

CAM-BRAIN

Growing an Artificial Brain with a Million Neural Net Modules Inside a Trillion Cell Cellular Automata Machine

Hugo de Garis*

Abstract

This paper reports on a research project which aims to build (i.e. grow/evolve) an artificial brain by the year 2001. This artificial brain should contain thousands of interconnected artificial neural network modules, and be capable of controlling approximately 1000 "behaviors" in a "robot kitten". The name given to this research project is "CAM-Brain", because the neural networks (based on cellular automata) will be grown inside special hardware called Cellular Automata Machines (CAMs). Using a family of CAMs, each with its own processor to measure the performance quality or fitness of the evolved neural circuits, will allow the neural modules and their interconnections to be grown/evolved at electronic speeds. State of the art in CAM design is about 10 to the power 9 or 10 cells. Since a neural module of about 15 connected neurons can fit inside a cube of 100 cells on a side (1 million cells), a CAM which is specially adapted for CAM-Brain could contain thousands of interconnected modules, i.e. an artificial brain.

Keywords CAM-Brain, Cellular Automata (CAs), Cellular Automata Machines (CAMs), Artificial Brains, Neurite Networks, Genetic Programming (GP), Genetic Algorithms (GAs), GenNets (Genetically Programmed Neural Network Modules), CA Networks, Artificial Nervous Systems, Incremental GP, Biots (Biological Robots), Darwinian Robotics, 1000-GenNet Biots, GenNet Accelerators, GenNet Shaping, CA Neurons, Darwin Machines, Nanotechnology, NanoCAM-Brain.

* Brain Builder Group, Evolutionary Systems Department, ATR Human Information Processing Research Laboratories, 2-2 Hikari-dai, Seika-cho, Soraku-gun, Kansai Science City, Kyoto, 619-02, Japan. tel : + 81 7749 5 1079, fax : + 81 7749 5 1008, email : degaris@hip.atr.co.jp

1. Introduction

The CAM-Brain Project introduces a new subfield into neural network research, which the author calls "Neurite Networks", where the distinction between the two is that with "Neurite Networks", the neural network gets GROWN, i.e.

it has an embryological component. A "neurite" is a neurobiological term meaning a "baby neuron which grows connections with other neurites". The artificial neurite networks introduced in this paper are based on a cellular automata (CA) network whose branchings are "Genetically Programmed" (i.e. they are grown under the control of a Genetic

Algorithm). A sequence of CA signals is sent down the middle of a CA "trail" (see **Figs. 9-11** at the end of this paper). When a signal hits the end of a trail, it makes the trail extend, turn left, turn right, branch left, branch right, split, etc., depending upon the state of the CA signal. These signal sequences are treated as the chromosomes of a Genetic Algorithm. Once the CA network is formed, other CA state transition rules make it behave like a neural network. The fitness of this CA based neural network is measured in terms of how well it performs some task, e.g. controlling some behavior of a biological robot (biot). ATR's "Brain Builder Group" (a part of the Evolutionary Systems Department) hopes to use such ideas to build Darwin Machines (i.e. machines which evolve), based on Cellular Automata Machines, as a tool to build an artificial brain.

As stated in the abstract, one of the research aims of ATR's new Evolutionary Systems Department is to build an artificial brain by 2001. The very title of the department reflects one of its fundamental assumptions, i.e. that hyper complex systems (such as biological brains or embryos) will probably have to be built using an evolutionary approach rather than using human design. Complexity levels will be so high (especially when nanotechnology (i.e. molecular scale technology) becomes a reality) that no human being will be able to predict or even analyze how these systems function. The author has given the concept of "evolutionary building of complex systems" a label. He calls it "Genetic Programming (GP)" [1,2,3,4].

This paper shows how an artificial neural network, based on cellular automata can be grown, using GP techniques. The ideas and results of this paper will serve as the conceptual basis for the construction of what the author calls "Darwin Machines" [2]. A Darwin Machine is a special hardware device used to perform GP in parallel. For example, Cellular Automata Machines could function in parallel to evolve the neurite networks described below. Each CAM would have a conventional programmable processor to measure the fitness of the evolved neurite network. A central processor could then perform the GA aspects of the evolution (e.g. calculate the next generation of chromosomes etc). Alternatively, a more distributed GA could be performed, where each CAM and its processor communicates only with its neighbors. It is hoped that using these Darwin Machines, it will be possible to build/evolve/GP a large number of neurite network modules and their connections to build an artificial brain capable of giving a biological robot (biot) some 1000 "behaviors".

This paper consists of the following sections. Section 2 gives a brief introduction to cellular automata and how cellular automata trails can be evolved into cellular automata networks. Section 3 expands on the initial ideas of section 2, especially in explaining how CA trails can be made to behave like neural networks. Section 4 gives details on the state transitions of the cellular automata trails, and section 5 presents ideas for future research.

2. The Genetic Programming of Cellular Automata Trails

A cellular automata is a set of "cells" (e.g. squares in a 2D grid, or cubes in a 3D grid) each of which has one of a finite number of states. State transition rules (applying to all cells in the grid) determine how a cell updates (synchronously) its state depending upon its present state and the states of its neighbors. **Fig. 1** shows an example of a CA state transition rule.

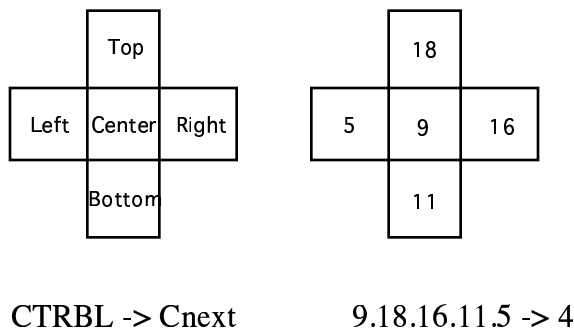


Fig. 1 A Cellular Automata State Transition Rule

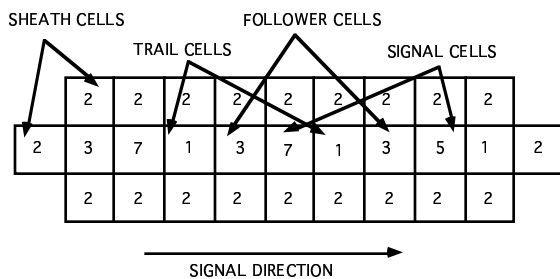


Fig. 2 A Cellular Automata Trail

A 3 cell wide CA "trail" as shown in **Fig. 2** can be fed a sequence of CA signals which propagate down the middle of the trail until they hit the end. When they do, CA state transition rules are defined

so that the trail is extended by one square, or made to turn left, turn right, split, branch left, branch right etc (e.g. [5]). The sequence of these CA signals is then evolved using a conventional Genetic Algorithm [6]. When one trail collides with another, a "synapse" is formed, as shown in **Fig. 3**. The two cells of the synapse then absorb oncoming signals, thus keeping the configuration of the intersecting trails intact. **Figs. 9-11** at the end of this paper show the results of a CA network evolution. The fitness in this case was simply the number of synapses formed. In **Fig. 9** there were 4 "CA neurons" (not shown). The chromosome was split into 4 and fed simultaneously into the starting points of the 4 CA trails.

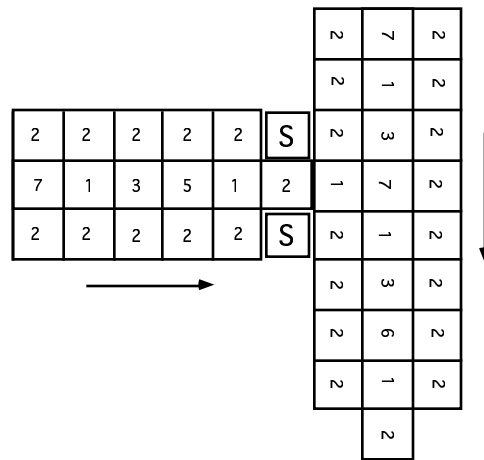


Fig. 3 A Synapse Between 2 CA Trails

3. Cellular Automata Based Neurite Networks

One can then evolve CA networks. There is too little space in this short paper to go into the many details, but CA rules can be defined to repair and clean up "destroyed" trails, i.e. those for which

collision circumstances make synapse formation impossible, in which case no CA rules are defined, so by default the background state (black, zero) becomes the next state, which can destroy the trail. Usually the network stabilizes after several hundred clock cycles, i.e. all signal sequences get absorbed at synapses. Once this happens, a second set of CA rules gets switched on which makes the CA network behave like a neural network. For example, there are three kinds of sheath cells, two for "axons" (excitatory and inhibitory) and one for "dendrites". Signal strengths in axons keep the same value they had at emission (at CA neurons), but once the axon signal passes through an axon-to-dendrite (A-->D) synapse (created in the CA net growth phase), it becomes a dendrite signal, which drops off in strength as it advances. Thus the dendrite signal strength depends on its distance from its (A-->D) synapse. Since these distances are evolvable, they are equivalent to the weights of conventional neural networks. Signal values can be positive or negative. At an excitatory synapse, the sign of the axon signal value is transmitted unchanged to the dendrite. At an inhibitory synapse, the sign of the axon signal value transmitted to the dendrite is reversed. Excitatory and inhibitory axons generate excitatory (+ve A-->D) and inhibitory (-ve A-->D) synapses when dendrite CA trails collide with them. Axon-axon (A-->A), dendrite-dendrite (D-->D) and dendrite-axon (D-->A) synapses are simply not formed. Two merging dendrite signals add their incoming signal strengths at the junction. CA rules can be defined which allow this. When a dendrite CA trail splits, special "gating" cells are formed at the split

junction, which are later used (when the CA trails behave like neural nets) to direct the dendrite signals to turn towards the neuron which grew them. Finally, the axon output signal strength at a CA neuron can be a non linear function of the sum of its incoming dendritic signal strengths. The strength of this axon signal remains unchanged as it travels through the axon.

4. State Transition Details

This section provides a more detailed account of some of the state transitions used to grow the cellular automata trails. A complete description of the more than 4000 hand crafted rules would require a treatise of book length, so only a brief selection will be given here. **Figs. 4-7** show the state transitions of the cells at the tip of a trail for the "extend", "turn left", "split left", and "T split" instructions. The equivalent "turn right" and "split right" cases can be obtained from mirror symmetry. **Fig. 8** shows how a synapse is formed when a dendrite collides with an axon. The state transitions shown below are really only a sample of many.

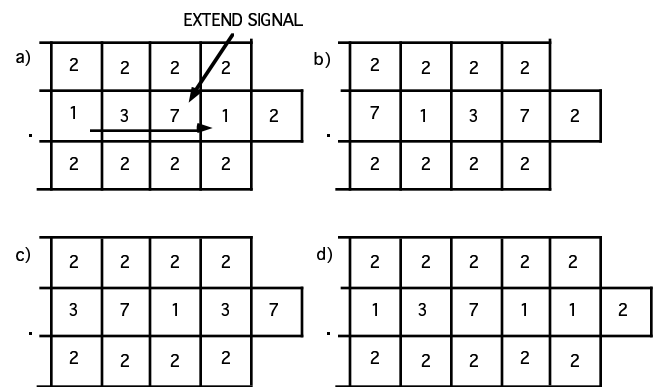


Fig. 4 Extend the Trail One Square

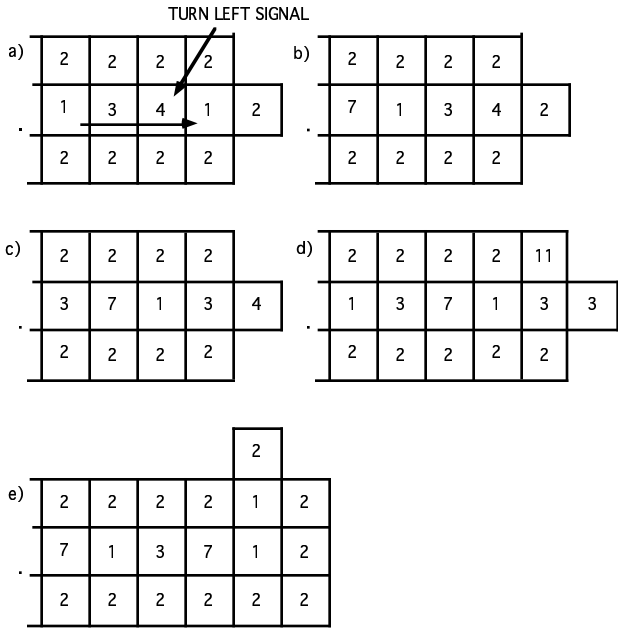


Fig. 5 Trail Turns Left

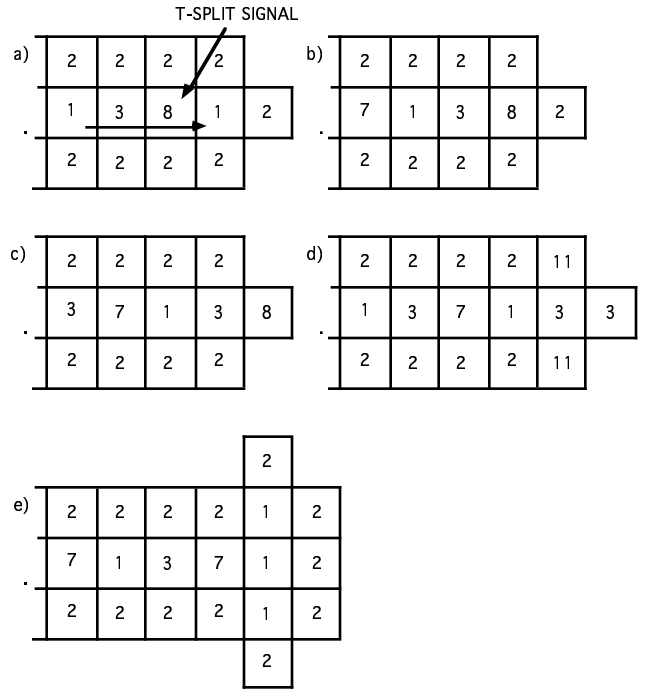


Fig. 7 Trail T-Splits

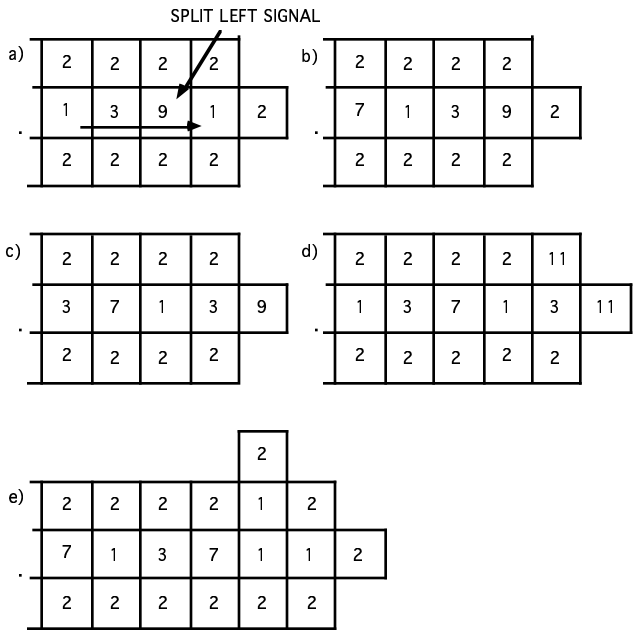


Fig. 6 Trail Splits Left

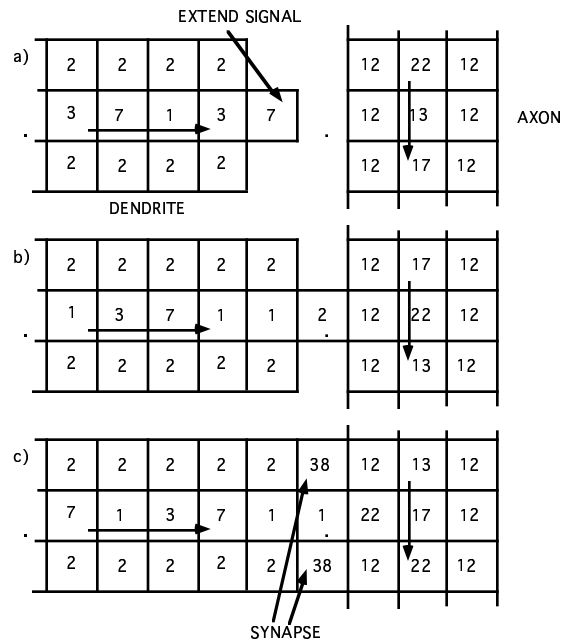


Fig. 8 Dendrite to Axon Synapsing

5. Future Research

A lot of work remains to be done. At the time of writing, the 2D version of a software simulation of CAM-Brain is nearing completion (having added nearly 4000 handcrafted CA state transition rules). Initial results are shown in **Figs. 9-11**. Unfortunately, on a Sun Sparcstation 10, the time necessary to grow a 16 neuron module is too long to be practical. Therefore it will be necessary to transfer the program to ATR's CM5 supercomputer. Another immediate problem was an insufficient connectivity between the neurons. It was found that on average, there were roughly two dendrite-to-axon collisions per neuron. This is probably not enough to permit evolvability. Hence the next immediate step will be to double the connectivity by permitting synapses to be formed at axon-to-dendrite collisions. Implementing such synapses will be more complex than those occurring at dendrite-to-axon collisions, because one needs to specify the direction that signals need to turn at the synapse in order to reach the neuron. Once the connectivity has been at least doubled, initial experiments will be carried out on a 4 neuron module on the Sparc10 to see how well it evolves. The author already has several years of experience with evolving simulated fully connected artificial neural modules (e.g. with 16 neurons) and found them to be highly evolvable [2]. Even if one cut 70% of their evolved weights (i.e. one reduced their value to zero), the evolved function of the module remained more or less intact. Therefore it is expected that CAM-Brain's neural circuits should still be evolvable despite the lower connectivity. If

the 2D version shows good evolvability (i.e. if the CA based neural net modules evolve successfully to perform desired functions) then the next step will be to try a 3D version. Since collisions occur more easily in 2D, a 3D version will be qualitatively different. To help conceive the 3D CA state transition rules (using a 6 neighborhood transition rule, i.e. "up, down, N, E, S, W"), the author uses sponge cubes stuck together with matches. A Silicon Graphics machine will be used to help visualize the cubes of the 3D cellular automata space. If the 3D version also proves to be evolvable, then in 1994, work will begin in earnest on a design of a Cellular Automata Machine adapted to CAM-Brain. Contact has already been made with the CAM building group at MIT [7] to buy one of their recent CAM8 machines. The architecture of this machine will be adapted to be suitable for CAM-Brain. It is also possible that the MIT post doc who built CAM8 may pay a 3-6 month working visit in the near future to help design the CAM-Brain hardware.

The beauty of using CAs as the basis for the CAM-Brain Project, is that they allow an initial growth of a structure which can later be used. Thus, in effect, one has a type of "evolvable hardware" (EHW) [8]. The author sees two broad categories of evolvable hardware approaches, which he labels "intrinsic" and "extrinsic". In "intrinsic EHW", the evolution occurs inside (intrinsic) the "hardware" itself, e.g. FPGAs (field programmable gate arrays, or other kinds of programmable logic devices (PLDs)) can have their hardware configuring bitstrings be conceived as

chromosomes in a Genetic Algorithm, so that one obtains a new hardware circuit for each chromosome, for each generation of the GA algorithm. In a CAM, the underlying hardware does not change, so strictly speaking, one is not doing EHW, but since a circuit gets evolved for each chromosome (where a chromosome is the sequence of signal cells which move down the middle of the CA trails), the process is equivalent to EHW. With "extrinsic EHW", one uses software to simulate the evolution of a hardware circuit, e.g. by evolving a high level symbolic circuit description (e.g. using a HDL (hardware description language)), and then writing (down loading) the elite chromosome's solution into the configurable hardware. Hence the configurable circuit is written to just once. The real evolution occurs outside (extrinsic) the hardware. The CAM-Brain Project can thus be looked upon as a type of intrinsic EHW.

Another feature of CAM-Brain, is that it will be possible to grow connections incrementally between one neurite module (i.e. a GenNet = Genetically Programmed Neural Network [1]) and another, and thus build/evolve/GP an artificial brain with thousands of GenNets. This would be a kind of incremental evolution. Since most papers on neural networks are concerned with only a single neural module, to be dealing with thousands of modules, as may be the case with the CAM-Brain Project, will be a breakthrough in neural network research. If successful, it will create a tool within which to build artificial brains. Initially, these artificial brains will contain only a "small" number

of modules, e.g. 100s, later 1000s etc. Gradually, as the number of modules increases, and insight is obtained as to how to evolve them incrementally, building artificial brains of increasing "intelligence" will become possible, thus creating a bridge between the two fields of Artificial Life (ALife) (whose long term goal is to create artificial life forms) and the more distant goal of Artificial Intelligence (AI) (whose long term goal is to create artificial intelligences). A sufficiently intelligent artificial life form becomes an artificial intelligence.

Commercially speaking, if electronic circuits can be successfully grown/evolved, rather than be humanly designed, it may be possible to create circuits of enormous complexity, and hopefully, superior functionality. If so, the notion of "evolvable hardware" may revolutionize the electronics industry.

By the year 2001, the last year of the CAM-Brain Project, our Brain Builder Group hopes to have the capacity to put a million neural net modules (GenNets) into a trillion cell Cellular Automata Machine. During the 1990s, our group also intends to design molecular scale CAMs and similar machines, so that the number of neural modules which can be evolved can jump from millions to Avogadro's number (i.e. a trillion trillion), once nanotechnology becomes a reality in the early 2000s [9].

References

- 1) Hugo de Garis, "Genetic Programming : Modular Evolution for Darwin Machines", IJCNN-90-WASH-DC, (Int. Joint Conf. on Neural Networks), January 1990, Washington DC, USA.
- 2) Hugo de Garis, "Genetic Programming", Ch.8 in book "Neural and Intelligent Systems Integration", ed. Branko Soucek, WILEY, 1991.
- 3) Hugo de Garis, "Artificial Embryology : The Genetic Programming of an Artificial Embryo", Ch.14 in book "Dynamic, Genetic, and Chaotic Programming", ed. Branko Soucek and the IRIS Group, WILEY, 1992.
- 4) Hugo de Garis, "Genetic Programming : GenNets, Artificial Nervous Systems, Artificial Embryos", WILEY manuscript.
- 5) E.F. Codd, "Cellular Automata", Academic Press, 1968.
- 6) D.E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.
- 7) T. Toffoli & N. Margolus, "Cellular Automata Machines", MIT Press, 1987.
- 8) Hugo de Garis, "Evolvable Hardware : Genetic Programming of a Darwin Machine", in "Artificial Neural Nets and Genetic Algorithms", R.F. Albrecht, C.R. Reeves, N.C. Steele (eds.), Springer-Verlag, 1993.
- 9) K.E. Drexler, "Nanosystems : Molecular Machinery, Manufacturing and Computation", Wiley, 1992.

Dr. Hugo de Garis

Hugo de Garis is an invited researcher of the Evolutionary Systems Department at ATR, in Kyoto. He obtained his PhD in the field of Genetic Programming (i.e. using evolutionary algorithms to build complex systems) from Brussels University, Belgium, in January 1992, and was an STA postdoctoral fellow at the Electrotechnical Lab (ETL) in Tsukuba in 1992. Dr. de Garis is now trying to grow/evolve an artificial brain inside a cellular automata machine.

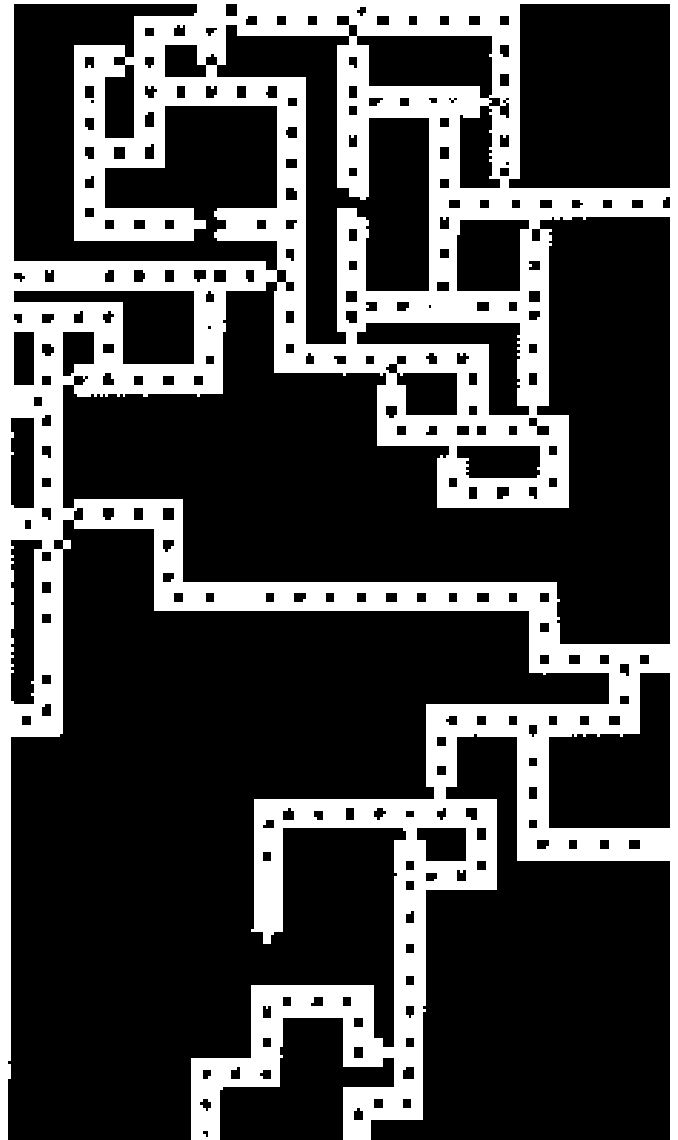


Fig. 9 Cellular Automata Trails

Fig. 10 Early Growth of a CA 16-Neuron Net

Fig. 11 Later Growth of a CA 16-Neuron Net