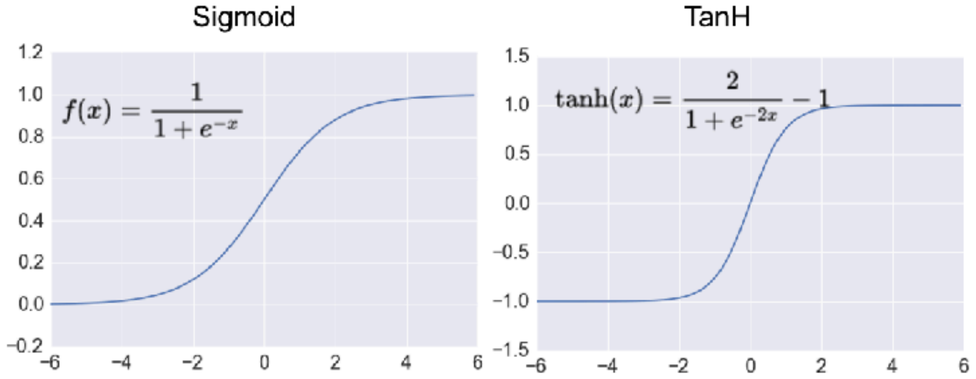
- *Sviluppare una rete neurale che, usando l'energia misurata e la forma dello sciame, fornisca una stima migliore dell'energia della particella*

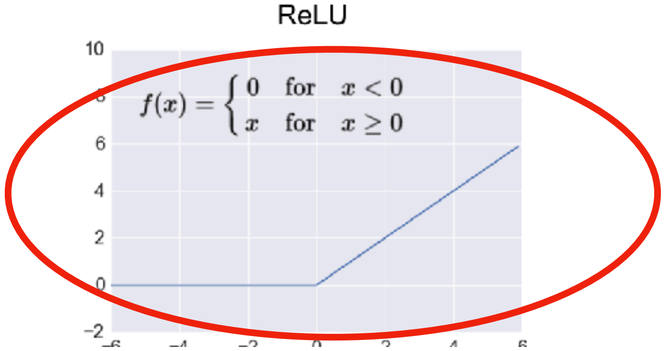


Reti neurali

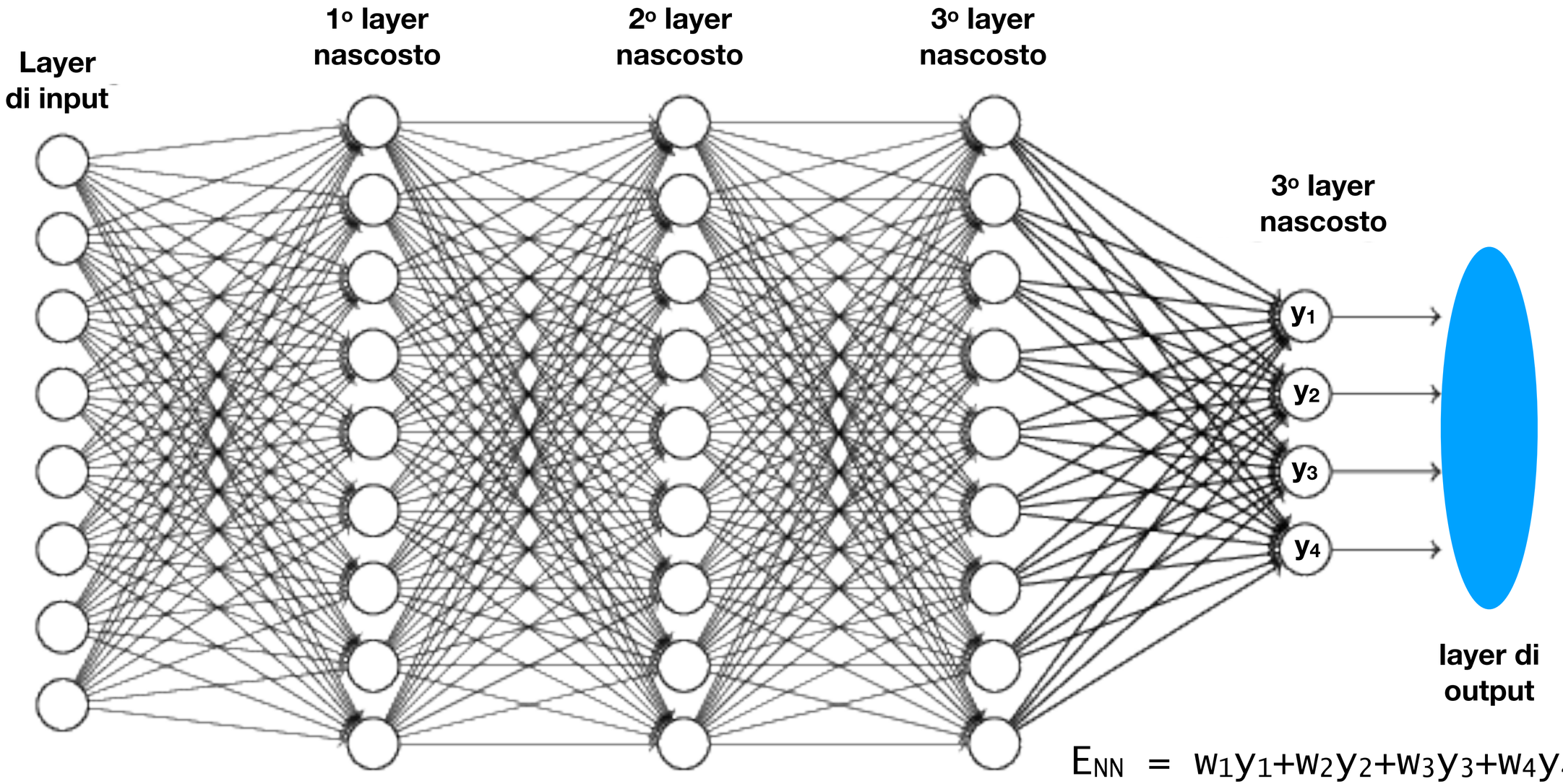
- *Le reti neurali sono classi di funzioni che possono approssimare funzioni piu' complesse*
- *Consistono di strati (layer), composti di neuroni*
- *Ogni neurone riceve degli input, li moltiplica per dei pesi, li somma, e li passa ad una funzione di "attivazione"*
- *Variando i pesi, varia la funzione approssimata dalla rete*
- *I pesi vengono scelti durante il training (allenamento)*







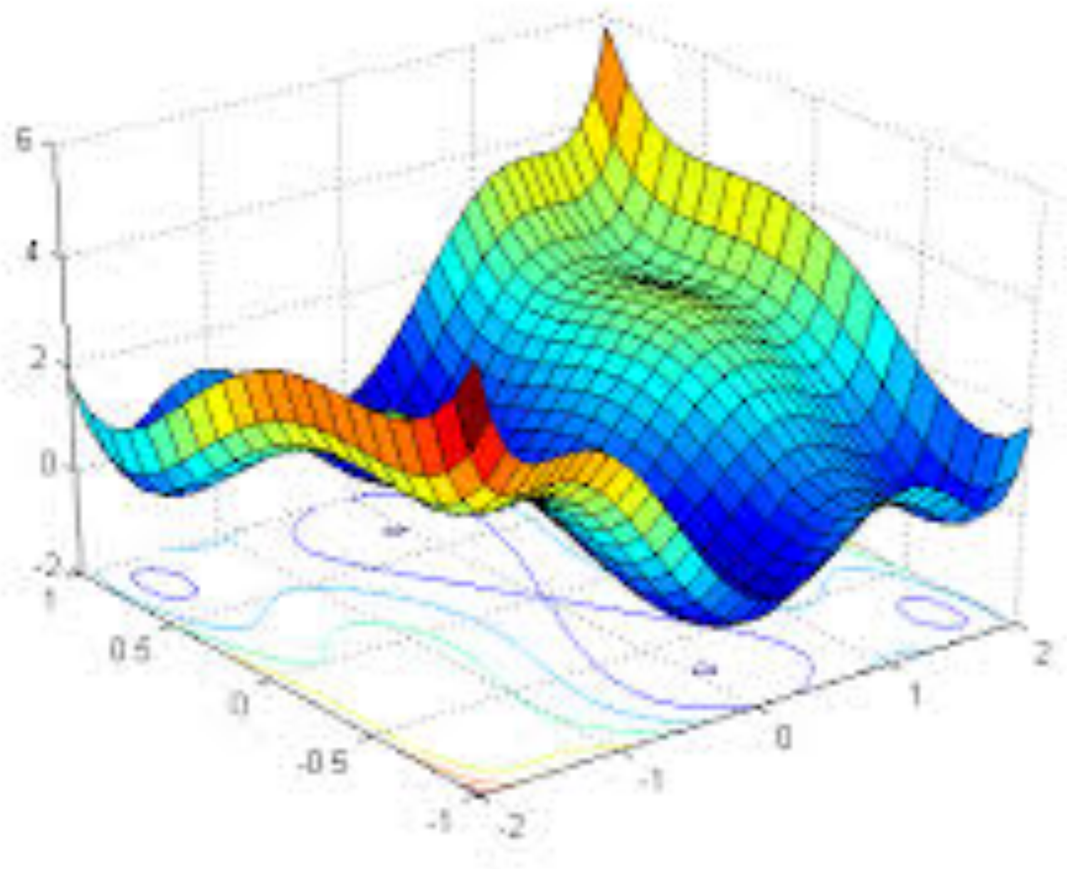
Tipi di layer



$$E_{NN} = w_1y_1+w_2y_2+w_3y_3+w_4y_4$$

Training

- *Durante il training, il valore di uscita della rete viene confrontato con il valore "bersaglio" (nel nostro caso il valore vero dell'energia)*
  - *Sulla base dei due valori, viene calcolata una funzione di perdita (anche detta funzione oggettiva)*
- *Durante il training, il programma che ottimizza la rete cambia i pesi per minimizzare la perdita*



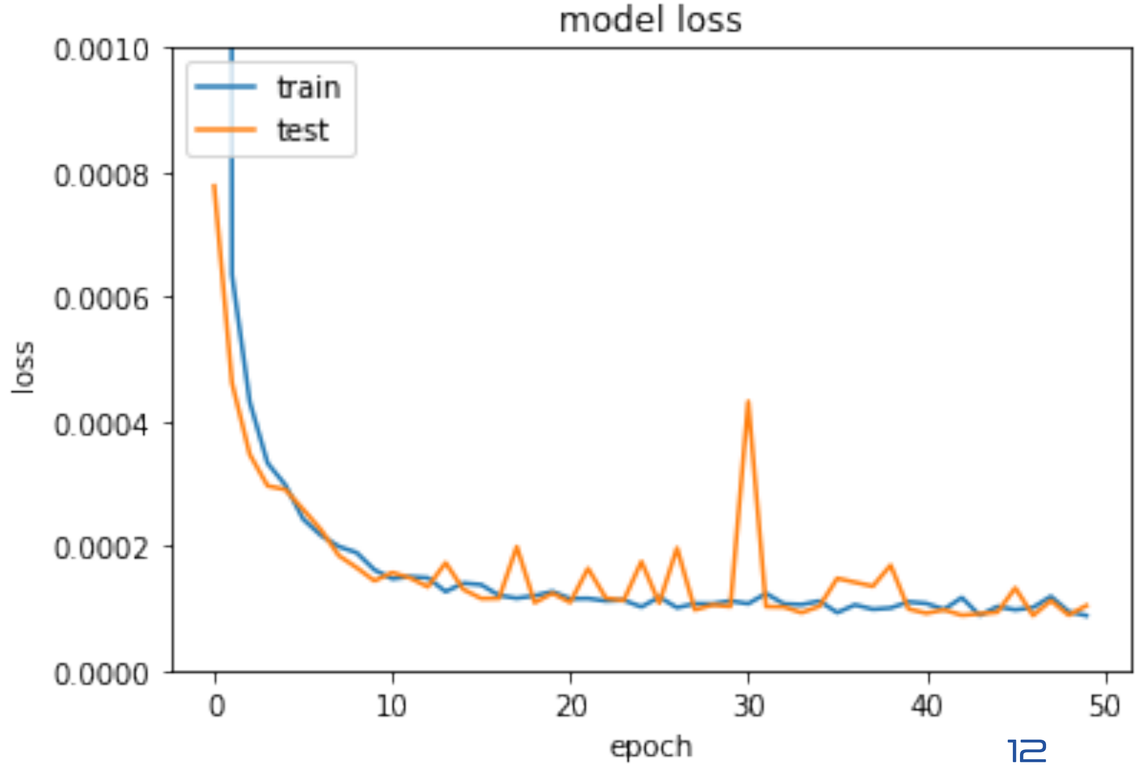
Come procede il training

- *I dati vengono divisi in due*
  - *campione di training (usato per scegliere i pesi)*
  - *campione di validazione (usato per controllare che la configurazione scelta non abbia imparato SOLO I DATI CHE HA VISTO)*
- *Esempio tipico*
  - *voglio trovare macchine rosse nelle immagine*
  - *campione di training solo ferrari*
  - *la rete impara a riconoscere le ferrari rosse, ma non le 500 rosse*

?????

Come procede il training

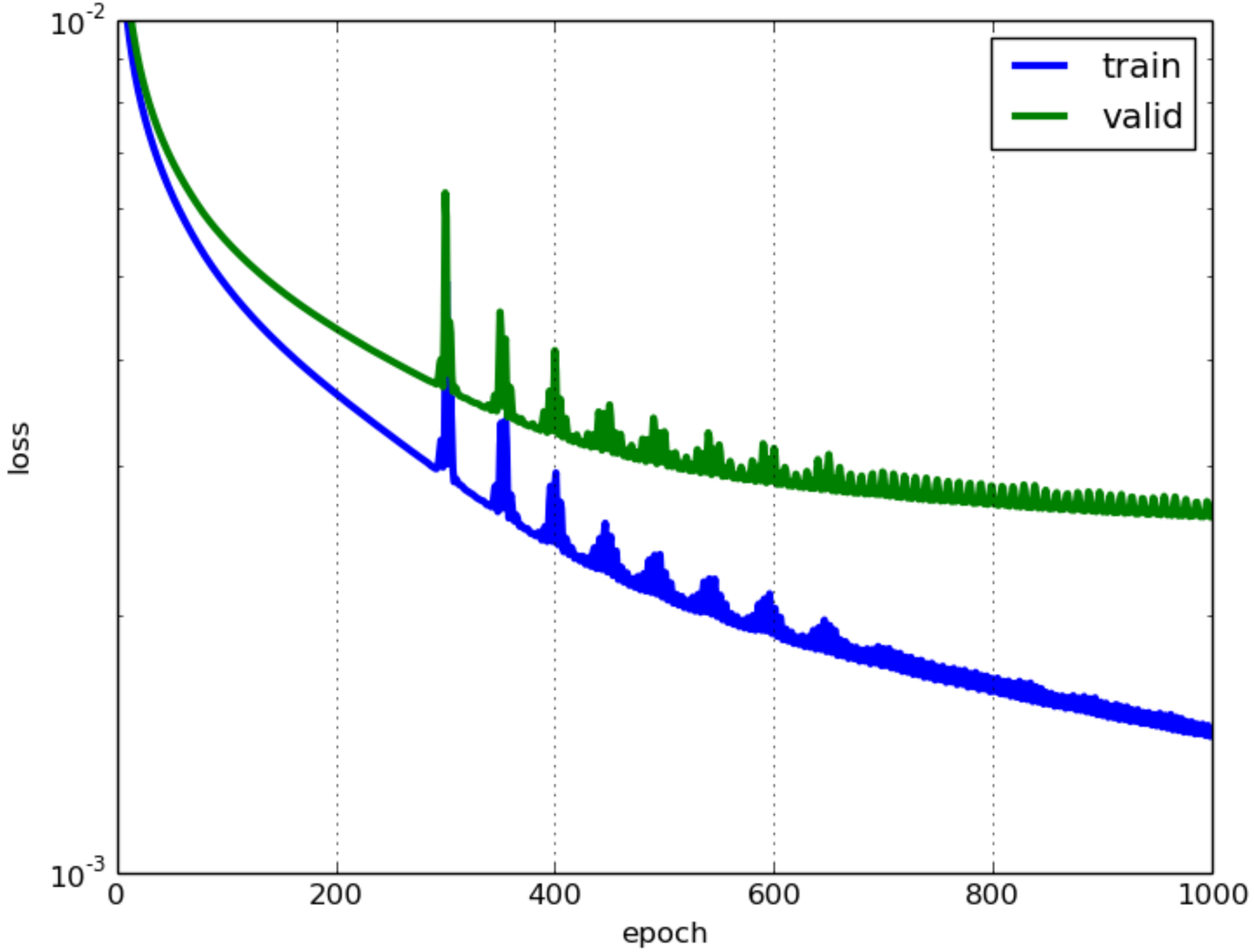
- *Il training e' diviso in epoche*
  - *durante un'epoca, ogni esempio nel dataset di training viene analizzato*
  - *una volta scelti i pesi, il risultato viene applicato al campione di controllo*



????

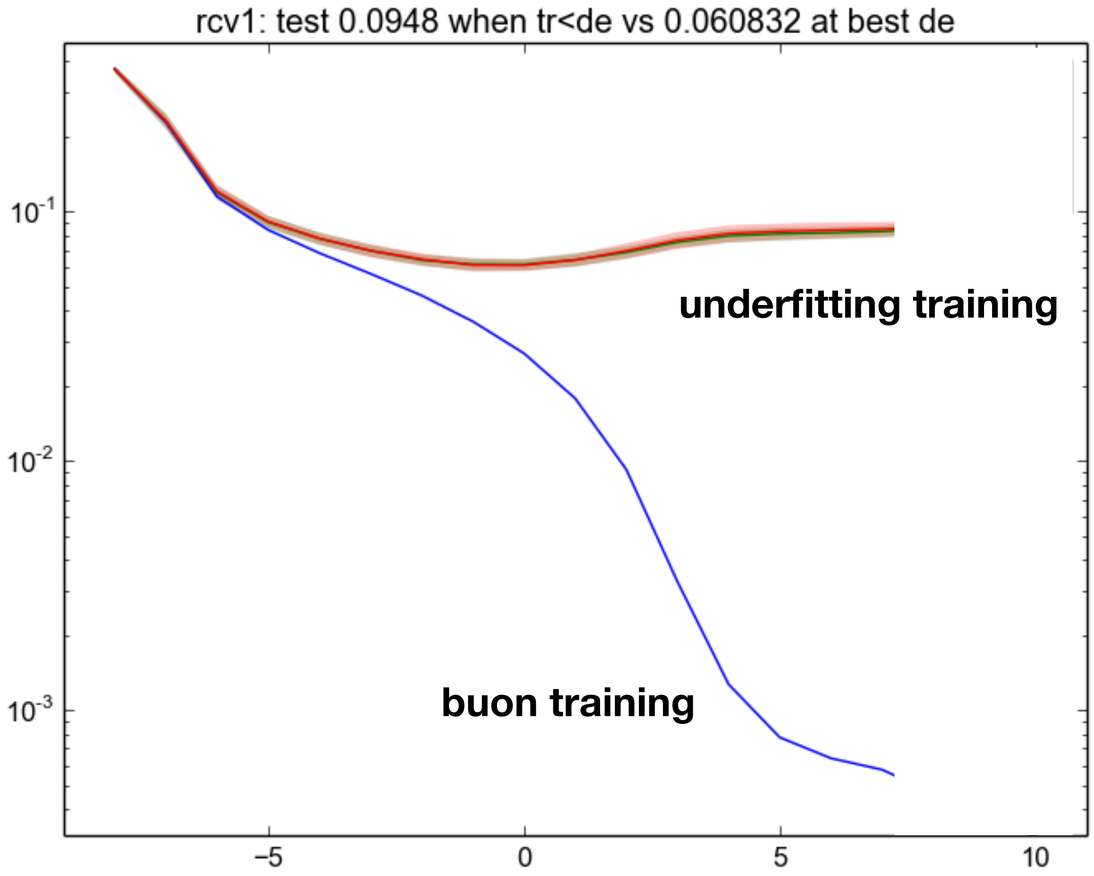
OVERFIT

- *L'Esempio delle Ferrari e' un caso di Overfit: la rete impara dei dettagli specifici del campione di training e il risultato non vale in generale*
  - *In questo caso, la loss function e' minore per il training che per la validazione*
  - *Tipicamente questo succede quando il modello e' troppo complesso e/o il campione di dati non e' abbastanza grande*



Underfit

- *A volte validazione = training ma il risultato non e' buona*
  - *Aumentando le epoche, il risultato non migliora*
  - *in questo caso, la rete non e' abbastanza complessa per imparar la funzione vera*
  - *in questi casi (tipicamente) si aumenta il numero di layer, i neurooni/layer, etc*
  - *tipicamente, aumentare il campione di training aiuta*

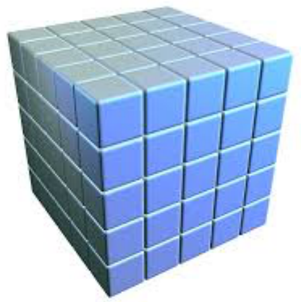


Come facciamo il training

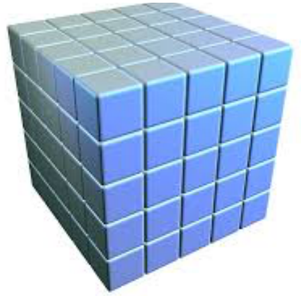
- *Usiamo un algoritmo di ottimizzazione (adam, rmsprop) etc*
- *Usiamo un framework per implementare reti neurali (Google Tensor Flow)*
- *Usiamo una libreria (Keras) che ci permette di costruire facilmente la rete con TF*
- *Usiamo librerie python (numpy, scikit, matplotlib) in un notebook jupyter*


Due parole su numpy

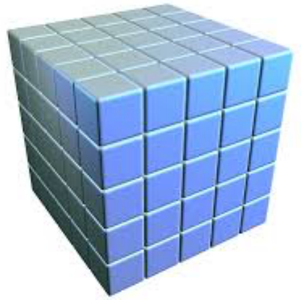
- *numpy e' un codice per gestire array multidimensionali di numeri*
- *Nel nostro caso, l'array puo' avere 4 indici*
  - *primo indice = numero dell'esempio*
  - *altri tre indici sono coordinate in x,y,z*
  - *il contenuto della casella (iX, iY, iZ) e' l'energia della cella in questione*
- *oppure un array 1-dimensionale*
  - *indice = numero dell'esempio*
  - *il contenuto della casella e' il valore dell'array in questione (esempio: energia misurata)*



Esempio 1

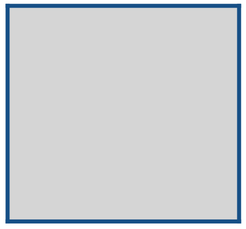


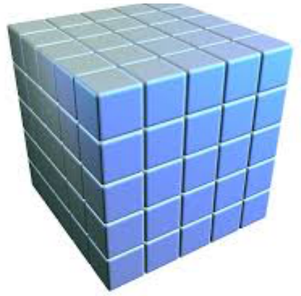
Esempio 2



Esempio 3

...





Esempio N

Due parole su numpy

- *Per acceder all'esempio 4 di un array 3 dimensionale X*
  - *X[3, :, :, :] (perché si inizia a contare da 0)*
- *Per acceder alla cella 2,3,5 dell'esempio 4 di un array 3 dimensionale X*
  - *X[3, 2, 3, 5]*
- *Per acceder al piano iY=2 del quarto esempio*
  - *X[3,:,2,:]*