Cellular Automaton

And

Spiking Cellular Automaton

Andrew D Newman

University of York

February 2021

3rd Year Project Initial Report for degree of BEng in Electronic Engineering

# Introduction

The goal of this project is to make a cellular automaton in a similar style to Conway’s Game of Life [1] using a biologically inspired Spiking Neural Network (SNN) for the cells.

Life is a rectangular grid automaton in which cells grow or die depending on the number of living neighbours they have. Complex behaviour emerges from the simple rules. Life cells can be viewed (loosely) as 1st generation neurons, being either alive or dead.

In 2nd generation neural nets each neuron has a potential, essentially becoming a real valued entity instead of a Boolean.

An SNN is a 3rd generation neural network in which neurons communicate via spikes of voltage, which more closely resembles the working of a biological brain. This has been shown [2] to have greater computational power per neuron than a 2nd generation network, and it is hoped that this will lead to more interesting behaviour of the automaton.

# Specification

This project will be entirely software oriented. Where possible platform independence will be aimed for, but the Microsoft Windows OS on an x64 architecture will be the primary target.

The software will display an automaton in a regular grid of cells in which the behaviour of each cell is determined by the state of its neighbours. The general goal of an automaton of this nature is to demonstrate complex behaviour emerging from a simple set of rules.

Conway’s objectives for the Game of Life [1] will serve as a starting point for our objectives for our own automaton:

1. “There should be no initial pattern for which there is a simple proof that the population can grow without limit.”
2. “There should be initial patterns that apparently do grow without limit.”
3. “There should be simple initial patterns that grow and change for a considerable period of time before coming to end in three possible ways: fading away completely (from overcrowding or becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering an oscillating phase in which they repeat an endless cycle of two or more periods.”

To these we add our own additional requirements relating to SNNs:

1. Information flow between cells must be in the form of spikes similar in nature to those conveyed by axons in biological nervous systems.
2. The behaviour of the pattern should be distinguishable from behaviours which are possible with a 1st or 2nd generation neural net.

We consider objective 1 to be rather vague (especially since the game of life itself has some simple examples that do grow without limit [1]) but its intention will serve as a guideline.

We have no proof that objective 5 is possible but will aim for it.

Because we begin this project with no idea what kind of SNN will allow us to meet those objectives, or how it will need to be configured or connected, the objectives for the project are somewhat different to the objectives for the automaton, and our goal is to create software which can then be used to find an automaton meeting the first set of objectives.

1. The software implementation will be organized according to the Model/View/Controller pattern
2. The software will allow different sizes of automaton to be modelled
3. The software will allow the simulation to be run at different speeds, stopped, or stepped one frame at a time
4. The software will allow selection of different neuron models
   1. There will be a LIF (Leaky Integrate and Fire) model
   2. There will be an Izhikevich model [3]
   3. There may be other models
5. The software will allow easy configuration of the selected model
   1. Any appropriate parameters will be made available for change via the GUI
   2. It will be possible to persist interesting combinations of parameters for reuse
6. The software will allow different spike shapes to be configured
7. The software will allow different connection patterns of neurons via synapses to be configured
   1. There will either be a synapse matrix editor built into the software or it will have the ability to load synapse matrices from image files
8. The software will display an animation of the automaton as it progresses
   1. The software may provide alternative display styles such as spikes vs neuron potentials
9. The software will allow different starting states to be configured.
   1. There will either be a state editor built into the software or it will have the ability to load states from image files
10. The software will allow a given starting state to be reused
    1. A given starting state, used multiple times with the same model and configuration, will give the same progression and end state each time

# Ethics

We do not anticipate any ethical issues arising in this project, beyond those typical of any other software project. This specific research will make only sporadic use of one computer.

From an environmental points of view it may be worth nothing that neural nets are very processor intensive and that if projects of this nature became common place we would expect them to drive a similar energy consumption pattern to that seen in bit coin mining (which is more energy intensive than actual mining [4]). Purpose built hardware should mitigate this effect to some extent.

# Literature Overview

## Fundamentals of Computational Neuroscience [2]

## This book makes a good case for spiking neuron behaviour being more than just the average rate of fire of the neurons (which is the justification for using 3rd generation neurons instead of 2nd), but then spends most of its time working in terms of rate of fire instead of spikes. It is clear how this makes the maths easier to handle, but it doesn't really provide the basis for the neuron models this project will need.

## Dynamical Systems in Neuroscience [5]

## This book is only partially relevant, but it is interesting reading. In particular, it makes it clear what the Izhikevich algorithm is actually doing. Page 17 describes the 4 possible modes of operation necessary to reflect the behaviour of biological neurons and I can see no way to create such a system using less than 2 variables, and I doubt it is possible with two variables without at least one quadratic (because a bifurcating system needs two attractors and we don't get that without two solutions to our equation).

## There is also an explanation of the Hodgkin-Huxley model, which attempts to model the actual behaviour of biological neurons, at least in the specific case of a giant squid axon. This model is clearly inappropriate for a cellular automaton (it's too complicated and slow to simulate) but could serve as the basis for a much faster approximation.

## Networks of Spiking Neurons: The Third Generation of Neural Network Models [6]

This paper makes a strong case for the time it takes for a firing neuron output to be received by another neuron to be of greater computational value than the weight of the connection (see section 2.1) and that the reason spiking nets are more computationally powerful than generation 1 and 2 nets is due to this, which suggests that an automaton could be constructed with fixed synapse weights, but with varying synapse delays.

## Simple Model of Spiking Neurons [3]

## This paper offers an algorithm for a biologically realistic spiking neuron that can be simulated efficiently. This model appears to be very popular in other research and we will be referring to it as the Izhikevich model in the rest of this review.

## There is a description of different biological neuron firing types that the model can mimic, which is an interesting reference, and suggests our cellular automaton should be able to contain multiple cell types in a single run.

## At the end of section II there is a suggestion (without clarification) that parameters u, a and b in the algorithm can be substituted with vectors in order to simulate slow conductance, which ties in nicely with the main point from W.Maass [6], but a conflicting statement that the authors don't find it necessary to do so (at least, not for cortical neurons).

Much of the paper is taken up demonstrating how accurately the algorithm can mimic the behaviour of biological cells, which is a nice to have and a good argument for trying it, but not essential.

It is pointed out that the potential of a neuron in this model can grow without limit and is only prevented from doing so by the reset that follows a spike being fired. However, although the Izhikevich model can act as an inhibitory neuron and even mimic the output of the biological ones in the brain, there does not appear to be anything limiting the negative potential a neuron can have. This is a general issue through all the literature reviewed here - it seems the existence of inhibitors is acknowledged but mostly ignored. We suspect this is because most work on neural nets has been geared towards forward connected nets that respond to an input event, rather than a continuously updating system such as our automaton that needs to self-regulate.

Izhikevich claims a 1ms simulation time for 10,000 neurons with 1,000,000 synapses. That is only a 100x100 grid of neurons, which is rather small for an automaton and would limit the size of feature we could hope to detect. It is also only 100 synapses per neuron, rather than the 10,000 typical of the brain [2]. There is a balance to be made between designing for speed and premature optimization and we will try to be reasonable, but it is obvious even without profiling that efficiency will be an important factor here.

We note that an automaton of this nature is extremely parallelizable and any attempt to fully optimize it should almost certainly be targeting a GPGPU. Our project needs to be flexible and will therefore target the CPU, and the investigative phase will only aim to be fast enough to be usable.

## Which Model to Use for Cortical Spiking Neurons? [7]

## This is a very readable summary of the pros and cons of different models of spiking neurons, and as such can also act as a list of different models to try in our automaton. It also makes an interesting observation about which models can display chaotic behaviour (which we want, because it encourages emergent behaviour) and how computationally expensive they are (we feel we should be prioritizing fast over biological accuracy here).

## Spiking Neural Networks [8]

## This is a summary of the history of Spiking Neural Net development. It repeatedly highlights the high computation cost of SNNs as a major problem, which strengthens our belief that efficiency will be important, but is otherwise not especially useful.

## Introduction to spiking neural networks: Information processing, learning and applications [9]

## This paper has two sections, the first being a concise run down on different mechanisms through which spike timing can carry information, which is a particularly useful reference.

## The second section is about training SNNs and is probably not applicable to an automaton.

## Cognitive Computing Building Block: A Versatile and Efficient Digital Neuron Model for Neurosynaptic Cores [10]

## The goal of the research in this paper is to create a neuron model that is "sufficient to support useful and interesting cognitive algorithms, while the cost should be no more than necessary in terms of power, area, and speed".

## They state "we were able to qualitatively replicate the 20 behaviors of the Izhikevich dynamical neuron model using a small number of elementary neurons" which is interesting. It implies a simpler model that has the Izhikevich model emergent from it, and emergent behaviour is a good thing in cellular automatons.

IBMs aim is a neuron model that can be effectively implemented in custom hardware, but a software simulation of it is certainly possible and should be efficient. They include a pseudo code implementation of their neuron and we would like to try it.

## There is also a pseudo code LIF implementation in here, but it uses a fixed leakage rate, which seems strange (an exponential decay appears to be more popular [2]). It also has no refractory period, which mean its output is not necessarily spikey at all.

## The algorithm works entirely in integer (fixed point) arithmetic, which is great for hardware implementations but means I may need a different kind of synapse matrix and spike to implement it in my own code, which could prove awkward. Fixed point maths is possible in software but not generally built into the available languages. We might assume that using floats instead will be close enough.

## There are many arguments for combining multiple kinds of neuron and our design will account for this from the start.

## A timeline of neuron models from 1907 to 2013 is provided, which presents some interesting search terms for additional models to try.

## Finally, this paper mentions stochastic integration (a random chance of synapses being integrated) which seems a very popular idea, and noise is given as an explanation for the speed of the brains response to visual stimuli [2] [5]. Unfortunately, any random influence will break repeatability in our automaton, which is one of our objectives. The same pattern of inputs to an automaton would usually result in the same behaviour, regardless of where or when it occurs, and we would rather respect that than take the biologically more realistic stochastic approach.

## A review of learning in biologically plausible spiking neural networks [11]

## This paper is a roundup of our current understanding of learning algorithms and their relationship with what we know of biological methods. This is interesting, but not especially useful to us as we do not anticipate our automaton learning things (we have no synaptic plasticity).

## However, it does make a good argument early on for both synaptic weight and delay being incorporated into synaptic connections, and we agree with it.

## A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve [12]

The original HHM (Hodgkin Huxley Model) paper, in which a model for the behaviour of a giant squid neuron is derived from experimental data. The result is a 4-dimensional family of exponential differential equations. We are curious to see if the biological approach gives more behaviour options than the various 2-dimensional options that seem more common.

Most of this paper is dedicated to deriving the equations from experimental data and verifying that they predict experimental results, but it does contain the necessary equations in a section titled "numerical methods".

Our expectations are that HHM will be far too slow for an automaton [7], but we are interested in seeing it anyway.

## An efficient method for solving fractional Hodgkin–Huxley model [13]

This paper presents a more efficient way to iterate the Hodgkin-Huxley model which we can consider but uses a subset of maths that we are not familiar with and will probably have to be considered as out of scope as a result.

## Classification of different neuron behavior by designing spiking neuron model [14, pp. 25-30]

A simple model with only two equations and four parameters, which is very reminiscent of the Izhikevich model, refers to it as inspiration and provides a similar set of possible output forms. The paper does not make it particularly clear why this model might be preferred over Izhikevich, but they do demonstrate their behaviours in response to short pulses of DC current, where Izhikevich seems inclined to use a unit step function as his input. This is encouraging, since we will be using spikes as our inputs and a DC pulse is more representative. Unfortunately, the equations used include an exponential which is also going to make them less efficient.

## Design Sketch

For reasons of speed and familiarity the controller will be written in C++. For simplicities sake the View will therefore also use C++. We expect to use Qt for the GUI in the View but will refrain from making the controller dependent on it.

For the automaton to be "interesting" it needs to avoid quickly collapsing into a quiescent state (see objective 3). For this to be true the number of neurons which fire as a result of N other neurons firing must average at least N. However, if the average is greater than N then we can expect the automaton to quickly saturate instead, with everything firing as fast as it is able, which is also uninteresting. Life avoids this by killing any cell that has 4 or more neighbours, but it does not seem appropriate to directly prevent a neuron from firing when it is overstimulated. Instead we expect to need inhibitory neurons to maintain balance. We feel quite strongly that an automaton should be a uniform grid of cells if possible, and it is not clear what pattern of inhibitory neurons would be appropriate to insert. Instead we will allow the creation of multiple layers of neurons. A layer of inhibitory neurons can then be included if desired.

For defining the synaptic connections between neurons, we will use another 2-dimensional array of synapses, each specifying a weight and time delay for its associated connection (see Figure 1). We will refer to this as a Synapse Matrix. It has similarities to the Convolution Matrix commonly used in graphics editing packages [15].

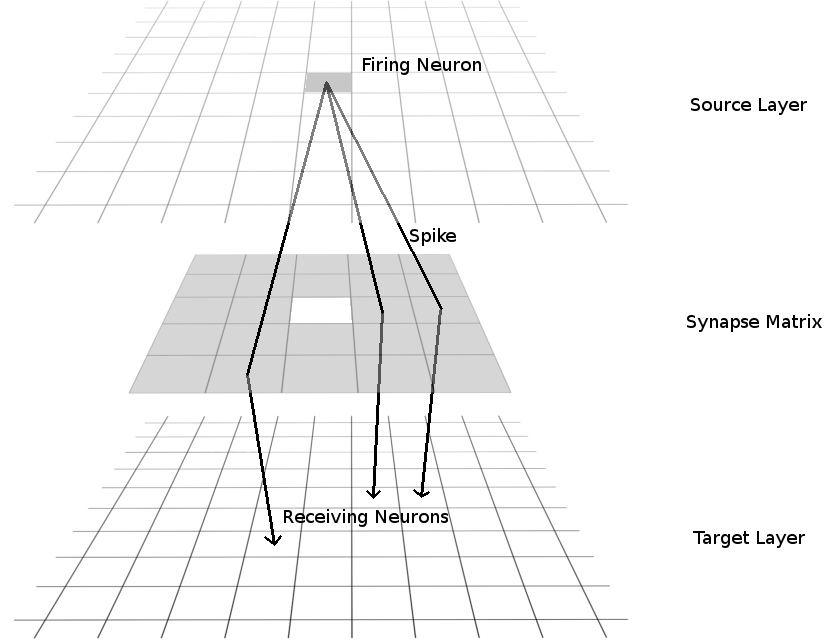


Figure 1 A synapse matrix connecting two layers of neurons

It will be possible for the source and target layer to be the same.

In the interests of efficiency, we will fire spikes along synapses using an event-based mechanism instead of polling. This has been used by IBM to decrease power consumption in True North [10] and will serve to decrease CPU usage for us.

# Timetable

|  |  |  |
| --- | --- | --- |
| Phase | Description | Target Delivery Date |
| Research | Reading and reviewing literature related to SNNs and writing this document. | 2021-02-04 |
| Interface | Building a GUI to use for experimenting with different models and parameters. | 2021-02-25 |
| Modelling | Building the models and experimenting with them to find an automaton that meets the objectives. | 2021-03-25 |
| Finalization | Decide on and precisely specify an automaton ruleset that gives good results. Potentially reimplement in a specialized form, depending on time and characteristics. | 2021-04-20 |
| Demonstration | Prepare for and give a presentation on the project. | 2021-05-04 |
| Report | Finish and deliver project report. | 2021-05-13 |

This project will take an agile approach, on account of not knowing where it will end up. I am confident of the timetable above regarding the interface phase, but modelling is more open ended, and finalization could take anything from 1 day to 30 depending on what is needed when we get there.

# References

|  |  |
| --- | --- |
| [1] | M. Gardner, “MATHEMATICAL GAMESThe fantastic combinations of John Conway's new solitaire game "life",” *Scientific American,* pp. 120-123, 10 1970. |
| [2] | T. P. Trappenberg, Fundamentals of Computational Neuroscience, Truro: Oxford University Press, 2002. |
| [3] | Eugene.M.Izhikevich, “Simple Model of Spiking Neurons,” in *IEEE Transactions on Neural Networks*, 2003. |
| [4] | M. J. Krause and T. Tolaymat, “nature.com,” 05 11 2018. [Online]. Available: https://www.nature.com/articles/s41893-018-0152-7. [Accessed 12 2020]. |
| [5] | E. M. Izhikevich, Dynamical Systems in Neuroscience, The MIT Press, 2007. |
| [6] | W.Mass, “Networks of Spiking Neurons: The Third Generation of Neural Network Models,” Elsevier Science Ltd., Graz, 1996. |
| [7] | Eugene.M.Izhikevich, “Which Model to Use for Cortical Spiking Neurons?,” *IEEE Transactions on Neural Networks,* vol. 15, no. 5, 2004. |
| [8] | S. Ghosh-Dastidar and H. Adeli, “Spiking Neural Networks,” *International Journal of Neural Systems,* vol. 19, no. 4, pp. 295-308, 2009. |
| [9] | F. Ponulak and A. Kasinski, “Introduction to spiking neural networks: Information processing, learning and applications,” *Acta Neurobiol Exp,* vol. 71, pp. 409-433, 2011. |
| [10] | A. S. Cassidy and etal, “Cognitive Computing Building Block: A Versatile and Efficient,” IBM, 2013. |
| [11] | A. Taherkhani and etal, “A review of learning in biologically plausible spiking neural networks,” *Neural Networks,* vol. 122, pp. 253-272, 2020. |
| [12] | A. L. Hdgkin and A. F. Huxley, “A QUANTITATIVE DESCRIPTION OF MEMBRANE CURRENT AND ITS APPLICATION TO CONDUCTION AND EXCITATION IN NERVE,” *Journal of Physiology,* vol. 117, pp. 500-544, 1952. |
| [13] | A. M. Nagya and N. H. Sweilamb, “An efficient method for solving fractional Hodgkin–Huxley model,” *Physics Letters,* vol. 378, pp. 1980-1984, 2014. |
| [14] | A. Kumar, S. Kansai and M. Hanmandlu, “Classification of different neuron behavior by designing spiking neuron model,” in *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, Tirunelveli, 2013. |
| [15] | GIMP, [Online]. Available: https://docs.gimp.org/2.8/en/plug-in-convmatrix.html. [Accessed 12 2020]. |