Assignment 3
Cellular Automata

Finn Joel Bjervig[*], August Forsman[†] and Erik Turesson[‡]

1MA256 - Modeling Complex Systems,
Department of Mathematics,
Uppsala university, Sweden

August 31, 2022

[*]Electronic address: `jobj8920@student.uu.se`
[†]Electronic address: `aufo8456@student.uu.se`
[‡]Electronic address: `ertu2293@student.uu.se`

Cellular Automata Theory

The idea behind cellular automata is that fixed cells that make up an array will change throughout time depending on their current state and the state of their neighbors. The rules such as: what cells to consider as neighbors and what the response for each constellation of states in the neighborhood is, constitutes the model that drives the evolution of the state. Furthermore, the array, or the region in which the spectacle plays out, can be of one, two or three dimensions (excluding time as a dimension).

The 184-Rule

This method is named after the base ten representation of the binary code of the rule

$$184_{10} = 10111000_2$$

Sequence	111	110	101	100	011	010	001	000
Response	1	0	1	1	1	0	0	0

Table 1: *The response constitutes the binary code which equals the number 184 in base ten*

Consider a one-dimensional a array of length L with equally many ones and zeros. It has a periodic boundary condition such that the first and last digit in the array are neighbors. Then one iterates the array looking at the neighborhood around each digit in the array. For the 184-rule the radius is one, meaning we are looking at the current element $a(i)$ and the two adjacent: $a(i-1)$ and $a(i+1)$. So we focus on a three-digit long sub-sequence for each iteration. There are eight unique ways to construct such a sequence, and depending on which, the central digit will change as a response. The response for each unique sequence is in table 1. Beware that the digits do not change in the current array during the iterative process. Instead, the response is stored in a new array corresponding to the next time step. If this continues for say, 50 time steps, for a uniformly random initial state of length 50 (out of 2^{50} unique initial states) one has a slight possibility (approximately one in 1.126 quadrillion) to generate the course of events depicted in figure 1. Crank the settings up a bit, to $T = 1024$, $L = 2048$ and we got ourselves a more aesthetically pleasing piece, as seen in figure 2

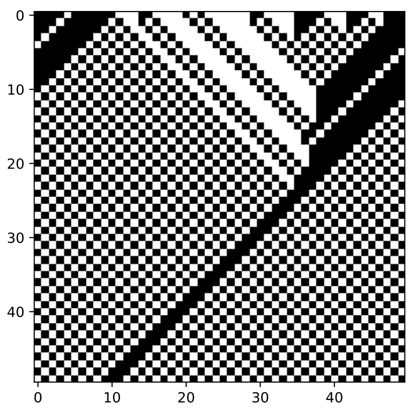


Figure 1: *Each row is the state and evolves with time in the the negative y-direction.* $T = 50$ *and* $L = 50$

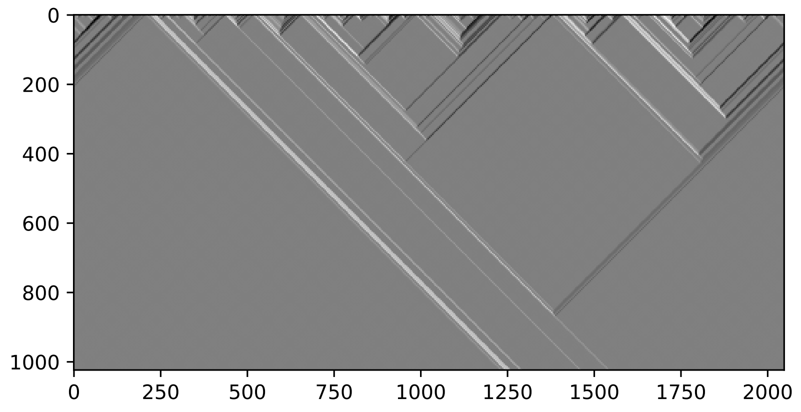


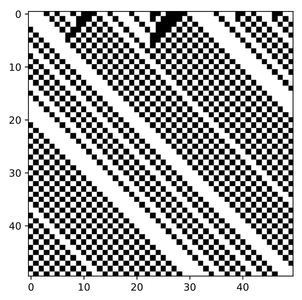
Figure 2: *With a higher resolution the figure makes for great abstract art, which you can tell your friends has a deep metaphysical meaning.* $T = 1024$ *and* $L = 2048$

(1) Moving into Equilibrium state

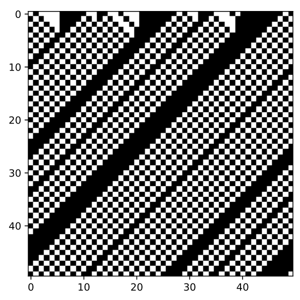
As time progresses, a pattern emerges regardless of the initial condition. It appears as if the cells are moving to the left, and most regions will converge to a checkboard pattern (depicted as gray in figure 2). If not, streaks of either ones or zeros travel diagonally across time and space. Looking at the figures 2 the ones (black) travel left and zeros (white) to the right with time. There is also a periodicity because of the arrays boundary condition. How far these streaks travel depends on the initial condition and is thus random.

(2) Changing density of ones and zeroes

Previously the digits for the IC has been picked from a uniform distribution such that there is an equal chance of picking a one as a zero, meaning there is a share of $\rho = 1/2$ active sites. What will happen if we change this parameter to something non-symmetric, as $\rho = 2/5$? This means there are 40% active sites present in the initial state. Future states will then have less active states and vice versa. See figure 3. Further, if one counts the share of active sites of the final state (last row in figures) they are the same as for the initial state (first row in figures). This can be explained with the structure of the rules in table 1, and the fact that there are equally many ones as zeros in the responses. The "drifting" of white and black regions in the plots are also a result of how the rules are constructed.



(a) $\rho = 2/5 = 40\%$



(b) $\rho = 3/5 = 60\%$

Figure 3: *The 184 rule with varying shares of active states, $T = 50$, $L = 50$. Clearly more active sites in the IC results in more active sites in future states (plot a), and vice versa (plot b).*

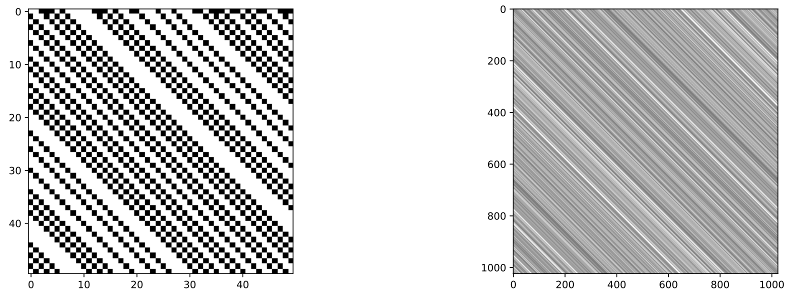
Rule 176

Now let's explore another rule

$$176_{10} = 1011000_2$$

Sequence	111	110	101	100	011	010	001	000
Response	1	0	1	1	0	0	0	0

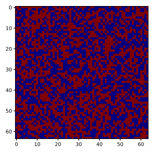
Table 2: *The response constitutes the binary code which equals the number 176 in base ten*

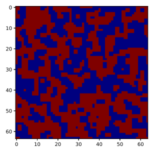


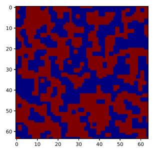
(a) $T = 50$, $L = 50$ (b) $T = 1024$, $L = 1024$

Figure 4: *The 176 rule of different resolution in time and space*

2-D Public Opinions


(a) t = 0


(b) t = 2


(c) t = 4

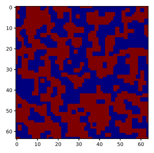

(d) t = 6

Figure 5: *neighborhoods of public opinions. Red corresponds to conservatives, and blue to liberals. The domain is square and of size* $N = 64 \times 64$

Per usual, the initial condition (figure 5a) is a uniformly random distribution of ones and zeros in the square 2D domain. Ones represent Liberals and zeros conservatives. In figure 5 the respective opinions are assigned the appropriate colors of blue and red, reflecting the U.S two political parties. Further, we see the emergence of neighborhoods over time (from left to right). The result is rather intuitive since the majority votes prevail as defined by the algorithm, which has a *von Neumann neighborhood*, meaning one neighborhood contains a central, left, right, top and bottom cell, see figure 6 for examples of how different scenarios play out. If the central node is of a certain opinion and the majority of the neighborhood shares that opinion, the central node will not change. But if the majority opinion is against the central node's opinion, it will change and join the majority. See the attached code for details

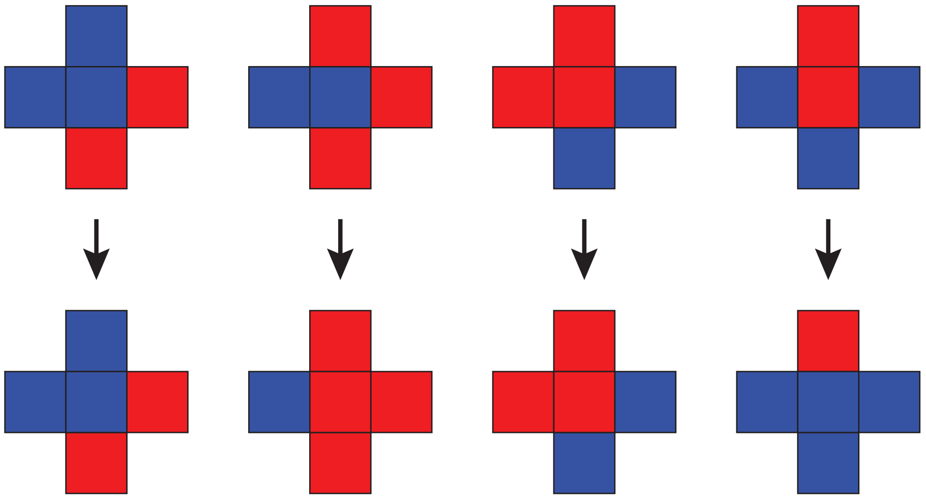


Figure 6: Examples of how the opinion of one individual evolves depending on the neighbors opinions.

Python Code

Code files are attached with submission.