Christopher Moran

Dr. MacLennon

CS 420

24 February 2020

Project 2: Creation of Spatial Structure by Activator/Inhibitor Cellular Automaton

Introduction

The purpose of this assignment was to introduce to the students how cellular automata could be activated and inhibited based on a variety of factors. In particular, the parameters that were changed over the course of the project were R1, R2, H, J1, and J2. R1 referred to the maximum distance at which nearby cellular automata would affect each other, and R2 referred to the maximum distance at which far away cellular automata would affect each other. H referred to the bias factor. J1 held the value that determined the activation system threshold, and J2 held the value that determined the threshold of the inhibition system. As these parameters were changed, the object was to allow the cellular matrix to run its course as the cellular automata interacted until it reached a deadlock. Then, using this final product, the correlation, entropy, joint entropy, and mutual information could be discerned.

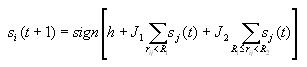
Methodology

The heart of this lab revolved around a 30x30 2d array whose elements were randomly set to either a 1 or a -1. Using this array, the first step in the project was to allow the array to progress along its path based upon several parameters. In particular, the parameters that were changed over the course of the project were R1, R2, H, J1, and J2. R1 referred to the maximum distance at which nearby cellular automata would affect each other, and R2 referred to the maximum distance at which far away cellular automata would affect each other. H referred to the bias factor. J1 held the value that determined the activation system threshold, and J2 held the value that determined the threshold of the inhibition system. As these parameters were changed, the object was to allow the cellular matrix to run its course as the cellular automata interacted until it reached a deadlock.

The process behind the progression of the array had to do a few simple calculations. Each automata, represented as a cell within the array, would look around at the other cells around it, as well as on the other edge, as the array represented a torus. Then, based upon the distance between it and the cells around it, found by this formula:

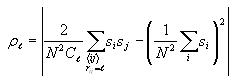


Each cell was able to determine whether or not it would change its state, using this formula:



In this equation, s with the j subscript represents the sum of the cell values around the s subscript i cell. The first summation represents all of the cells within radius R1 of the Si cell, and the next one represents the sum of the cells between R1 and R2. Everything beyond R2 was disregarded. If the right side of that equation was greater than 0, then the cell would have been set to 1. Otherwise it would have been set to -1. This would happen to each cell, randomly, as it progressed through the entire array. Once the array had been traversed through, the program would either repeat the process if changes had been made, or it would determine that there could be no further changes made, as there would have been none in the preceding process.

Once that was done with, the program would have been left with the final array product, which is needed to compute the rest of the project. The first thing that would have been found following the final array is the correlation. The correlation can be found using this equation:



This equation finds the sum of all correlations with the distance set to l, where l ranges from 0 to 14.

Once that was completed, the program would move on the calculate the entropy of the array. This is a multistep process, though none of them are all too complicated. The first step here would be to calculate the sum of converted cell values. Converted cell values can be represented by this equation:



Where s simply represents the 1 or -1 value of a cell. This value would be summed over the course of the entire array, as represented by this equation:



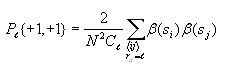
Which gives the probability of state 1. Conversely, the probability of state -1 is given with this equation:

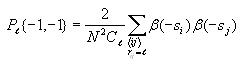


These two values are needed to calculate the overall entropy, represented as H(s)



This value is required to calculate the mutual information, but first the joint entropy needs to be calculated. This is done in a similar way to how the correlation was calculated, but there is also a lot of commonality with the original overall entropy calculation. However, before the program can deduce the joint entropy, it needed to first find a few different values using these equations:







Then, using these three entropy values, the joint entropy can be calculated with this equation:



It should be noted that these equations are run through for each length l, where l ranges from 0 to 14

Lastly, the mutual information was calculated, and this is an even simpler calculation:



Though, like the joint entropy and correlation, it has to run through for each value l, ranging from 0 to 14.

Results