

Cellular Automata

Diego Castillo Reyes^a, Marthon Leobardo Yañez Martínez^a, Aldo Escamilla Resendiz^a and Muñoz González Eduardo^a

^aResearcher

Dra. Miriam Pescador Rojas

Abstract—Cellular automata are a mathematical and computational model used to simulate dynamic systems. This work presents a review of cellular automata, their history, classification, and applications. Additionally, an example of one-dimensional and two-dimensional cellular automata is shown.

Keywords—Automata, Cellular, Genetic, Algorithms, Simulation

Contents

1	Introduction	1
2	Background	1
2.1	One-Dimensional Cellular Automata	2
2.2	Motivation	3
2.3	Developments	3
	<i>John Conway's Game of Life • Rules of the Game • Initial Configuration and Evolution • Emergent Patterns • Significance and Applications • Simulation and Visualization • Conclusion</i>	
2.4	Stephen Wolfram's Research on Cellular Automata	4
	<i>Elementary Cellular Automata • Rule Classification • Rule 30 and Rule 110 • Applications and Implications • A New Kind of Science • Conclusion</i>	
2.5	Applications	4
3	Cellular Automata Algorithm	5
4	Parameters Required on a Cellular Automata	5
4.1	Grid Structure	5
4.2	Cell States	5
4.3	Neighborhood	5
4.4	Transition Function	5
4.5	Boundary Conditions	5
4.6	Time Steps	5
4.7	Output and Visualization	5
5	Versions of Cellular Automata	5
5.1	By Dimensionality	5
5.2	By Cell States	5
5.3	By Neighborhood Type	6
5.4	By Rule Type	6
5.5	By Boundary Conditions	6
5.6	Special Types of Cellular Automata	6
6	Analogy with the Nature	6
6.1	Dimensionality	6
6.2	Cell States	6
6.3	Neighborhoods	6
6.4	Rule Types	6
6.5	Boundary Conditions	6
6.6	Special Types of Cellular Automata	6
7	Usage Examples	7
7.1	A computational tumor growth model experience based on molecular dynamics point of view using deep cellular automata	7
	<i>Overview of Cellular Automata and CNN Integration • Implementation Details • Results and Validation</i>	
7.2	Implementing Fuzzy Cellular Automata in Breast Cancer Image Segmentation	7
	<i>Overview of Cellular Automata and Fuzzy Logic Integration • Implementation Details • Advantages of the Approach • Experimental Results</i>	

8	Implementation Repositories	7
9	Conclusion	8

1. 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.

Firstly, CA are typically spatially and temporally discrete. They are composed of a finite or denumerable set of homogeneous, simple units called cells. At each time unit, the cells instantiate one of a finite set of states. They evolve in parallel at discrete time steps, following state update functions or dynamical transition rules. The update of a cell's state is obtained by taking into account the states of cells in its local neighborhood, meaning there are no actions at a distance.

Secondly, CA are abstract. They can be specified in purely mathematical terms, and physical structures can implement them.

Thirdly, CA are computational systems. They can compute functions and solve algorithmic problems. Despite functioning differently from traditional Turing machine-like devices, CA with suitable rules can emulate a universal Turing machine¹, and therefore compute anything that is computable according to Turing's thesis.[14]

2. Background

The concept of cellular automata was invented by Stanislaw Ulam and John Von Neumann in the 1940s while they were working at the Los Alamos National Laboratory. The work on cellular automata began in the 1940s, with significant developments occurring throughout that decade. Von Neumann's comprehensive work on self-replicating automata was published posthumously in 1966 in the book "Theory of Self-Reproducing Automata", edited by Arthur W. Burks.

A cellular automaton (CA) is a one-dimensional array (which can be infinitely long in both directions) of cells. Time progresses in discrete steps, and at each step, every cell is in one of a finite number of possible states. The state of each cell changes at each tick of the clock, and this new state is determined entirely by the current state of the cell and its immediate left and right neighbors. This state change is governed by a function called the local rule, which is identical for all cells. The CA operates autonomously without any external input. The collection of all cell states at any given time is referred to as the configuration or global state of the CA, representing its stage of evolution. Starting from an initial configuration at time $t = 0$, the CA evolves deterministically according to the local rule applied to each cell at each clock tick. (see Fig. 1)

When the local rule is applied to each cell, it transforms the entire set of configurations into itself. This transformation is known as the global map or global rule of the CA. This description captures the essence of a CA, which is a widely studied structure.

¹[13, The Stanford Encyclopedia of Philosophy (Winter 2021 Edition), Edward N. Zalta (ed.)]

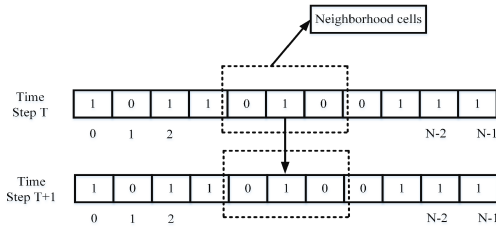
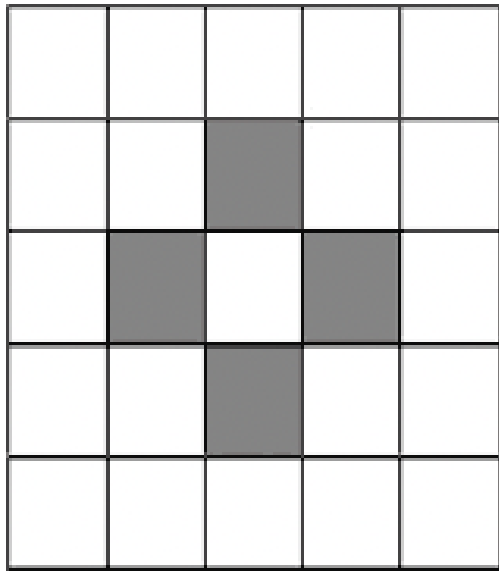


Figure 1. Evolution of a CA at each time step. [16]

The automaton initially described by von Neumann[1] consists of a two-dimensional infinite array of uniform cells, with each cell connected to its four orthogonal neighbors. This structure was originally termed a cellular space, though the term CA is now more commonly used. Von Neumann introduced this concept as a formal model for self-reproducing biological systems. His key ideas can be traced back to his earlier work on modeling biological systems. Von Neumann aimed to apply the rigor of axiomatic and deductive methods to the study of complex natural systems. His concept of a self-reproducing automaton is a remarkable adaptation of the idea behind constructing a universal Turing Machine (TM).



The von Neumann
neighbourhood

Figure 2. Von Neumann's cellular automaton.

2.1. One-Dimensional Cellular Automata

Most of the dynamic features of cellular automata can be observed in the study of the one-dimensional case. In this context, we define the neighborhood of a cell c with a radius r as the r cells to the left of c and the same number of cells to the right of c . Including c itself, this neighborhood comprises $2r + 1$ cells. For the simplest case where $r = 1$ and $k = 2$ (the allowable states), a three-cell neighborhood with two possible states (0 and 1) for each cell can be expressed in $2^3 = 8$ different ways. All eight neighborhood states with $r = 1$ and $k = 2$ are illustrated in Figure 3, and in general, there are k^{2r+1} one-dimensional neighborhood states.[10]

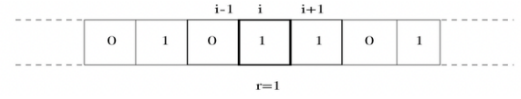


Figure 3. Eight possible neighborhood states with $r = 1$ and $k = 2$.

Let us adopt the notation $c_i(t)$ to denote the state of the i -th cell at time t (Figure 4).

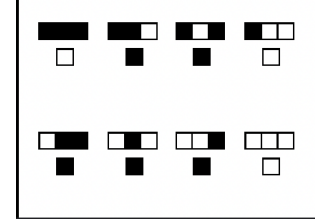


Figure 4. State of the i -th cell at time t .

At the next time step, $t + 1$, the cell state will be $c_i(t + 1)$. Mathematically, we can express the dependence of a cell's state at time step $t + 1$ on the states of its left and right nearest neighbors $c_{i-1}(t)$ and $c_{i+1}(t)$ at time step t by the relation

$$c_i(t + 1) = \phi(c_{i-1}(t), c_i(t), c_{i+1}(t)), \quad (1)$$

where ϕ is the local transition function. For example, consider the simple transition rule

$$c_i(t + 1) = (c_{i-1}(t) + c_i(t) + c_{i+1}(t)) \mod 2, \quad (2)$$

where $\mod 2$ means taking the remainder after dividing the indicated sum by 2, resulting in either 0 or 1. We can put this rule into a transition table format by computing $c_{i-1}(t) + c_i(t) + c_{i+1}(t) \mod 2$ for the eight possible input values:

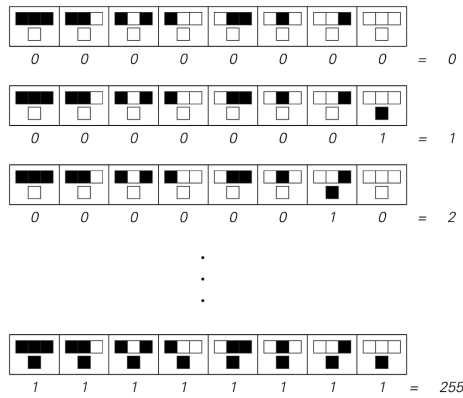
$c_{i-1}(t)$	$c_i(t)$	$c_{i+1}(t)$	$c_i(t + 1)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Table 1. Transition table for the rule $c_i(t + 1) = (c_{i-1}(t) + c_i(t) + c_{i+1}(t)) \mod 2$.

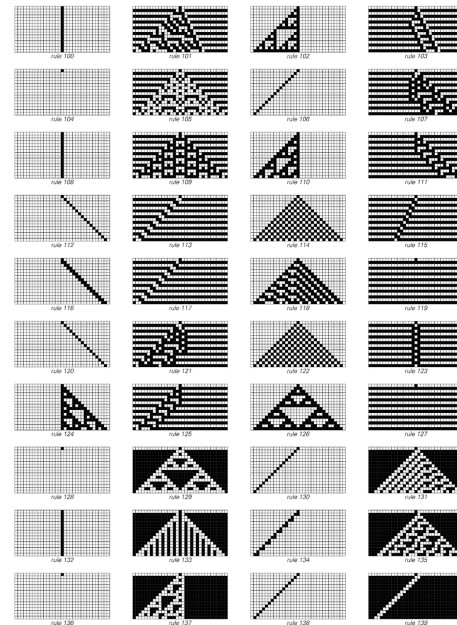
A very convenient way to illustrate the allowable rules for one-dimensional cellular automata with $r = 1$ and $k = 2$ is to indicate the state (color) of the middle cell at the next time step, given the state (color) of itself and its two nearest neighbors (Figure 4).

Here, the middle cell at time t (in the top row) has its state altered according to the states of its two nearest neighbors and its own state to yield the cell's new state at time $t + 1$ (bottom row). This also represents the rule in Equation 2. Observe that in four of the eight cases, a black cell appears. One consequence of this is that if we took a disordered array of a large number of cell sites, then the average fraction (or density) of black cell sites that evolve after one iteration will be 0.5.

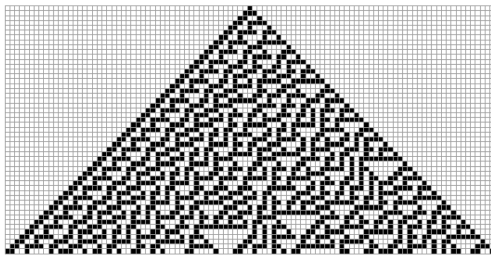
The cellular automaton defined by this rule has the graphical representation depicted in Figure 5, starting with a single black cell with each subsequent generation of the automaton appearing on the next line down.



(a) The 256 rules



(b) Generations for some rules

Figure 6. The 256 rules and the generations for some of them.**Figure 5.** Graphical representation of the evolution of the rule starting with a single black cell. Each new line downward represents the evolution of the automaton at the next time step.

So, how many possible rules are there for $r = 1$ and $k = 2$? Since there are eight possible neighborhood states of three cells and each of these results in two possible state outcomes for the middle cell, there are $2^8 = 256$ possible transition rules by which this can be achieved as seen in Figure 6. Wolfram described these as elementary cellular automata.

2.2. Motivation

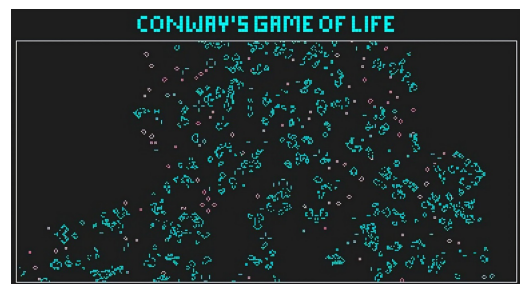
Cellular automata, a concept derived from computer science, are increasingly employed as models in ecological research. This concept spans a variety of topics, including automata theory and artificial intelligence. Cellular automata, as representations of this concept, create miniature worlds where each cell houses an automaton. The behavior of a cellular automaton can be remarkably complex, often resembling lifelike processes. This complexity has inspired a new field of study known as artificial life or, more cautiously, complex systems. Conferences in this field attract computer and mathematical scientists, as well as a growing number of participants from geography, ecology, geology, and other disciplines, along with a popular audience. Currently, the field of ecology has accumulated sufficient experience with cellular automaton models to move beyond initial simple models and harness the full potential of this tool for comprehensive simulation. However, validation and interpretation continue to be significant challenges in utilizing these models effectively.[11]

The primary motivation behind cellular automata was to understand and model complex systems using simple, local rules. This idea was rooted in the study of biological processes and the desire to create self-replicating machines.

2.3. Developments

John Conway's work (1970s): Conway popularized cellular automata with his "Game of Life", a bidimensional binary cellular automaton. This game demonstrated how simple rules could lead to complex emergent behavior, sparking widespread interest and research in cellular automata.

Fig. 7 An example of Conway's Game of Life.

**Figure 7.** Conway's Game of Life.

Stephen Wolfram's work (1980s): Wolfram conducted extensive research on cellular automata, classifying them into four types based on their behavior and demonstrating their potential as models of natural processes and as computational systems.

2.3.1. John Conway's Game of Life

John Conway's Game of Life is a well-known example of a two-dimensional cellular automaton. Devised by the British mathematician John Horton Conway in 1970, it gained significant interest due to its simplicity and the complex behaviors it can exhibit.

2.3.2. Rules of the Game

The Game of Life is played on a rectangular grid of cells, where each cell can be in one of two states: alive (1) or dead (0). The state of each

cell evolves through discrete time steps according to a set of simple rules based on the states of its eight neighbors (the Moore neighborhood). A dead cell with exactly three live neighbors becomes a live cell, akin to reproduction. A live cell with two or three live neighbors remains alive, reflecting a stable environment. Conversely, a live cell with fewer than two live neighbors dies, simulating underpopulation, and a live cell with more than three live neighbors dies, simulating overpopulation.

2.3.3. Initial Configuration and Evolution

The evolution of the grid begins from an initial configuration of live and dead cells. The subsequent generations are determined by simultaneously applying the aforementioned rules to every cell in the grid. This process is repeated for as many time steps as desired, creating a dynamic and evolving pattern.

2.3.4. Emergent Patterns

One of the most fascinating aspects of the Game of Life is the emergence of complex patterns from simple initial conditions. Still lifes, such as the Block and Beehive, are patterns that do not change from one generation to the next. Oscillators, like the Blinker and Toad, return to their initial state after a fixed number of generations. Spaceships, including the Glider and Lightweight Spaceship (LWSS), translate themselves across the grid, demonstrating movement.

2.3.5. Significance and Applications

The Game of Life has been extensively studied not only as a mathematical curiosity but also for its implications in various fields. In computational theory, the Game of Life is Turing complete, meaning it can simulate any computation given enough resources, demonstrating that simple rules can lead to universal computation[6]. In the study of complex systems, it serves as a model for emergent behavior, illustrating how simple interactions can lead to intricate and unpredictable patterns [2]. Researchers in artificial life use the Game of Life to study processes such as self-replication, evolution, and the emergence of life-like properties from non-living systems [5].

2.3.6. Simulation and Visualization

The Game of Life is often simulated using computers to explore its vast array of patterns and behaviors. Many software tools and libraries have been developed for this purpose. Golly is a popular open-source application that allows users to create, simulate, and analyze patterns in the Game of Life and other cellular automata[18]. Various online platforms also enable users to interactively explore the Game of Life directly in their web browsers, making it accessible for educational and research purposes.

2.3.7. Conclusion

John Conway's Game of Life exemplifies the power and elegance of cellular automata. Its simple rules lead to complex and fascinating behaviors, making it a rich subject for study in mathematics, computer science, and beyond. The Game of Life continues to inspire and challenge researchers, illustrating the profound connection between simplicity and complexity in dynamic systems.

2.4. Stephen Wolfram's Research on Cellular Automata

Stephen Wolfram is a prominent figure in the study of cellular automata, and his research has significantly advanced the understanding of these systems. Wolfram's work explores the complexity that arises from simple rules and how such systems can model natural phenomena.[3]

2.4.1. Elementary Cellular Automata

Wolfram's research primarily focuses on one-dimensional cellular automata, known as elementary cellular automata. These consist of a line of cells, each of which can be in one of two states: 0 (dead) or 1 (alive). The state of each cell in the next time step depends on its current state and the state of its two immediate neighbors. Despite

the simplicity of these rules, elementary cellular automata can exhibit a wide range of behaviors, from predictable and repetitive to highly complex and chaotic.

2.4.2. Rule Classification

Wolfram classified the behavior of cellular automata into four classes based on their long-term behavior:[4]

Class 1: These automata evolve into a homogeneous state. Any random initial configuration eventually leads to a uniform state. This behavior is similar to systems that settle into a stable equilibrium.

Class 2: These automata evolve into a set of stable or oscillating structures. Patterns emerge that are stable or repeat periodically. This behavior is akin to the formation of stable structures or cycles in nature.

Class 3: These automata produce random or chaotic patterns. Small changes in the initial configuration can lead to significant differences in the evolution of the automaton, reflecting chaotic systems in nature.

Class 4: These automata generate complex structures that can interact with each other in complicated ways. This class is particularly interesting because it can support universal computation, meaning these automata can perform any computation given the right initial conditions and rules.

2.4.3. Rule 30 and Rule 110

Among the elementary cellular automata, Rule 30 and Rule 110 are particularly notable. Rule 30 generates a pattern that appears random and is used in random number generation.[7][9] It exhibits chaotic behavior from simple initial conditions. Rule 110, on the other hand, has been proven to be Turing complete, meaning it can simulate any computation. This demonstrates that even simple rules can lead to computational universality.

2.4.4. Applications and Implications

Wolfram's research has broad implications across various fields. In physics, cellular automata provide models for understanding complex systems and phenomena such as fluid dynamics and pattern formation. In computer science, they offer insights into computation theory and the nature of complexity. Wolfram's work also intersects with biology, where cellular automata can model the growth patterns of organisms and the dynamics of ecosystems.

2.4.5. A New Kind of Science

Wolfram's seminal book, *A New Kind of Science*, encapsulates his research and presents a comprehensive view of how simple computational rules can lead to complex behaviors. The book argues that the universe itself might operate according to principles similar to those of cellular automata, suggesting a computational foundation for all natural processes.

2.4.6. Conclusion

Stephen Wolfram's research on cellular automata reveals the profound complexity that can arise from simple rules. His classification of automata behaviors, the study of specific rules like Rule 30 and Rule 110, and the broader implications for understanding natural phenomena highlight the importance of cellular automata in both theoretical and applied sciences. Wolfram's work continues to inspire researchers and offers a unique perspective on the computational nature of the universe.

2.5. Applications

Cellular automata have been used in various fields, including:

- **Computer Science:** Parallel computation, cryptography, and image processing.
- **Physics:** Modeling physical systems, such as fluid dynamics and crystal growth.

- **Biology:** Simulating biological processes, such as population dynamics and pattern formation.

Fig. 8 Application of cellular automata on Computer Science.

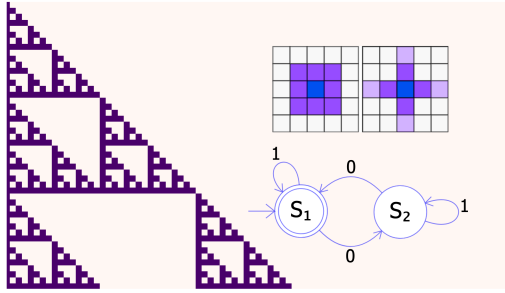


Figure 8. Example of cellular automata.

3. 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

```

1 Initialize gridHeight × gridWidth, set the initial states on
  the grid and create newGrid as a copy of the grid.;
2 while i < maxTimeSteps do
3   for x in gridWidth do
4     for y in gridHeight do
5       neighbors = getNeighbors(grid, neighborhood,
6                               x, y);
7       newGrid[x][y] = applyRules(grid[x][y],
8                                     neighbors, rules);
9   Display the state of newGrid grid = newGrid;
10  i++;

```

4. Parameters Required on a Cellular Automata

CA, several key parameters need to be defined. These parameters determine the structure, behavior, and evolution of the CA. The primary parameters include:

4.1. Grid Structure

- **Dimension:** The number of dimensions of the CA grid. Common configurations are one-dimensional (a line of cells) and two-dimensional (a grid of cells).
- **Size:** The number of cells in each dimension. For example, a one-dimensional CA might have 100 cells, while a two-dimensional CA might have a 100x100 grid.

4.2. Cell States

- **Number of States:** The finite set of possible states each cell can be in. For example, in a binary CA, each cell can be in one of two states: 0 or 1.
- **Initial Configuration:** The initial state of all cells at time $t = 0$. This can be a specific pattern, a random configuration, or any other predefined setup.

4.3. Neighborhood

- **Radius:** The distance defining the neighborhood of a cell. In a one-dimensional CA, a radius of 1 means the neighborhood includes the cell itself and its immediate left and right neighbors.
- **Shape:** The shape of the neighborhood. Common shapes include the von Neumann neighborhood (cells directly adjacent in cardinal directions) and the Moore neighborhood (cells adjacent in both cardinal and diagonal directions).

4.4. Transition Function

- **Local Rule:** The function that determines the next state of a cell based on its current state and the states of its neighbors. This function can be deterministic or probabilistic.
- **Update Scheme:** Whether the CA updates all cells simultaneously (synchronously) or one at a time (asynchronously).

4.5. Boundary Conditions

- **Periodic Boundaries:** Cells at the edge of the grid wrap around to the opposite edge, creating a toroidal structure.
- **Fixed Boundaries:** Cells at the edge of the grid have a fixed state or do not change state based on neighbors outside the grid.
- **Reflective Boundaries:** The states of edge cells are influenced by reflecting the states of neighbors within the grid.

4.6. Time Steps

- **Discrete Time:** CA evolution occurs in discrete time steps, typically denoted by $t = 0, 1, 2, \dots$
- **Duration:** The number of time steps the CA runs for, which can be predefined or based on a stopping condition.

4.7. Output and Visualization

- **State Representation:** How the states of the cells are represented visually or numerically (e.g., using different colors or symbols for different states).
- **Data Collection:** Methods for recording the states of cells at each time step for analysis and visualization.

By defining these parameters, one can fully specify a Cellular Automaton model and simulate its evolution. The flexibility in choosing these parameters allows CAs to be adapted to a wide range of applications, from simple theoretical models to complex simulations in various fields of science and engineering.

5. Versions of Cellular Automata

CA have evolved into various versions and classifications based on their dimensionality, state complexity, neighborhood structure, and rule sets. Here are some of the key versions of Cellular Automata:

5.1. By Dimensionality

- **One-Dimensional Cellular Automata:** Each cell has two neighbors (left and right). These are the simplest form and are often used for theoretical studies. An example is the elementary cellular automata defined by Stephen Wolfram.
- **Two-Dimensional Cellular Automata:** Each cell has multiple neighbors depending on the neighborhood structure. Conway's Game of Life is a well-known example.
- **Higher-Dimensional Cellular Automata:** These include three-dimensional and higher-dimensional grids, which are used for more complex simulations but are less common due to increased computational complexity.

5.2. By Cell States

- **Binary Cellular Automata:** Each cell can be in one of two states, typically denoted as 0 and 1. This is the simplest form and is widely studied.

- **Multi-State Cellular Automata:** Cells can exist in more than two states. These automata are used to model more complex systems where each cell can represent multiple possible conditions.
- **Continuous-State Cellular Automata:** Cells can take on a range of continuous values, rather than discrete states, allowing for the modeling of more nuanced phenomena.

5.3. By Neighborhood Type

- **Von Neumann Neighborhood:** Consists of the four orthogonal neighbors (north, south, east, and west) of a cell in a two-dimensional grid.
- **Moore Neighborhood:** Includes the eight surrounding cells (both orthogonal and diagonal neighbors) in a two-dimensional grid.
- **Extended Neighborhoods:** Larger neighborhoods that include more distant neighbors, such as the Margolus neighborhood.

5.4. By Rule Type

- **Deterministic Cellular Automata:** The next state of each cell is determined entirely by the current state and the states of its neighbors, following a fixed rule.
- **Probabilistic Cellular Automata:** The next state of each cell is determined probabilistically, allowing for the modeling of stochastic processes.
- **Totalistic Cellular Automata:** The state of a cell depends only on the total number (or sum) of particular states in the neighborhood, not on the exact configuration of those states.

5.5. By Boundary Conditions

- **Periodic Boundary Conditions:** The grid is treated as if it wraps around on itself, creating a toroidal structure.
- **Fixed Boundary Conditions:** The states of the cells at the boundaries are fixed and do not change.
- **Reflective Boundary Conditions:** The states of the cells at the boundaries reflect the states of their neighbors inside the grid.

5.6. Special Types of Cellular Automata

- **Elementary Cellular Automata:** A class of one-dimensional binary cellular automata defined by Stephen Wolfram, consisting of 256 possible rules.
- **Conway's Game of Life:** A two-dimensional binary cellular automaton devised by John Conway, famous for its simple rules leading to complex behavior.
- **Langton's Ant:** A two-dimensional Turing machine with very simple rules but complex emergent behavior.
- **Fuzzy Cellular Automata:** Combines cellular automata with fuzzy logic to handle uncertainty and imprecision in state transitions.
- **Quantum Cellular Automata:** An extension of cellular automata principles to quantum computing, allowing cells to be in superpositions of states.

These different versions of cellular automata enable the modeling and simulation of a wide range of systems, from simple theoretical constructs to complex real-world phenomena.

6. Analogy with the Nature

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

6.1. Dimensionality

Just as nature operates at different scales, from the linear growth of a vine to the branching of trees and the intricate structures of coral reefs, CAs can also be one-dimensional, two-dimensional, or even higher-dimensional. One-dimensional CAs, like a line of ants following

each other, are the simplest and often used for theoretical studies [7]. Two-dimensional CAs resemble the spread of moss on a rock surface, where each cell interacts with its immediate neighbors in a grid pattern [2]. Higher-dimensional CAs, though more complex, can model phenomena such as the three-dimensional growth of crystals or the propagation of signals in a neural network.

6.2. Cell States

In the same way that an organism can be alive or dead, hibernating or active, cells in a CA can exist in various states. Binary CAs, where each cell can be either 0 or 1, are analogous to on/off states, such as a plant being either alive or dead. Multi-state CAs extend this analogy by allowing for intermediate states, similar to a tree showing different stages of leaf growth, colors, and shedding. Continuous-state CAs go even further, representing states on a spectrum, akin to the varying shades of green in a forest canopy.

6.3. Neighborhoods

The interactions in nature are often local. A tree might only affect and be affected by the trees around it. Similarly, in a CA, the concept of a neighborhood defines which cells influence a given cell's state. In a von Neumann neighborhood, a cell's state is influenced by its four orthogonal neighbors, like a central plant interacting with the four closest plants around it. In a Moore neighborhood, the interaction extends to include diagonal neighbors as well, resembling a plant affected by all surrounding plants in a 3x3 grid.

6.4. Rule Types

The rules that govern a CA's evolution can be deterministic or probabilistic, mirroring the predictability or randomness found in nature. Deterministic CAs follow fixed rules, much like the consistent seasonal changes that influence plant growth. Probabilistic CAs, on the other hand, incorporate randomness, similar to how weather patterns can unpredictably affect an ecosystem. Totalistic CAs simplify these interactions by considering the sum of states in the neighborhood, analogous to how the overall density of a forest might influence the growth of an individual tree rather than the specific arrangement of nearby trees.

6.5. Boundary Conditions

Nature often operates within boundaries, such as the edge of a forest or the banks of a river. In CAs, boundary conditions define the limits of the grid. Periodic boundaries wrap around, creating a seamless, toroidal structure, similar to the cyclical nature of seasons or life cycles. Fixed boundaries impose a rigid edge, like the shorelines of an island. Reflective boundaries mimic natural barriers that reflect influence, akin to a mountain reflecting sound waves.

6.6. Special Types of Cellular Automata

Several special types of CAs draw direct analogies from natural phenomena. Elementary Cellular Automata, defined by Stephen Wolfram, explore simple, one-dimensional rules that can lead to complex behavior [7], much like simple genetic rules can lead to diverse biological traits. Conway's Game of Life, a two-dimensional CA, mirrors the lifecycle of organisms with birth, survival, and death rules, showing how simple interactions can create complex, lifelike patterns [2]. Langton's Ant is an example of how simple local rules can produce unexpected, emergent behavior, similar to how individual ant movements result in organized colony paths [5].

Fuzzy Cellular Automata combine CA principles with fuzzy logic to handle uncertainty [12], much like how plants might adapt their growth based on varying environmental conditions rather than strict rules. Quantum Cellular Automata extend these concepts into the quantum realm, where cells can exist in superpositions of states, analogous to quantum particles exhibiting multiple states simultaneously [8].

These diverse versions of Cellular Automata reflect the richness and variety found in natural systems, allowing for the modeling and simulation of both simple and complex phenomena across different fields of study.

7. Usage Examples

In the following section, we present some articles where you can find implementations of cellular automata.

7.1. A computational tumor growth model experience based on molecular dynamics point of view using deep cellular automata

The article *A computational tumor growth model experience based on molecular dynamics point of view using deep cellular automata* discusses a novel approach to modeling cancerous tumor growth by integrating cellular automata (CA) with deep learning techniques, particularly convolutional neural networks (CNNs). This integration aims to leverage the strengths of both methods to create a robust and scalable solution for simulating tumor development.[17]

7.1.1. Overview of Cellular Automata and CNN Integration

Cellular automata are dynamic systems composed of a grid of cells, each of which can be in one of several states. The state of each cell at any given time is determined by a set of rules that consider the states of neighboring cells. Traditional CA models operate based on predetermined rules, which can be rigid and may not adapt well to varying conditions. By contrast, when CA rules are implemented within a CNN framework, these rules can potentially be learned from data, making the system adaptive and capable of handling more complex and varied scenarios.

7.1.2. Implementation Details

The article details how the CA is represented as a CNN. The process involves defining the CA as a mapping between possible pixel configurations in a binary image and corresponding output pixel values. This mapping is applied to randomly generated binary images to create training data, which are then used to train the CNN. The network architecture typically involves a 3×3 convolution layer followed by several 1×1 convolution layers. This setup allows the network to perform local operations repeatedly, mimicking the behavior of CA.

The integration of CA with CNNs provides several advantages:

- **Scalability:** Leveraging the parallel processing capabilities of modern GPUs optimized for CNN operations allows for faster simulations and predictions, especially for large grid sizes.
- **Granular Predictions:** By integrating CA rules within a CNN framework, it is possible to achieve detailed predictions at the cellular level. This granularity is crucial for applications like tumor growth prediction, where understanding cellular interactions at a micro-level can provide invaluable insights.
- **Adaptability:** Unlike traditional CA models that are constrained by predefined rules, the CNN-based approach allows for the rules to be learned from data, enhancing the system's ability to generalize and adapt to different conditions.

7.1.3. Results and Validation

The proposed model was validated by comparing its output to established mathematical models of tumor growth, specifically the Gompertz growth model. The results showed that the CNN-CA model aligns well with the Gompertz model, demonstrating its accuracy and robustness. Additionally, the model was tested on various tumor growth datasets, highlighting its versatility and predictive capabilities.

This research paves the way for more sophisticated and adaptable models in the field of tumor growth simulation, offering a promising direction for future studies in oncology.

7.2. Implementing Fuzzy Cellular Automata in Breast Cancer Image Segmentation

The article *Breast Cancer Images Segmentation using Fuzzy Cellular Automaton* presents an innovative approach for segmenting breast cancer images by integrating Cellular Automata (CA) with Fuzzy Logic, resulting in the so-called Fuzzy Cellular Automaton. This method aims to enhance the accuracy and flexibility of segmenting the mass region in mammograms, achieving highly promising results with an accuracy of 98.66% on the mini-MIAS dataset.[15]

7.2.1. Overview of Cellular Automata and Fuzzy Logic Integration

Cellular Automata (CA) are dynamic systems comprised of a grid of cells, each of which can be in one of several states. The state of each cell at any given time is determined by a set of rules that consider the states of neighboring cells. Traditional CA models operate based on predetermined rules, which can be rigid and may not adapt well to varying conditions. By incorporating Fuzzy Logic into CA, the model gains the ability to categorize pixels more flexibly, enhancing its adaptability to different imaging scenarios.

7.2.2. Implementation Details

The article outlines how CA is combined with Fuzzy Logic to form a Fuzzy Cellular Automaton. This process involves several key steps:

1. **Preprocessing:** The mammogram images are converted to grayscale, and noise is reduced using Gaussian Blur and Binary Threshold techniques. Background and pectoral muscle removal are performed to isolate the region of interest (ROI).
2. **Segmentation:** An improved version of the CA algorithm is applied to the preprocessed images. The segmentation starts with a pixel (seed) and expands to neighboring pixels with similar shades, computing the strength of the neighboring cells based on the CA rules.
3. **Fuzzy Logic Integration:** To refine the segmentation, Fuzzy Logic is applied. Two fuzzy sets are constructed based on the grayscale distributions of the strength determined by the CA algorithm. These sets model the presence of a certain pixel in the suspicious region and the contrary event.

7.2.3. Advantages of the Approach

Integrating CA with Fuzzy Logic offers several advantages:

- **Improved Accuracy:** The method achieves a high accuracy rate of 98.66% on the mini-MIAS dataset, significantly improving the precision of mass region segmentation.
- **Flexibility:** Fuzzy Logic provides a more adaptable technique for pixel categorization, allowing the model to handle variations in imaging conditions more effectively.
- **Visual Clarity:** The visually progressive nature of CA, combined with the adaptability of Fuzzy Logic, results in clearer and more interpretable segmentation results for medical specialists.

7.2.4. Experimental Results

The proposed approach was validated on the mini-MIAS dataset, comprising 322 mammograms, 118 of which contain suspicious masses. The results showed that the Fuzzy Cellular Automaton outperformed the traditional CA approach, with a mean Intersection over Union (IoU) score of 83.79%, compared to 65.11% for the CA approach.

8. Implementation Repositories

In this section, we present some repositories where you can find implementations of cellular automata.

Sandspiel

Made by Max Bittker, Sandspiel is a falling sand game that provides a creative and interactive environment for exploring cellular automata. Users can experiment with different elements, such as sand, water,

fire, and plants, and observe how they interact and evolve over time. Sandspiel is available online and offers a user-friendly interface for designing and simulating complex cellular automata systems.

You can access the repository at: <https://github.com/MaxBittker/sandspiel>

terra

JS library for cellular automata and simple biological simulations. textitTerra provides a flexible and extensible framework for creating and simulating cellular automata models. It includes various built-in rules and configurations for different types of automata, making it easy to experiment with different systems and behaviors. terra is designed to be user-friendly and accessible for both beginners and advanced users.

You can access the repository at: <https://rileyjshaw.com/terra/>

9. Conclusion

In this paper, we have explored the concept of Cellular Automata (CA) and its various types, including deterministic and probabilistic CAs. We have also discussed the importance of boundary conditions in CA models and highlighted some special types of CAs that draw analogies from natural phenomena.

Furthermore, we have presented two usage examples of CA in different domains. The first example demonstrates the integration of CA with deep learning techniques for modeling tumor growth, showcasing the scalability and adaptability of the approach. The second example showcases the integration of CA with fuzzy logic for breast cancer image segmentation, highlighting the improved accuracy and visual clarity achieved through this integration.

Overall, Cellular Automata provide a powerful framework for modeling and simulating complex phenomena in various fields. The versatility and adaptability of CA make it a valuable tool for understanding and analyzing natural systems. With further research and development, CA-based models can contribute to advancements in fields such as biology, medicine, and environmental science.

In conclusion, Cellular Automata offer a promising avenue for studying and simulating both simple and complex phenomena, providing insights into the dynamics of natural systems and enabling the development of innovative solutions in various domains.

References

- [1] J. Von Neumann, "The General and Logical Theory of Automata," in *J. von Neumann Collected Works*, A. Taub, Ed., 1963.
- [2] M. Gardner, "Mathematical games: The fantastic combinations of john conway's new solitaire game "life"," *Scientific American*, vol. 223, no. 4, pp. 120–123, 1970.
- [3] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, no. 3, pp. 601–644, 1983. DOI: 10.1103/RevModPhys.55.601.
- [4] S. Wolfram, "Universality and complexity in cellular automata," *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 1–35, 1984. DOI: 10.1016/0167-2789(84)90245-8.
- [5] C. G. Langton, "Studying artificial life with cellular automata," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 120–149, 1986. DOI: 10.1016/0167-2789(86)90237-X.
- [6] P. Rendell, "Turing machine universality of the game of life," *Fundamenta Informaticae*, vol. 29, pp. 231–243, 2002. DOI: 10.3233/FI-2002-292301.
- [7] S. Wolfram, *A New Kind of Science*. Champaign, IL: Wolfram Media, 2002, ISBN: 978-1-57955-008-0.
- [8] G. K. Brennen, D. P. O'Leary, and S. S. Bullock, "Criteria for exact qca initialization," *Quantum Information and Computation*, vol. 3, no. 3, pp. 234–242, 2003.
- [9] M. Cook, "Universality in elementary cellular automata," *Complex Systems*, vol. 15, no. 1, pp. 1–40, 2004. [Online]. Available: <http://www.complex-systems.com/pdf/15-1-1.pdf>.
- [10] J. L. Schiff, *Cellular Automata: A Discrete View of the World*. Hoboken, NJ: John Wiley & Sons, Inc., 2007, ISBN: 978-0-470-11484-7.
- [11] A. Dewdney, "Cellular automata," in *Encyclopedia of Ecology*, S. E. Jørgensen and B. D. Fath, Eds., Oxford: Academic Press, 2008, pp. 541–550, ISBN: 978-0-08-045405-4. DOI: <https://doi.org/10.1016/B978-008045405-4.00147-6>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780080454054001476>.
- [12] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Hoboken, NJ: John Wiley & Sons, 2010, ISBN: 978-0-470-27895-2.
- [13] L. De Mol, "Touring Machines," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Winter 2021, Metaphysics Research Lab, Stanford University, 2021. [Online]. Available: <https://plato.stanford.edu/archives/win2021/entries/turing-machine/>.
- [14] F. Berto and J. Tagliabue, "Cellular Automata," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, Eds., Winter 2023, Metaphysics Research Lab, Stanford University, 2023.
- [15] I.-A. Ion, C. Moroz-Dubenco, and A. Andreica, "Breast cancer images segmentation using fuzzy cellular automaton," *Procedia Computer Science*, vol. 225, 2023, 27th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES 2023), ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2023.10.087>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050923012450>.
- [16] A. Shafique, K. H. Khan, M. M. Hazzazi, I. Bahkali, Z. Bassfar, and M. U. Rehman, "Chaos and cellular automata-based substitution box and its application in cryptography," *Mathematics*, vol. 11, no. 10, 2023, ISSN: 2227-7390. DOI: 10.3390/math11102322. [Online]. Available: <https://www.mdpi.com/2227-7390/11/10/2322>.
- [17] H. N. Matin and S. Setayeshi, "A computational tumor growth model experience based on molecular dynamics point of view using deep cellular automata," *Artificial Intelligence in Medicine*, vol. 148, 2024, ISSN: 0933-3657. DOI: <https://doi.org/10.1016/j.artmed.2023.102752>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S093336572300266X>.
- [18] T. G. Gang, *Golly - A Game of Life Simulator*, Accessed: 2024-06-22. [Online]. Available: <http://golly.sourceforge.net/>.