A Mathematical and Physical Base for 'A Standard Model of the Mind'

Emmett Redd, A. Steven Younger
Physics, Astronomy, and Materials Science, Missouri State University, 901 S National Ave., Springfield, MO 65897
EmmettRedd@missouristate.edu, SteveYounger@missouristate.edu

Abstract

This paper describes a mathematical and physical base for 'A Standard Model of the Mind'. It is a super-Turing model constrained by physical reality of the brain's construction. The constraints are noise and quantized charge transfer. The model has computing power beyond the Universal Turing Machine (UTM), which Turing himself claimed to be insufficient to model the brain. The super-Turing model meets Turing's desire for a more complex model. Allen Newell expressed difficulties in modelling the brain with the UTM not being as computationally complex as his functional analysis indicated. We will describe the model and note neuron operations compatible with it. Because of mathematical idealities in both Turing and super-Turing models, physical devices cannot directly implement either model. However, both Turing machines and super-Turing models can point the direction to the design and operation of physical devices. Brain modeling should use the more powerful super-Turing model to describe its operation. Our research seeks to develop artificial neural networks based on this model. The super-Turing model is guiding analog- and digital-hybrid hardware development of these neural networks. We will describe the progress on an optical implementation with encouraging, chaos-mimicking results and on designs for an electronic implementation. We describe a possible spectrum of super-Turing inspired devices. We will call on the community to help further the devices and their use in 'A Standard Model of the Mind'.

I. Introduction

As mathematics formalizes the Standard Model of physics, 'A Standard Model of the Mind' (SMM) should also have mathematics and physics at its foundation. Brain modelers have long suggested the Universal Turing Machine (UTM) as the mathematical model for the brain. Turing himself claimed it was insufficient and described features needed in a more complex computation model (Siegelmann 2013). Newell was also aware of difficulties with the UTM modelling the brain since it could not "compute an uncountable number of functions--…"(Newell 1990) (Section II below).

Siegelmann and others studied the computational complexity of Analog Recurrent Neural Networks (ARNNs) in the mid-1990s (Siegelmann 1993, 1994, 1998). Three models (or families thereof) resulted in super-Turing computational complexity. One has axioms (stochastic signals and rational weights) similar to properties of the physical universe and is super-Turing. We will describe the models in Section III.

In the incipient SMM (Laird 2017), the *deliberate act* level is one to two levels above the *neuron* and *neural circuit* levels (Newell 1990) addressed by this paper. Since this is the case, we will spend most of our effort in contributing details to those two levels of Newell's *Unified Theories of Cognition* rather than Laird's incipient SMM. However, since a level's operation is independent of those above or below, inserting these ideas into an overall SMM at the levels below the *deliberate act* level are all that is necessary to affect the whole SMM. It is at these levels that the physical constraints we discuss are most crucial. We will discuss the physical constraints and neuron operations that lead us to pick one of the super-Turing ARNN models as the best description for computation for the brain. Since it appears sufficient to explain the complexity of brain computations, we here argue that appeals to quantum coherence effects (Penrose 1994) in the macromolecular band below the biological band (Newell 1990) are not required.

Section V will discuss the analog- and digital-hybrid optical system we have used to test super-Turing operation by mimicking chaos. We will also discuss an accelerator board design using analog electronics implementing the super-Turing model. Its use and placement will be like a Graphic Processing Unit.

In Section VI, we will discuss a spectrum of practical ways to implement neural simulations inspired by the super-Turing model. It will also discuss the future directions of our (and we hope) the SMM community's work looking at various ways to use super-Turing inspired tools.

II. Computation and the Brain

The following subsections will summarize the computation and brain discussions of both Turing and Newell.

Turing and the Brain

Here is a condensation of 'Turing on Super-Turing and Adaptivity' (Siegelmann 2013) specifically addressing brain computation issues. The article also discusses the application of super-Turing computation theory to a much wider array of biological processes. The point here is not to describe Turing's ideas about the brain for insertion into the SMM, but rather, to point out the insufficiencies of the UTM in describing brain functions.

Over the span of about a dozen years, Turing described a computational machine stronger than his UTM. The goal was to describe a 100-year-in-the-future machine, which would stand a chance in winning the Imitation Game (Turing 1952). Near the end of these dozen years, he elucidated why the UTM was not up to the task (Turing 1950):

> Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner.

Turing's multi-year description included four (sometimes-overlapping) features that a stronger machine would be or have, i.e., 1. A Series of Machines, 2. Learning and Adaptivity, 3. Randomness, and 4. Rich Information. The super-Turing model also has these four features (Siegelmann 2013). Some of the features align with the constraints of the physical world. Section IV will discuss those constraints in detail.

In the super-Turing model, a Series of Machines results from 'the combined process of learning and computation'. Learning and Adaptivity may be thought of as a given within any neural network model. However, the super-Turing model described has a higher speed and accuracy than the UTM in implementing learning. Randomness is inherent in the analog signals in the ARNN model; UTMs eschew randomness. Since real numbers are exponentially richer in number than the natural numbers, the super-Turing model, by using or approximating the use of real numbers, has exponentially richer information space than a UTM.

These latter two seem to be the most important, with randomness being the most subtle. Information richness will be discussed in Section III; randomness' subtleties will be discussed here.

In digital computer applications, Pseudo Random Number Generators (PRNGs) are used (Monte Carlo modelling, Deep Learning (Neelakantan 2015), etc.) but fail at many tasks, i.e. cryptography, slot machines (Koerner 2017), etc. Reasons for their failure include determinism and repeated sequences. Determinism is expressly not random. Some tasks will not work with some PRNGs (Jones 2010). Properly seeding the PRNGs needs real random numbers. Super-Turing ARNNs likely need properly tuned randomness.

Newell and Computation

Newell (1990) gives N^M for the number of functions in a computational system where there are M inputs and N possible outputs for each input. He illustrates the rapid growth of this number and references a theorem where countably unbounded inputs and outputs results in an uncountably infinite number of functions (Minsky 1967). He then says, "This result shows the futility of the hope for computers that provide *all* possible functions--…" and goes on to discuss computing classes. At the time, the UTM was from the strongest known computation class (Newell 1990). He saw it not having an uncountably infinite state space as a difficulty in its use to model the brain. However, there were no alternatives. We now have an alternative that allows computation of an exponentially greater number of functions and removes the Newell-noted difficulty. Below, we will describe a computation class, which has an uncountably infinite state space and is physically realizable. It can model the brain.

III. Super-Turing Analog Recurrent Neural Networks

Study of the computational power of ARNNs started in the early 1990s (Siegelmann 2013). The expectation was that they would be no more powerful than a UTM. After several unsuccessful attempts proving this, the goal changed to proving they were more powerful. This met with success (Siegelmann 1993, 1994).

The research culminated with a book (Siegelmann 1998) describing the computing power of ARNNs for a comprehensive set of features. In summary, the features were constraints on the synaptic weights and on the signals. The weights were constrained to be integer, rational, or real. The signals were constrained to be deterministic or stochastic. Table I shows a matrix of computing powers because of these constraints. It also summarizes which synaptic weights are physically realizable.

In the second column (Deterministic Activations) of Table 1, the 'regular' computation power is the same as the one of finite automata. The state space for it is finite. The 'P' computation power is the same as the one of the UTM. The state space for it is countably unbounded, designated as $\aleph_0$. The 'P/poly' computational power results from polynomial advice given to a UTM. P/poly has super-Turing computational power and a state space the same size as real numbers, i.e. uncountably infinite, designated as $2^{\aleph_0}$. Unfortunately, the quantized nature of the universe mitigates against weights having real values. Therefore, real weights are not physically realizable as noted in the lowest right table entry. Additionally, there are no physically realizable super-Turing computation powers in the second column.

Table 1. Computing Power of ARNNs. Given the constraints on synaptic weights and neuron activations (signals) different computational powers result. The physical universe also has its constraints of quantized charge putting further limits on what computational powers are possible. See text for power descriptions.

Synaptic Weights	Deterministic Activations	Stochastic Activations	Physically Realizable
Integer	regular	regular	Yes
Rational	P	BPP/log*	Yes
Real	P/poly	P/poly	No

The remaining computation power is BPP/log*. It requires rational weights and stochastic activations. It has super-Turing computation power. It also has an uncountably infinite state space (Cabessa 2014). Fortunately, it is physically realizable for two reasons. Collections of quantized particles provide rational weights. Inherently (thermal, etc.) noisy signals provide stochastic activations. It is super-Turing and has the state space of real numbers because it is a probabilistic (coin flipping) Turing machine that uses binary coins having *real* probabilities. "…a long sequence of coin flips allows indirect access to the real valued probability…" (Siegelmann 1998).

IV. Physical Constraints

Two main physically realizable features discriminating between the super-Turing models described above are stochastic signals and rational weights. From the discussion above, these are the minimum requirements. The BPP/log* model family has these features and indicates that a physically realizable super-Turing computation model exists. This existence is crucial and these two features can help guide their implementation in the physical world.

Without a doubt, digital computers can provide rational weights. (We will use that fact in our mixed-signal neural network design in Section V.) In fact, the quantum nature of charge argues against real-numbered weights or signals. A neuron's ion channels can each only admit quantized packets of charge. This causes the cell potential to increase in (albeit small) steps until a threshold is reached. This is a rational number process because ions passing through cause the weight inherent in a synaptic process to be some fraction of the ions needed to exceed the threshold. This means that the neuron's synaptic processes can implement the rational-number-weight axiom of the super-Turing model. Therefore, both the brain and other physically realizable devices can perform at a level that satisfies Turing's own comments above and quoted in (Siegelmann 2013).

Newell explicitly says that the neuron pulse codings "are all statistics on the pulse sequences as a stochastic process." (Newell 1990) Some mathematicians may complain that random physical processes (noise in signals) are not truly stochastic. However, they are the nearest possible physical manifestation of stochasticity. While there are several physical processes that underlie this random noise, the thermal voltage noise of 25 mV is clearly sufficient. Contrary to this, digital systems have invalid signal voltage ranges so noise does not contaminate its signals.

The discussion above shows that *neural circuits* can perform super-Turing operations while digital computers cannot perform all operations necessary for this super-Turing model. Therefore, a digital system is not optimum for simulating neural operations.

V. Super-Turing Inspired Physical Neural Networks

We were researching Fixed-Weight Learning (FWL) (Younger 1999) on optical hardware (Younger 2009) when we became aware of super-Turing ARNNs (Siegelmann 1998). Since FWL is recurrent and we were combining the optical signals in an analog fashion, we referred to Siegelmann as a justification for our research. When the opportunity arose, we informed Siegelmann of our research. That led to a collaboration between the three of us that ultimately resulted in the combined understanding of this position paper. It also led to design of a Mixed-Signal Neural Network better implementing the super-Turing theory in a physical device (Younger 2014). With the existing optical system, we have shown computation consistent with super-Turing operation (Younger 2017). The optical and mixed-signal systems will be discussed separately below.

Optical Neural Network Mimics Chaos

While a super-Turing machine should be able to compute an exponentially larger number of functions, the only definitive 'test' for super-Turing operation that we know is mimicking chaos (Siegelmann 1995). We would entertain suggestions for other tests to verify super-Turing operation or its improved performance in any specific domain.

Unfortunately, with the tests for chaos, one cannot prove that a system or device is chaotic; one can only determine whether it is consistent with chaos or not (Kaplan 1995). A reason for this is that the next output from a supposed-chaotic system could start a sequence which replicates an earlier portion of the sequence. Another way of stating this is one must take an infinite sequence to make sure the outputs never start repeating. A similar idea to this is that it takes an infinite number of outputs from a device to prove it is super-Turing (Costa 2017). So, whether chaotic or super-Turing, one can only say, "The (process, device) is consistent with (chaos, being super-Turing)."

The optical system shown in Fig. 1 was consistent with chaos in a 10000-point time series (Younger 2017). The paper also shows how Digital Recurrent Neural Networks (DRNNs) running on a conventional computer failed at mimicking chaos. One DRNN had precision of 9-bits for comparison to the approximate precision of the optical system. This appears to be a low level of precision, but even a purely digital neural network can work at these low levels (Höhfeld 1992). The other had approximately twice the number of bits of precision. (Comparing to the full 64-bit precision of the digital computer calculating the DRNNs appeared to be an unacceptable apples-to-oranges test.) The two DRNNs failed to indicate the chaos of the Logistic function in two tests: Autocorrelation and Largest Lyapunov Exponent.

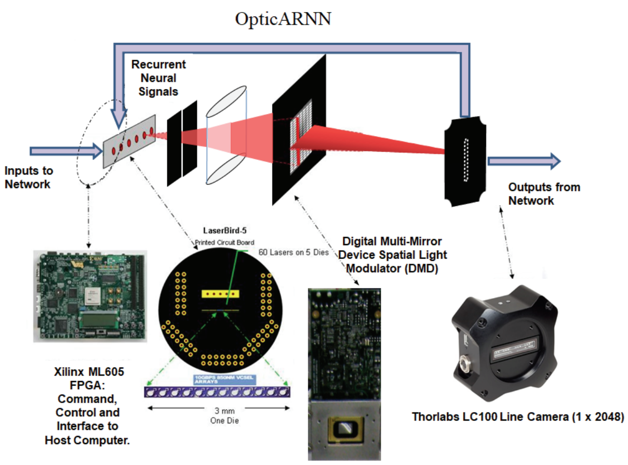

Figure 1. The OpticARNN uses optical signals in red to perform rational synaptic weight multiplications via the DMD and analog neural summations in the line camera. Digital signals are in blue.

The Autocorrelation Test rules out chaos by showing that a finite number of unique points successively repeats in the time series. In the time series resulting from the 9- and 18-bit DRNNs, there were, respectively, only 22 and 109 unique points. Therefore, the DRNNs did not exhibit chaos. The optical neural network did not repeat in its time series. Its results are consistent with chaos since chaotic systems do not repeat.

In the Largest Lyapunov Exponent Test, a positive exponent indicates the trajectories starting at two nearby points diverge; negative exponents indicate a convergence of trajectories. It is a mathematical measure of the *sensitive dependence on initial conditions* feature of chaos. Another name for this feature of chaos is *the butterfly effect*. The DRNNs both failed to have any positive exponents. The optical system had a positive exponent of 1.083 (actually indicating more divergence than even the training data for the three neural networks trained and tested).

In both tests, the optical neural network showed consistency with chaos while neither of the DRNNs did. Future work may insert comparable precision pseudo- and physical-randomness to test whether the modified DRNNs can mimic chaos. Further discussions are below.

Mixed-Signal Neural Network

Figure 2 shows a conceptual diagram for a mixed-signal neural network. The intent is to build an Analog Neural Network Accelerator (ANNA) board implementing the portion of the model illustrated in black for insertion on a computer bus.

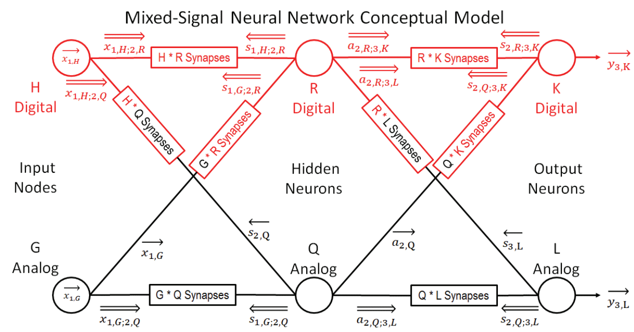

Figure 2. Mixed-Signal Feed-Forward Neural Network. Red is the digital side and Black is the analog side. Right pointing arrowhead indicates fan out; left pointing one indicates fan in. Recurrent connections are externally connected.

The super-Turing features of rational weights and stochastic signals guided its design. The computer operating the red side to the neural network would provide rational weights to the analog side. This board would keep some signals and their inherent noise solely in the analog domain. Using Commercial-Off-The-Shelf components, ANNA could have up to 24 neurons and 144 synapses. At this point, the design is only a multi-layer perceptron. Other connection schemes, even dynamic ones could be had, but the additional required components would take space, causing removal of some synapses and neurons. Successive designs would seek to incorporate those features. Connectors would allow signal connection to other boards for deeper networks or back to inputs for recurrent networks.

VI. Super-Turing Inspired Device Spectrum

Newell (1990) noted that the most general machine in a computation class could "realize *any* function realized by *any* other machine in the class." This is a *universal machine* of the class. Any non-universal machine of the class can compute some, but not all functions of the class. We have presented one device, short cutting some of the BPP/log* super-Turing axioms, whose output is consistent with chaos and the BPP/log* computation class. We have presented one design implementing the super-Turing model guidance more fully.

The two devices in Section V reflect that there is a super-Turing hierarchy of computing classes (Siegelmann 1998) with these two devices being on a spectrum of super-Turing inspired devices capable of computing different sets of functions. They also indicate that strict adherence to the BPP/log* axioms may not be required. The spectrum may reflect various levels of adherence to the axioms. As discussed below, some may not have super-Turing power because their digital portions strip too much of the randomness away. In addition, any suggested randomness generator needs critical analysis to determine its suitability for any particular task.

Here is a list from our research efforts of provisional entities on that spectrum (in speculated order from lowest to highest power):

1. **Digital computer with Turing's radium** (Turing 1951)
2. **9-bit DRNN using 9-bit PRNG**
3. **9-bit DRNN using RdRand system call**
4. **18-bit DRNN using 18-bit PRNG**
5. **18-bit DRNN using RdRand system call**
6. **64-bit Phoneme Recognition NN using MATLAB's rand()**
7. **64-bit Phoneme Recognition NN using RdRand system call**
8. **OpticARNN**
9. **Mixed-Signal Neural Network**

Since radium might not overcome the designed-for-noise-immunity digital computer, the first entry likely falls off the list as remaining a Turing machine. To get radioactivity generated random numbers to work is more complicated than Turing's idea of just exposing the computer to radium.

Because of the results detailed above, the second and fourth entries maybe should not be included on the spectrum since PRNGs repeat. However, tests are possible on these entries.

Since about 1999, many Intel based computers have had an entropy-based random number generator and other chipmakers included one later. A system call (**RdRand**) accesses it. While there are questions about its security for encryption, it can add true randomness to a computer's calculation. Whether this is sufficient for a computer to mimic chaos, we can test whether the modified DRNNs do.

The tests using Phoneme Recognition in entries 6 and 7 will not determine whether they are or are not on the super-Turing device spectrum. Rather, they are tests seeking to show in a practical way that super-Turing guided devices have improved performance over conventional computation. Other researchers could use the listed (P)RNG functions to perform to perform similar tests on their brain simulations or neural networks.

Further testing with OpticARNN also seeks for a practical way to show improved performance.

Progress on ANNA will be slow until a funding source can be convinced it is worth the investment. Should the tests above show improved performance, they may provide reasons not to build the ANNA boards. That is, ANNA's cost might not be justified by the expected improvement in computing power. However, ANNA boards (or others similarly designed) should be the nearest to implementing the super-Turing model since they adhere most closely to the BPP/log* axioms. Its rational weights only have powers of two in the denominators. Since the theory uses all rational numbers in its weights, ANNA is not likely the *universal machine* for the BPP/log* computation class. Similarly, we do not know whether the brain's variety of computable functions uses a more significant fraction of the rational number set and, thereby, needs to be modelled by the *universal* BPP/log* *machine*.

We actually encourage others to join in this testing. Its results will be useful to the SMM community.

VII. Conclusion

Our call for action is the same as presented in the quote below (Siegelmann 2013).

> If we limit our research to Turing computation only, we may fail in our efforts to create true Artificial Intelligence and likewise we may misunderstand the mechanisms upon which life is based.

At this point, SMM simulations are restricted to digital computers inspired by the Universal Turing Machine. This paper argues that super-Turing inspired computations would be better for calculating the SMM. We realize specific tools inspired by the super-Turing model do not yet exist for adoption within the SMM community. While we continue to develop mixed-signal neural networks wherein some signals remain always analog, the optical experiment may hint at super-Turing consistent operation with fewer analog (i.e. random) operations. We listed several other systems accessible in our research group which might be made to exhibit chaos. We are continuing to investigate those as well. Two list entries provide our example experiments, which use ubiquitous computer hardware available to other researchers, to prompt them to test their systems for improved performance themselves. When we or when the SMM community helps make super-Turing inspired computation devices available, the community will make significant progress on the underlying mathematics and physics of A Standard Model of the Mind.

References

Cabessa, J. 2014. Personal communication.

Costa, J.F. 2017. Personal communication.

Höhfeld, M. and Fahlman, S.E., 1992. Probabilistic rounding in neural network learning with limited precision. Neurocomputing, 4(6), pp.291-299.

Jones, D. 2010. Good Practice in (Pseudo) Random Number Generation for Bioinformatics Applications. Accessed 8-10-17. http://www0.cs.ucl.ac.uk/staff/d.jones/GoodPracticeRNG.pdf

Kaplan, D. and Glass, L. 1995. Understanding Nonlinear Dynamics. New York: Springer-Verlag.

Koerner, B. 2017. Russians Engineer a Brilliant Slot Machine Cheat—And Casinos Have No Fix. Accessed 8-10-17. https://www.wired.com/2017/02/russians-engineer-brilliant-slot-machine-cheat-casinos-no-fix/

Laird, J.E., Lebiere, C. & Rosenbloom, P.S. 2017. A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. AI Magazine.

Minsky, M.L. 1967. Computation: finite and infinite machines. Prentice-Hall, Inc.

Neelakantan, A. 2015. Adding Gradient Noise Improves Learning for Very Deep Networks. arXiv:1511.06807 ("seeds" imply PRNG).

Newell, A. 1990. Unified Theories of Cognition. Cambridge, MA: Harvard Univ. Press.

Penrose, R. 1994. Mechanisms, microtubules and the mind. Journal of Consciousness Studies, 1(2), pp. 241-249.

Siegelmann, H.T. 1993. Foundation of Recurrent Neural Networks. Rutgers University, Ph.D. Thesis.

Siegelmann, H.T. & Sontag, E.D. 1994. Analog Computation via Neural Networks. Theoretical Computer Science 131, pp. 331-360.

Siegelmann, H.T. 1996. The simple dynamics of super Turing theories. Theoretical Computer Science 168.

Siegelmann, H.T. 1998. Neural Networks and Analog Computation: Beyond the Turing Limit. Boston, Mass.: Birkhauser.

Siegelmann, H.T. 2013. Turing on Super-Turing and Adaptivity. Progress in biophysics and molecular biology, 113(1).

Turing, A.M. 1950. Programmers' Handbook for Manchester Electronic Computer, University of Manchester Computing Laboratory. www.alanturing.net/programmers_handbook/ .

Turing, A.M. 1951. Can Digital Computers Think? BBC May 15, 1951.

Turing, A. M., Braithwaite, R., Jefferson, G., and Newman, M. 1952. Can Automatic Calculating Machines Be Said To Think? BBC January 10, 1952.

Younger, A.S., Conwell, P.R. and Cotter, N.E., 1999. Fixed-weight on-line learning. *IEEE Transactions on Neural Networks*, *10*(2), pp.272-283.

Younger, A.S. and Redd, E., 2009, June. Fixed-weight learning neural networks on optical hardware. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on* (pp. 3457-3463). IEEE.

Younger, A.S., Redd, E. and Siegelmann, H., 2014, July. Development of physical super-Turing analog hardware. In *International Conference on Unconventional Computation and Natural Computation* (pp. 379-391). Springer, Cham.

Younger, A.S., Redd, E., Siegelmann, H., and Bell, C. 2017. A Physical Machine Based on a Super-Turing Computational Model. UCNC 2017 Unconventional Computing and Natural Computing. https://ucnc2017.csce.uark.edu/wp-content/uploads/sites/176/2017/06/PC-2017-Physical-Machine.pdf

Acknowledgements

We thank Hava Siegelmann, first, for proving the super-Turing theories and, second, for jumping at the chance in becoming our colleague and working with us to physically realize her theories.

This work was supported in part by U.S. National Science Foundation Grant ECCS-1201790.