Cellular Automata
Sources

Sources:

• Bar-Yam: Dynamics of Complex Systems
• Gros: Complex and Adaptive Dynamical Systems
• http://en.wikipedia.org/wiki/Cellular_automaton
• http://mathworld.wolfram.com/CellularAutomaton.html
• http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
Overview

- Deterministic 1-D cellular automata
- 2-D cellular automata: Game of Life, pattern formation
- sand pile model and self-organized criticality
- branching process
Cellular Automata

- Class of models of dynamical systems
- Appealingly simple yet producing rich set of behaviors. Example of “emergence”.
- Applications in very many diverse scientific fields: crystallization, tumor growth, chemical reactions, traffic flow, ...
- System composed of “cells”, usually in some lattice-like arrangement
- Cells can be in a (finite) set of discrete states
- A cell's state at the next time step depends on the states of its neighbors at the previous time step
1-D Example, binary states

Rule: majority rule: assume state corresponding to majority of 3-neighborhood

How to deal with edges:
1) define extra update rules for edges
2) keep edge cells clamped to specific value
3) periodic boundary conditions: connect outer edges (most elegant)
More 1-D CAs

• Each cell has 2 states, i.e. the three cells in the neighborhood can have 8 possible states

• A specific CA is defined by how it maps the 8 possible states of each cell's neighborhood onto the cell's next state

• How many different CAs are there? (Wolfram code)
Rule 30

Very “complex” pattern emerging. Has been used as pseudo random number generator, but there seem to be better ones.
Conus textile (cloth of gold cone)
Note: this rule has been shown to be Turing complete, i.e. it is as powerful as any computer. To this end, the program to be run and the input data are encoded as the initial configuration. After a finite time, the CA's configuration represents the output of the program.
Rule 184

Traffic flow interpretation: car moves right only if the cell to the right is not occupied
Behavior for different densities

\[ \tau_t = \begin{cases} 
\Theta(\rho, t) & \text{if } \rho < \frac{1}{2}, \\
\frac{1 - \rho}{\rho} \Theta(\rho, t) & \text{otherwise}, 
\end{cases} \]

\[ \Theta(\rho, t) = 1 - \frac{[4\rho(1 - \rho)]^t}{\sqrt{\pi t}}. \]

Wolfram Classification

- Class 1: quickly goes to stable homogeneous state
- Class 2: quickly goes to stable or oscillating patterns, locality
- Class 3: pseudo-random or “chaotic” behavior
- Class 4: development of structures interacting in “complex” and interesting ways
Examples of the 4 classes

class I

class II

class III

class IV

„Computation at the edge of chaos“; „Life at the edge of chaos“
2D Cellular Automata
Conway's Game of Life
• 2-D cellular automaton on a square lattice
• Cells in two states: on/off (alive/dead)
• Cells interact with their 8 neighbors
• “totalistic”: update rule depends only on total number of neighbors in each state, not their spatial pattern
• Deterministic “zero player” game. Everything is completely determined by the initial condition
Game of Life Rules

1) Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.

2) Any live cell with more than three live neighbours dies, as if by overcrowding.

3) Any live cell with two or three live neighbours lives on to the next generation.

4) Any dead cell with exactly three live neighbours becomes a live cell.
• Illustration of Game of Life rules:

- block
- blinker
- glider

• Some common/interesting patterns:

Gosper's glider gun
Play time:
• http://www.ibiblio.org/lifepatterns/

Notes:
• Game of life has very “simple” rules, but is capable of generating very rich (maybe “complex”?) patterns.
• Game of life is in fact capable of universal computation, i.e., it is no more or less powerful than a Turing machine.
2D Pattern Formation CA
Setup:
• Regular grid of “cells”
• each in binary state $s_i \in \{-1, 1\}$
• asynchronous updating: pick single unit
  at random, update, pick next, …
• update rule:

$$s_i(t) = \text{sgn}\left(h + J_1 \sum_{r_{ij} < R_1} s_j(t-1) + J_2 \sum_{R_1 \leq r_{ij} < R_2} s_j(t-1)\right) ,$$

where $\text{sgn}(x) = \begin{cases} 
-1 : x \leq 0 \\
+1 : x > 0 
\end{cases}$

Parameters:
• strength of activation/inactivation: $J_1, J_2$
• range of activation/inactivation: $R_1, R_2$
• “bias”: $h$
% Parameters
length = 200;    % size of layer
fraction = 0.2;  % fraction of units updated each turn
j1 = 1.0;        % strength of activation
j2 = -0.3;       % strength of inactivation
r1 = 3;          % range of activation
r2 = 6;          % range of inactivation
h = 0;           % strength of bias

% Initializations
X = unidrnd(2,length,length)*2-3;
f_size = 2*ceil(r2)+1;
f = j2*ones(f_size);
for(x=1:f_size)
    for(y=1:f_size)
        rr = (x-(f_size+1)/2.0)^2+(y-(f_size+1)/2.0)^2;
        if rr <= r1^2
            f(x,y)=j1;
        elseif rr > r2^2
            f(x,y)=0;
        end
    end
end

figure(1)
imagesc(f); % display filter
colormap(gray);
axis equal;

% Main loop
while(1)
    figure(2)
    imagesc(X); % display state
colormap(gray);
axis equal;
axis off;
pause

    % synchronous update leading to oscillations, so update
    % only a fraction of the units
    New = ((h+conv2(X,f,'same'))>0)*2-1;
    Update = unifrnd(0,1,length,length)<fraction;
    X(Update)=New(Update);
end
Simulations

Varying bias field $h$:

- $j_1=1$, $j_2=-0.3$, $r_1=3$, $r_2=6$, $h=6$
- $j_1=1$, $j_2=-0.3$, $r_1=3$, $r_2=6$, $h=0$
- $j_1=1$, $j_2=-0.3$, $r_1=3$, $r_2=6$, $h=-6$
Simulations

Varying inactivation range \( r_2 \):

- \( j_1=1, j_2=-0.3, r_1=3, r_2=6, h=-6 \)
- \( j_1=1, j_2=-0.3, r_1=3, r_2=12, h=-6 \)
- \( j_1=1, j_2=-0.3, r_1=3, r_2=18, h=-6 \)
Using smaller $J_2$: 

\[ j_1 = 1, \quad j_2 = -0.01, \quad r_1 = 3.5, \quad r_2 = 6, \quad h = 0 \]

\[ j_1 = 1, \quad j_2 = -0.1, \quad r_1 = 3.5, \quad r_2 = 6, \quad h = 0 \]
Discussion of Pattern Formation CA

• nice example of formation of patterns from local interactions
• fairly abstract compared to reaction diffusion DE (Turing instabilities)
• same scheme of local activation/global inhibition
• range of qualitatively different patterns from same dynamics
• Basis for simple models for formation of animal coat patterns such as spots on a leopard or stripes on a zebra
• role of boundary conditions: periodic, non-square lattice
CA Extensions, Generalizations, Applications

- Spatial structure: Higher dimensional lattice, triangular or hexagonal lattice
- Stochasticity: probabilistic component in the update rule
- History dependence: new state of cell may depend on states of neighbors several time steps back
- Introducing conservation laws
- Numerous applications in many fields of science: physics, biology, social sciences, ...