Cellular Automata Fundamentals, Game of Life, and GIS

Yasin Mohammadi

December 11, 2025

Why Cellular Automata (CA)?

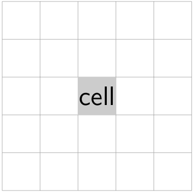
- Many spatial phenomena are:
  - **Discrete**: urban vs. non-urban, land vs. water, forest vs. burned
  - **Local**: what happens at one location depends on nearby locations
  - **Dynamic**: evolve over time (years, months, etc.)
- Cellular automata match this structure:
  - Space $\rightarrow$ grid of cells (like a raster in GIS)
  - Time $\rightarrow$ discrete steps $t = 0, 1, 2, \ldots$
  - Dynamics $\rightarrow$ local rules applied to neighbours
- Natural fit for:
  - Urban growth and land-use change
  - Spread of phenomena (fires, pollution, disease)

What is a Cellular Automaton? (Intuition)

- A cellular automaton is:
  - A grid of cells
  - Each cell has a state (e.g. 0/1, land-use class)
  - At each time step, all cells update their state
- The new state of each cell depends on:
  - Its own current state
  - The states of its neighbours
  - A **common rule** applied everywhere
- Updates are usually **parallel**: all cells update simultaneously.

Basic Components of a CA

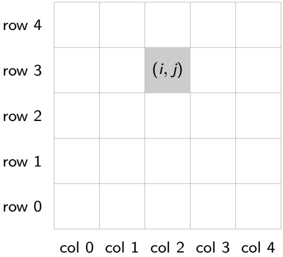
- **Lattice / cells**: discrete positions in space (1D, 2D, 3D)
- **States**: possible values each cell can take
- **Neighbourhood**: which cells influence a given cell
- **Transition rule**: how states change over time
- **Time**: discrete steps $t = 0, 1, 2, \ldots$



2D lattice (grid of cells)

Cell / Lattice: 2D Grid (GIS View)

- In GIS, the lattice is analogous to a **raster**:
  - Each pixel $\leftrightarrow$ one CA cell
  - Each cell has coordinates (row, column) or (x, y)



States

- Each cell has a **state** at time t:

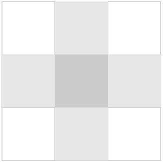
$$s_{i,j}(t) \in S$$

  where S is a finite set of states.
- Examples:
  - **Binary** CA: $S = \{0, 1\}$
    - 0: non-urban, dead, empty
    - 1: urban, alive, occupied
  - **Multi-state** CA: $S = \{0, 1, 2, 3, \dots\}$
    - 0: water, 1: agriculture, 2: forest, 3: urban, ...

2	3	1	0	2
0	1	0	1	0

Neighbourhoods: Von Neumann and Moore

Von Neumann (radius 1)



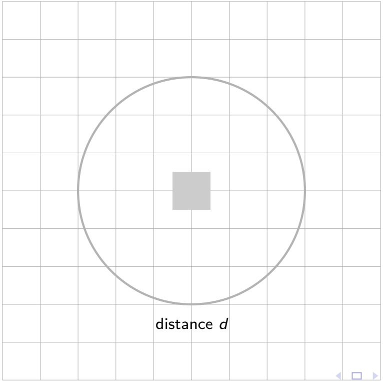
- 4 neighbours
- Share an **edge** with center

Moore (radius 1)

- 8 neighbours
- Share an **edge or corner**

Neighbourhoods: Distance-Based / Circular

- Define neighbours by a **metric distance**:
  - All cells within distance *d* of the center cell.
  - Often Euclidean distance on the grid.
- In GIS:
  - Example: neighbours within 1 km.
  - If raster resolution is 100 m, radius $\approx$ 10 cells.



Transition Rules

- A transition rule gives the new state:

$$s_{i,j}(t+1) = f\big(s_{i,j}(t), \text{neighbourhood}_{i,j}(t)\big)$$

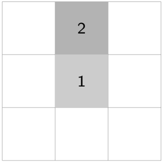
- **Deterministic** rules:
  - Same input neighbourhood $\Rightarrow$ same output.
  - Example (urbanisation):

$$s_{i,j}(t+1) = \begin{cases} 1 & \text{if at least 3 of 8 neighbours are urban} \\ s_{i,j}(t) & \text{otherwise} \end{cases}$$

- **Stochastic** rules:
  - Output may depend on probabilities.
  - Example: if ≥ 3 urban neighbours, cell becomes urban with probability 0.8.

Example Rule: Simple Forest Fire CA

- States:
  - 0: empty
  - 1: tree
  - 2: burning
- Example Moore-neighbourhood rule:
  - If a cell is a **tree** and at least one neighbour is **burning** $\Rightarrow$ it becomes **burning**.
  - If a cell is **burning** $\Rightarrow$ it becomes **empty**.
  - Otherwise, state stays the same.



Tree with burning neighbour

Time and Iterations

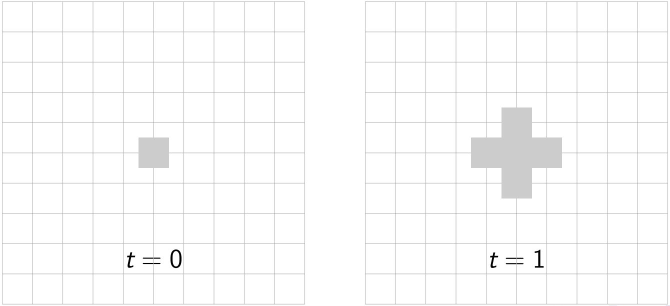
- Time is **discrete**:
$$t = 0, 1, 2, \ldots$$
- At each step $t \to t + 1$:
  - All cells update their state using the rule f.
- **Initial condition** (at $t = 0$):
  - Defines the starting configuration:
    - e.g. initial urban extent in year 2000
- For GIS applications:
  - One time step can represent 1 year, 5 years, etc.
  - After N steps: predicted map at year $t_0 + N$.

Tiny Example: Urban Expansion CA

- Lattice: 10×10 grid.
- States:

$$0 = \text{non-urban}, \quad 1 = \text{urban}$$

- Neighbourhood: Moore (8 neighbours).
- Rule:
  - If a non-urban cell has ≥ 3 urban neighbours $\Rightarrow$ becomes urban.
  - Once urban, it stays urban.



Game of Life: Overview

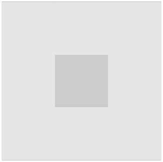
- Classic 2D cellular automaton defined by John Conway.
- Simple local rules, but complex global behaviour.
- Good teaching example for:
  - Lattice, states, neighbourhoods
  - Transition rules and emergent patterns
- We use it as a bridge from abstract CA to spatial thinking.

Game of Life as a Cellular Automaton

- **Lattice**:
  - Infinite 2D grid (in practice: finite $N \times M$ array).
- **States**:

$$S = \{0, 1\}, \quad 0 = \text{dead}, \ 1 = \text{alive}$$

- **Neighbourhood**:
  - Moore neighbourhood (8 neighbours, radius 1).
- **Time**:
  - Discrete steps $t = 0, 1, 2, \ldots$



Game of Life: Local Rule

Let $s_{i,j}(t) \in \{0, 1\}$ be the state of cell (i, j) at time t.

Define $n_{i,j}(t)$ as the number of **alive** neighbours (in the Moore neighbourhood).

The rule:

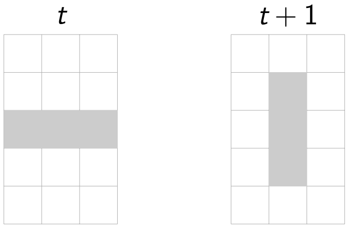
- If a cell is **alive** ($s_{i,j}(t) = 1$):
  - It stays alive if $n_{i,j}(t) = 2$ or 3.
  - Otherwise it dies (overcrowding or loneliness).
- If a cell is **dead** ($s_{i,j}(t) = 0$):
  - It becomes alive if $n_{i,j}(t) = 3$ (reproduction).
  - Otherwise it stays dead.

Formally:

$$s_{i,j}(t+1) = \begin{cases} 1 & \text{if } s_{i,j}(t) = 1 \text{ and } n_{i,j}(t) \in \{2, 3\}, \\ 1 & \text{if } s_{i,j}(t) = 0 \text{ and } n_{i,j}(t) = 3, \\ 0 & \text{otherwise.} \end{cases}$$

Game of Life: Blinker Pattern Example

- **Blinker**: simplest oscillating pattern.
- Period 2:
  - At time t: three cells alive in a horizontal line.
  - At time $t+1$: three cells alive in a vertical line.



Game of Life: From Local Rule to Global Behaviour

- Starting from a simple initial configuration, the pattern evolves over time.
- Typical behaviours:
  - **Still lifes**: stable patterns that do not change.
  - **Oscillators**: repeat after some period.
  - **Spaceships**: moving patterns that travel across the grid.
- Demonstrates how:
  - Simple local rules $\Rightarrow$ complex emergent behaviour.
  - Same idea is used in more realistic CA for urban growth, land-use change, etc.

City Growth CA with DEM: Concept

- **Goal**: simulate urban growth on a 2D grid using elevation (DEM) as a constraint.
- **Lattice**:
  - 20×20 grid (each cell $\rightarrow$ one land parcel).
- **States**:

$$s_{i,j}(t) \in \{0, 1\}, \quad 0 = \text{non-urban}, \ 1 = \text{urban}.$$

- **Additional data**:
  - $h_{i,j}$: elevation (DEM value) at cell (i, j).
- **Neighbourhood**:
  - Moore neighbourhood (8 neighbours, radius 1).

Transition Rules (Using Urban Neighbours and Elevation)

Let $n_{i,j}(t)$ be the number of **urban** neighbours (in the 8-cell Moore neighbourhood).

Rules (for non-urban cells):

- If $n_{i,j}(t) \geq 5$ **and** $h_{i,j} < 800$ m, then the cell becomes urban.
- If $n_{i,j}(t) \in \{3, 4\}$ **and** $h_{i,j} < 500$ m, then the cell becomes urban.

Persistence:

- Once a cell is urban, it remains urban:

$$s_{i,j}(t) = 1 \Rightarrow s_{i,j}(t+1) = 1.$$

Formally:

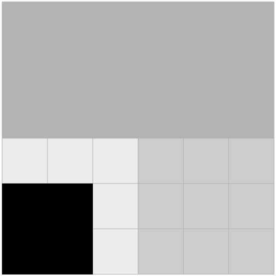
$$s_{i,j}(t+1) = \begin{cases} 1 & \text{if } s_{i,j}(t) = 1, \\ 1 & \text{if } s_{i,j}(t) = 0,\ n_{i,j}(t) \geq 5,\ h_{i,j} < 800, \\ 1 & \text{if } s_{i,j}(t) = 0,\ n_{i,j}(t) \in \{3, 4\},\ h_{i,j} < 500, \\ 0 & \text{otherwise.} \end{cases}$$

Sample DEM and Initial Urban Seed (20×20)

- We use a synthetic DEM on a 20×20 grid:

$$h_{i,j} = 100 + 30\,i + 20\,j \quad \text{(meters)}$$

  where $i, j = 0, 1, \ldots, 19$.
- Elevation increases towards the top-right corner:
  - Low elevation (valley) in bottom-left.
  - High elevation (mountain) in top-right.
- **Initial urban area** at low elevation:
  - A small 2×2 block of urban cells near the bottom-left.



Light gray: low elevation

Dark gray: high elevation

Black: initial urban seed

Bottom-left $\Rightarrow$ valley

Top-right $\Rightarrow$ mountain

Simulation Steps and Interpretation

- At each time step:
  1. Count urban neighbours $n_{i,j}(t)$ for each cell (Moore neighbourhood).
  2. Apply elevation-constrained rules to update $s_{i,j}(t+1)$.
- **Expected behaviour**:
  - Urban area spreads outward where:
    - There is enough existing urbanisation (neighbours).
    - Elevation is low enough (below 500 m or 800 m).
  - High-elevation cells (above 800 m) remain non-urban.
- Demonstrates:
  - How a CA can integrate local neighbourhood effects.
  - How physical constraints (DEM) limit urban expansion.