

## Apprendimento da esempi

Stefano Ferrari

Università degli Studi di Milano  
stefano.ferrari@unimi.it

### Reti Neurali

Stage 2016

## Un semplice problema

Prevedere la traiettoria di un oggetto (vincolato ad un piano)

- ▶ velocità iniziale
- ▶ angolo di uscita
- ▶ posizione relativa del ricevitore dal lanciatore
- ▶ altri parametri da considerare (che non consideriamo)
  - ▶ resistenza del mezzo
  - ▶ accelerazione di gravità non costante
  - ▶ viscosità

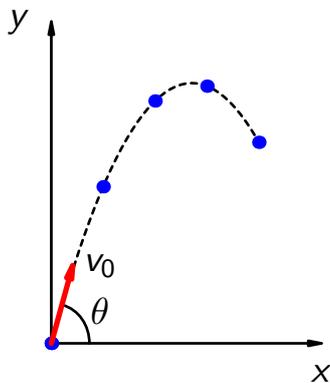
Per esempio: data la posizione relativa e l'angolo di uscita, stabilire la velocità iniziale.

## Approccio “biologico”

- ▶ fai un po' di prove
  - ▶ fatti un'idea di come funziona
- ▶ fai del tuo meglio
- ▶ aggiusta il tiro
  - ▶ in base ai risultati

## Approccio “algoritmico”

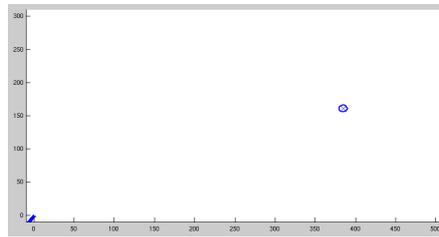
- ▶ trova un buon modello matematico del fenomeno
  - ▶ qualcuno ha già fatto un po' di prove e ha trovato una spiegazione convincente
- ▶ stima i parametri del tuo caso
- ▶ vai!
  - ▶ eventualmente, raffina i parametri



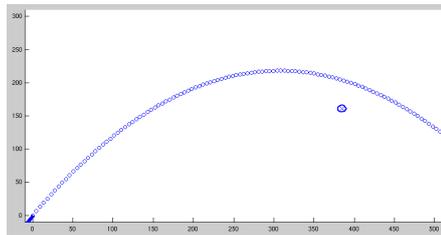
$$y = x \tan \theta - \frac{g x^2}{2 v_0^2 \cos^2 \theta}$$

$$v_0 = \sqrt{\frac{g x^2}{2 \cos^2 \theta (x \tan \theta - y)}}$$

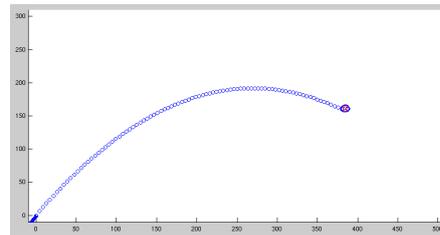
Si gioca!



velocità: ??



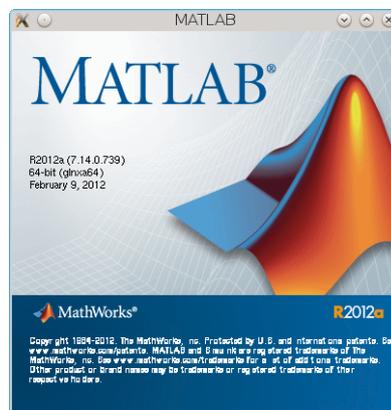
velocità: 80



velocità: 75

## MATLAB

- ▶ MATLAB è l'ambiente di simulazione numerica che useremo per le esercitazioni.



## Hit the target!

1. Scaricare  
`http://homes.di.unimi.it/ferrari/RetiNeurali_2016/reti_neurali_01.zip`
2. Scompattare sul desktop
3. Da Matlab, lanciare *hit\_the\_target*

## Sfruttare i dati

- ▶ Se possiamo vedere abbastanza esempi di problemi risolti e se abbiamo abbastanza memoria, non serve capire il problema per trovare una ragionevole soluzione.
- ▶ Il diavolo si nasconde nei dettagli:
  - ▶ *Abbastanza* e *ragionevole* vanno ben definiti.
  - ▶ In genere dipendono dal tipo di problema.
- ▶ Tuttavia, per problemi semplici funziona:
  - ▶ stima della distanza tra due luoghi in una città conosciuta;
  - ▶ stima del clima in determinati periodi dell'anno.

## $k$ -Nearest Neighbors

- ▶ Il predittore ai primi  $k$  vicini ( $k$ -Nearest Neighbors,  $k$ -NN) è un predittore *pigro* (*lazy*).
  - ▶ I dati di addestramento non vengono elaborati, ma solo memorizzati.
- ▶ E' usato per predire il valore di una funzione,  $f$ , in punti non conosciuti, dato un campionamento della funzione stessa (dati di addestramento),  $\{(x_i, y_i) \mid y_i = f(x_i)\}$ .
- ▶ Per un punto sconosciuto,  $x$ , il valore di  $f(x)$  è stimato dal valore dei suoi  $k$  vicini più prossimi, per un dato valore di  $k$ , usandone una opportuna media.

## $k$ -Nearest Neighbors — Esempio

