Cellular Automata

Castillo Reyes, Diego
Escamilla Reséndiz, Aldo
Muñoz Gonzalez, Eduardo
Yañez Martinez, Marthon Leobardo

June 24, 2024

Introduction

Cellular automata (CA) are *discrete, abstract computational systems* that have proved useful both as general models of complexity and as more specific representations of non-linear dynamics in a variety of scientific fields.

Background

Cellular automata (CA) were conceptualized by Stanislaw Ulam and John Von Neumann in the 1940s at the Los Alamos National Laboratory. Von Neumann's extensive work on self-replicating automata was published posthumously in 1966. A CA consists of a one-dimensional array of cells that evolve over discrete time steps.

Cellular Automata Algorithm

Algorithm 1: Basic Cellular Automaton

Input: `gridWidth`: Width of the grid, `gridHeight`: Height of the grid, `states`: Set of possible states for the cells, `neighborhood`: Set of relative positions defining the neighborhood of each cell, `rules`: Set of state transition rules, `maxTimeSteps`: Maximum number of time steps

Output: The final state of the grid

```
Initialize gridHeight × gridWidth, set the initial states on the grid and
  create newGrid as a copy of the grid.;
while i ¡ maxTimeSteps do
    for x in gridWidth do
        for y in gridHeight do
            neighbors = getNeighbors(grid, neighborhood, x, y);
            newGrid[x][y] = applyRules(grid[x][y], neighbors, rules);
    Display the state of newGrid;
    grid = newGrid;
    i++;
```

Parameters Required on a Cellular Automata (I)

Several key parameters determine the structure, behavior, and evolution of a Cellular Automaton (CA):

- **Grid Structure:**
  - *Dimension:* One-dimensional (line of cells) or two-dimensional (grid of cells).
  - *Size:* Number of cells in each dimension (e.g., 100 cells or 100x100 grid).
- **Cell States:**
  - *Number of States:* Possible states each cell can be in (e.g., 0 or 1).
  - *Initial Configuration:* Initial state of cells at time $t = 0$.
- **Neighborhood:**
  - *Radius:* Distance defining the neighborhood of a cell.
  - *Shape:* Common shapes include von Neumann and Moore neighborhoods.

Parameters Required on a Cellular Automata (II)

- **Transition Function:**
  - *Local Rule:* Determines the next state of a cell based on its current state and neighbors.
  - *Update Scheme:* Synchronous (all cells) or asynchronous (one cell at a time).
- **Boundary Conditions:**
  - *Periodic:* Cells wrap around to the opposite edge.
  - *Fixed:* Edge cells have a fixed state.
  - *Reflective:* Edge cells influenced by reflecting states within the grid.
- **Time Steps:**
  - *Discrete Time:* Evolution occurs in discrete time steps.
  - *Duration:* Number of time steps the CA runs.
- **Output and Visualization:**
  - *State Representation:* Visual or numerical representation of cell states.
  - *Data Collection:* Recording states of cells for analysis and visualization.

Versions of Cellular Automata (I)

CA have evolved into various versions based on their dimensionality, state complexity, neighborhood structure, and rule sets.

By Dimensionality:

- *One-Dimensional:* Cells have two neighbors (left and right). Example: Elementary cellular automata by Stephen Wolfram.
- *Two-Dimensional:* Cells have multiple neighbors. Example: Conway's Game of Life.
- *Higher-Dimensional:* Three-dimensional and higher, used for complex simulations.

By Cell States:

- *Binary:* Cells in two states (0 or 1).
- *Multi-State:* Cells in more than two states.
- *Continuous-State:* Cells take on a range of continuous values.

Versions of Cellular Automata (II)

By Neighborhood Type:

- *Von Neumann:* Four orthogonal neighbors (N, S, E, W).
- *Moore:* Eight surrounding cells (orthogonal and diagonal).
- *Extended:* Larger neighborhoods including more distant neighbors.

By Rule Type:

- *Deterministic:* Next state determined by current state and neighbors.
- *Probabilistic:* Next state determined probabilistically.
- *Totalistic:* State depends on the total number of particular states in the neighborhood.

By Boundary Conditions:

- *Periodic:* Grid wraps around, creating a toroidal structure.
- *Fixed:* States at the boundaries are fixed.
- *Reflective:* States at boundaries reflect the states of their neighbors inside the grid.

Special Types of Cellular Automata

Special Types:

- *Elementary CA:* One-dimensional binary CA by Stephen Wolfram, 256 possible rules.
- *Conway's Game of Life:* Two-dimensional binary CA by John Conway.
- *Langton's Ant:* Two-dimensional Turing machine with simple rules and complex behavior.
- *Fuzzy CA:* Combines CA with fuzzy logic for uncertainty in state transitions.
- *Quantum CA:* Extends CA principles to quantum computing, allowing superpositions of states.

These versions enable modeling and simulation from simple theoretical constructs to complex real-world phenomena.

Analogy with Nature (I)

CA come in various forms, inspired by different aspects of natural systems. These versions differ based on their dimensionality, complexity, neighborhood structure, and rules.

Dimensionality:

- *One-Dimensional:* Simple, theoretical studies (e.g., line of ants).
- *Two-Dimensional:* Grid pattern, interactions with neighbors (e.g., moss on a rock).
- *Higher-Dimensional:* Complex models (e.g., 3D growth of crystals).

Cell States:

- *Binary:* Two states, like alive or dead (0 or 1).
- *Multi-State:* Intermediate states, similar to leaf growth stages.
- *Continuous-State:* Spectrum of states, akin to shades of green in a forest.

Analogy with Nature (II)

Neighborhoods:

- *Von Neumann:* Four orthogonal neighbors (N, S, E, W).
- *Moore:* Includes diagonal neighbors (3x3 grid).

Rule Types:

- *Deterministic:* Fixed rules, like seasonal changes.
- *Probabilistic:* Randomness, similar to weather patterns.
- *Totalistic:* Sum of states in neighborhood, like forest density.

Boundary Conditions:

- *Periodic:* Wrap around, toroidal structure.
- *Fixed:* Rigid edge, like shorelines.
- *Reflective:* Mimic natural barriers, reflecting influence.

Special Types of Cellular Automata

Special Types:

- *Elementary CA:* Simple rules leading to complex behavior.
- *Conway's Game of Life:* Birth, survival, and death rules creating lifelike patterns.
- *Langton's Ant:* Simple rules resulting in organized paths.
- *Fuzzy CA:* Combines CA with fuzzy logic for uncertainty.
- *Quantum CA:* Cells in superpositions of states, like quantum particles.

These versions reflect the richness and variety found in natural systems, allowing for modeling and simulation of phenomena across different fields.

Usage Examples

Implementations of Cellular Automata:

7.1. A Computational Tumor Growth Model Experience

- *Overview of Cellular Automata and CNN Integration*
- *Implementation Details*
- *Results and Validation*

7.2. Implementing Fuzzy Cellular Automata in Breast Cancer Image Segmentation

- *Overview of Cellular Automata and Fuzzy Logic Integration*
- *Implementation Details*
- *Advantages of the Approach*
- *Experimental Results*