

SISTEMI COMPLESSI

In quest'ultima parte del corso affronteremo brevemente lo studio dei fenomeni complessi ed exploreremo il concetto di self-organisation.

Introdurremo pertanto alcuni algoritmi basati su automi cellulari, reti neurali, algoritmi genetici e 'growing networks'

AUTOMI CELLULARI

Uno degli aspetti affascinanti della fisica è la possibilità di ricondurre un fenomeno naturale a poche semplici leggi. L'applicazione di queste leggi permette di riprodurre anche se in maniera approssimata i fenomeni complessi che osserviamo in natura.

La prima classe di modelli che affronteremo va sotto il nome di Automi Cellulari.

Questi ultimi vennero introdotti da von Neumann e Ulam nel 1948 come idealizzazioni matematiche di sistemi dinamici dove lo spazio e il tempo sono discreti e le quantità di interesse possono assumere uno tra un numero discreto e finito di valori. Questi ultimi evolvono tramite leggi locali.

Le caratteristiche principali di un automa cellulare sono:

1. lo spazio è discreto ed è composto da un array regolare di elementi. Ogni elemento può assumere uno tra un insieme finito di valori
2. le regole di aggiornamento dello stato di un elemento dipendono soltanto dai valori assunti dagli elementi 'primi-vicini'

3. il tempo è discreto. lo stato di ogni elemento è aggiornato allo stesso istante per tutti gli elementi del veicolo. L'automa evolve pertanto a tempi discreti.

La motivazione originaria che spinse a sviluppare questi metodi era legata allo studio ~~dei~~ di sistemi cellulari in biologia, dai quali prese il nome.

Consideriamo ora un esempio di automa cellulare ad 1 dimensione.

Supponiamo che ogni elemento della catena possa assumere soltanto due valori $\{0, 1\}$

L'evoluzione di un elemento dipenderà esclusivamente dal valore assunto dai due elementi vicini.

Consideriamo i $2^3 = 8$ casi possibili e fissiamo una regola di evoluzione

	Pattern#							
Temp	7	6	5	4	3	2	1	0
t	111	110	101	100	011	010	001	000
	↓	↓	↓	↓	↓	↓	↓	↓
t+1	101	1	0	1	1	0	1	0

La regola, che prende il nome di regola 90, è fissata dall'equivalente numero - in base dieci - che forma:

$$01011010 = 2^1 + 2^3 + 2^4 + 2^6 = 90_{10}$$

Partiamo da una catena con un elemento a 1, al centro, e studiamo l'evoluzione temporale

t	0	0	0	0	0	0	1	0	0	0	0	0	0
t+1	0	0	0	0	0	1	0	1	0	0	0	0	0
t+2	0	0	0	0	1	0	0	0	1	0	0	0	0
t+3	0	0	0	1	0	1	0	1	0	1	0	0	0
t+4	0	0	1	0	0	0	0	0	0	0	1	0	0
t+5	0	1	0	1	0	0	0	0	0	1	0	1	0
t+6	1	0	0	0	1	0	0	0	1	0	0	0	1

Per gli elementi al bordo considereremo valide condizioni periodiche.

Per l'implementazione dell'algoritmo in un programma, è possibile inglobare le regole in un array

```
int regole[8] = {0, 1, 0, 1, 1, 0, 1, 0};
```

Supponendo che la catena abbia dimensione L , abbiamo bisogno di due array:

- una per lo stato al tempo t

```
int * cell = new int[L];
```

- ed una per calcolare l'evoluzione del sistema al tempo $t+1$

```
int * new_cell = new int[L];
```

Fissata la condizione iniziale per il sistema, durante l'aggiornamento della catena dovremo applicare una condizione di continuità sul bordo:

```
// calcoliamo l'evoluzione della catena
for (int i=0; i<L; i++) {
    int left = cell[(i-1+L)%L];
    int center = cell[i];
    int right = cell[(i+1)%L];
    int pattern = (left<<2) | (center<<1) | right;
    pattern &= 0x7;
    new_cell[i] = rules[pattern];
}
```

Per una catena cellulare unidimensionale
sono possibili $2^8 = 256$ regole.

Altre regole interessanti sono le regole

Numero	111	110	101	100	011	010	001	000
150	1	0	0	1	0	1	1	0
181	1	0	1	1	0	1	0	1
182	1	0	1	1	0	1	1	0
18	0	0	0	1	0	0	1	0

Provate a simulare l'evoluzione di alcuni
sistemi cambiando le regole. Invece di inizializzare
la catena allo stesso modo, è possibile partire
da una configurazione random. Verificare se
i pattern ottenuti sono sensibili alle condizioni iniziali.

Una delle prerogative interessanti degli automi cellulari è la loro capacità di auto-organizzarsi.

Un elemento importante della teoria dei sistemi complessi è che semplici regole possano portare a comportamenti complessi: il comportamento non è casuale ma ha una sua struttura.

MODELLI DEL TRAFFICO

Il lavoro di modellizzazione del traffico è iniziato alla fine degli anni '50 alla General Motors da Robert Herman.

La teoria che descrive il flusso del traffico, e che viene utilizzata ancora oggi, è stata proprio iniziata da Herman e Elliot Montroll in quegli anni.

Un approccio alternativo alla soluzione di equazioni differenziali è basato su automi cellulari.

Consideriamo il modello di Nagel-Schreckenberg.

Modello:

- consideriamo una strada unidirezionale con autoveicoli che entrano da una parte ed escono da quella opposta.
- la posizione e la velocità delle auto possono occupare soltanto posizioni intere
- impostare un valore nominale di velocità v_{max} , la densità di autoveicoli e la probabilità che un'auto rallenti.

L'algoritmo che controlla il movimento di un'antandole ad ogni iterazione è il seguente:

- 1) se $v_j < v_{max}$, aumentare la velocità v_j dell'auto di una unità ($v_j = v_j + 1$)
- 2) calcolare la distanza che la separa dalla macchina successiva, d
se $v_j > d$, ridurre la velocità in modo da prevenire incidenti $v_j = d - 1$
- 3) con probabilità p , ridurre la velocità di un'antandole di una unità
 $v_j = v_j - 1$
- 4) aggiornare la posizione dell'auto in modo che
 $x_j(t+1) = x_j(t) + v_j$

Esempio

Simulare il traffico in una strada lineare introducendo 10 antandoli.

Parametri:

- lunghezza strada: 50 unità
- $v_{max} = 2$ unità/step
- $P_{arresto} = \frac{1}{2}$

Calcolare: flow rate versus car density