Final Project Report: Automatic Detection of Interesting Cellular Automata

CS 294-082 – Experimental Design for Multimedia Machine Learning (Graduate) – Fall 2020

|  |  |
| --- | --- |
| Randy Fan, Qitian Liao (Student Authors)  University of California, Berkeley  [randyfan@berkeley.edu](mailto:randyfan@berkeley.edu), liao1024@berkeley.edu | Dr. Gerald Friedland (Faculty Advisor)  Department of EECS, University of California, Berkeley  fractor@eecs.berkeley.edu |

**ABSTRACT**

**The introduction of Machine Learning and Deep Neural Networks has revolutionized a variety of task. Our project explores the potential of using Deep Neural Networks to detect interesting Cellular Automata. A cellular automaton is a collection of colored cells on a grid that evolves according to a set of rules. The rules are applied iteratively for as many time steps as desired.**

One challenge is that Cellular Automata tend to have huge parameter search spaces to find interesting results within. The vast majority of rules within this space will be junk rules with only a small fraction of a percentage being interesting rules. Between the boring rules that either die out tend to chaos, there is a tiny sweet spot of interesting patterns. Manually searching for these patterns would be unrealistic. In this paper, we will discuss our approach to detect the interesting patterns using image processing, Convolutional Neural Networks (CNNs) and ImageNet.

KEYWORDS

Cellular Automata, Image Processing, Convolutional Neural Network, ImageNet, Dynamic patterns

1 INTRODUCTION / PROBLEM

In this paper, we will focus on outer totalistic generations of two-dimensional Cellular Automata. They consist of a two-dimensional grid of cells, which live, die, or are in a “dying” transition date depending on a set of rules. Specifically, the cells are updated according to their previous values, and the sum of the values of the other cells in the neighborhood. We decided to use Moore neighborhood for the update rules, which are the surrounding eight cells of a center cell. An alternative would be Neumann neighborhood, which only includes four cells. Rules are classified as boring if they lead to a pattern which is static noise with no discernable patterns moving across the screen. For this project, we define interesting rules specifically as those that have clear gliders (small patterns that move across the grid) with some individual characteristics. Under most circumstances, it is impossible to tell whether a rule is interesting or boring just by looking at the parameters. A user would have to randomly go through thousands of random rules before finding an interesting one. Even more daunting, under the assumption that we use a Moore neighborhood and a maximum of 10 possible states, there are 29 survival rules, 29 born rules, and 210 states, which leads to a total of 228 combinations of rules. Because of this gigantic number of possible combinations, automatic detection of interesting rules will be very helpful, and it will make sure that users do not have to manually go through the process.

2  BACKGROUND / RELATED WORK

// TODO

3  METHOD

3.1  Data Collection Pipeline

To obtain boring rules, we manually went through random examples, and collected rules that died out immediately or generated static noise or boring non-glider patterns. For interesting rules, we used Cellular Automata Rules Lexicon, and examples provided in Visions of Chaos and only recorded those with gliders. We ended up with 105 boring and 35 interesting set of rules.

We generalized the Cellular Automata generation algorithm as follows:

**Cellular Automata Generation Algorithm**

if center\_cell == max\_state:

for num\_neighbors in survive\_arr:

if total - 1 == num\_neighbors:

return center\_cell

return center\_cell - 1

elif center\_cell != 0 and center\_cell != max\_state:

return center\_cell - 1

else:

for num\_neighbors in born\_arr:

if total == num\_neighbors:

return max\_state

return 0

We used the algorithm above to generate frames and create images. Next, we applied data augmentation due to the limited number of rules that we identified. We reused the boring rules 10 times, and the interesting rules 30 times with different initial configuration. We generated 140 images for each of the patterns.

Our next step is image stitching. We chose to use frames from 100 to 108. Because when we were running examples, we noticed that this is roughly when the patterns were all apparent. Each pattern corresponds to one stitched together 3 by 3 image.

Then we tried to train it with a naive convolutional neural network with four layers with batch size 16 and for 10 epochs, and we were able to achieve a pretty good validation accuracy of 93.18%.

4  CONTRIBUTIONS AND FUTURE WORK

// TODO

ACKNOWLEDGMENTS

Randy Fan, Qitian Liao

REFERENCES

|  |  |
| --- | --- |
| [1] |  |
| [2] | Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.dy of Positive Youth Development. Journal of Early Adolescence 25 (1): 17-71. |
| [3] | . |
| [4] | . |
| [5] | . |

Context for Reviewers

The second page, which will be for the reviewers only, and not included in the proceedings, should provide more context for reviewers. You may format this page as you prefer. Examples of content you might provide on this page include: description of tentative layout and content of the poster, description of any handouts that may be available, brief description of author expertise and background as it relates to the poster (remember not to include actual author names or institutions in your anonymous submission). You should also justify how your poster will engage participants in discussion.