



OSTP Nanotechnology-Inspired Grand Challenge: Sensible Machines (extended version 2.5)

R. Stanley Williams
Hewlett-Packard Laboratories

Erik P. DeBenedictis
Sandia National Laboratories

October 20, 2015

<p>Committee (also reviewers):</p> <p>Thomas M. Conte, IEEE and Georgia Tech Paolo A. Gargini, ITRS 2.0 David J. Mountain, IEEE and LPS Elie K. Track, IEEE and nVizix</p>	<p>Reviewers and team members</p> <p>IEEE Rebooting Computing: Arvind Kumar, IBM, Mark Stalzer, Caltech</p> <p>ITRS 2.0: Mustafa Badaroglu, Qualcomm, Geoff W. Burr, IBM, An Chen, Globalfoundaries, Shamik Das, MITRE, Andrew B. Kahng, UCSD, Matt Marinella, Sandia</p> <p>Sandia: Sapan Agarwal, John B. Aidun, Frances S. Chance, Michael P. Frank, Conrad D. James, Fred Rothganger, John S. Wagner</p> <p>SRC: Ralph Cavin, Victor Zhirnov</p>
--	--

Sandia review and approval completed: tracking Number: 342796 Unclassified, unlimited release.

OSTP Nanotechnology-Inspired Grand Challenge

Above and Beyond Exascale Computing: Sensible Machines

1. Introduction

This white paper is a response to the Request for Information (RFI) issued by the White House Office of Science and Technology Policy (OSTP) seeking suggestions for *Nanotechnology-Inspired Grand Challenges for the Next Decade*. We describe the ambitious but achievable goal of building ‘Sensible Machines’ that can solve problems that cannot be solved by any computing machine that exists today, and find solutions to very difficult problems in a time and with an energy expenditure many orders of magnitude lower than achievable by today’s information technology. The program will require collaboration among researchers from a broad spectrum of disciplines, including nanoscience, computer science and engineering, neurophysiology, applied mathematics, quantitative psychology, and electrical engineering. We describe a computational system that would both utilize and coordinate the advances suggested in two examples described in the RFI: *devices no bigger than a grain of rice that can sense, compute, and communicate without wires or maintenance for 10 years, enabling an “internet of things” revolution and computer chips that are 100 times faster yet consume less power*. We view these two advances as inevitable, and assume that they will occur by 2025 even without any special attention from OSTP because the commercial sector is already mobilized and moving in these directions. What is not being addressed commercially in the US, and what is urgently needed, is a completely different paradigm for understanding how to arrange and sort through petabytes and more of data to find correlations and answers to questions that we do not even know how to ask, or in other words, Sensible Machines. The central thesis of this white paper is that although our present understanding of brains is limited, we know enough now to design and build circuits that can accelerate certain computational tasks; and as we learn more about how brains communicate and process information, we will be able to harness that understanding to create a new exponential growth path for computing technology. Just as modern airplanes do not use feathers and flapping wings like the birds that inspired their development, Sensible Machines may learn from brains how to process more information per energy expended even though they will utilize different signals and dynamical networks.

Exascale computing enabled by nanotechnology

In terms of measurable improvement over time, computing has been by far the most successful technology of the past century. Information technology influences almost every aspect of our lives and has been the largest driver of U. S. economic growth for at least three decades. Numerous plots of computing efficiency, e.g. energy expended per bit operation vs. date achieved, have shown a fairly continuous exponential improvement of twelve orders of magnitude from the first electronic computers of the 1940s to today’s large data centers. Since the introduction of the silicon integrated circuit in the 1960’s, most of that efficiency improvement has come from continuously scaling down the size of the transistors on a chip, i.e. Moore’s Law. In 2015, companies are now shipping products that utilize the 14 nm node for chip manufacturing described by the International Technology Roadmap for

Semiconductors (ITRS). The expert consensus is that integrated circuits will continue to operate with transistors down to a minimum feature size of 5 nm, but manufacturing reliable circuits with those dimensions will require heroic efforts and investments by the industry. Thus, the end of traditional Moore's Law scaling is within sight (really!). However, this does not mean the end of exponential improvements in computing energy efficiency or speed, but rather a great opportunity to reinvent information science and technology. During the height of the Moore's Law era, the entire computing industry and supply chain were so focused on harnessing the improvements available from scaling CMOS technology that many other possibilities for improving and/or expanding computing were ignored while the industry raced to the minimum transistor size. In fact, data processing has not been the bottleneck for computer system performance for the past decade – orders of magnitude more time and energy are expended in moving bits around the memory and storage subsystems of a computer than in performing logic operations. Thus, many regard the end of feature size reduction as the incentive to re-examine, refine and rebalance computing machines, which would not only dramatically improve performance on traditional forms of computing but may also enable entirely new types of problems to be solved and science to emerge.

The basic architecture of computers today is still the same as those built in the 1940's – the von Neumann architecture – with separate compute, high speed memory and (slow) high density storage components that are connected together (mainly) electronically. The recent rise of solid-state nonvolatile memory and storage devices provides the opportunity to collapse the traditional computer data hierarchy and to store with immediate availability all the data required for computation directly adjacent to (or even mixed within) the processors. Advances in integrated silicon nanophotonics will allow data to be transmitted over any distance larger than a few millimeters on chips, from chip to chip, and beyond. The resulting decrease in latency and energy expended and increase in bandwidth for data communication between storage and processor may provide three orders of magnitude improvement in system computing efficiency by 2025 even without any further improvements in processor technology, as long as software evolves to take advantage of the hardware improvements. Thus, the present trend for total computer performance improvement should continue to the point where a 50 exaflops computer that runs within a 20 MW power envelope could be at the head of the TOP500 list by 2025, without any requirement for an OSTP sponsored Grand Challenge. However, to go above and beyond that performance will require a serious research effort that begins now and seeks answers to fundamental questions about computation originally posed by Turing and von Neumann that remain unanswered.

From the standpoint of theoretical thermodynamic efficiency, how good would a 50 exaflops computer running on 20 MW actually be? It would expend an average of 0.4 picoJoules per flop (which includes all the energy and overhead for computing, communicating and storing), or approximately 20 attoJoules (10^{-15}) per each bit operation. A theoretical limit for a single irreversible bit operation is $kT \ln 2$, which at room temperature is ~ 4 zeptoJoules (10^{-21}), or still about four orders of magnitude lower than the potential exascale supercomputer of 2025. Clearly, there will still be enormous potential for further dramatic improvement, even without having to consider adiabatic or reversible computing. However, to realize that potential will require revolutionary changes in every aspect of the computer, from devices to system architecture and software. Since even relatively minor changes to computing hardware can require a decade to be implemented, new research thrusts that go beyond today's sustaining R&D to create new opportunities and paradigms are needed now as a Grand Challenge.

Indeed, this problem is widely recognized and there are several research programs in place, such as the NSF sponsored UC Berkeley Center for Energy Efficient Electronics Sciences (E³S), but an OSTP-sponsored Grand Challenge would strengthen multidisciplinary collaboration and lead to the critical mass required to have a significant impact. There are several efforts within the broader research community to self-organize and form ad hoc groups, such as the 9th Kavli Futures Symposium: The Intersection of Nanoscience and Neuroscience [Kavli 13], Sandia National Laboratories Neuro-Inspired Computational Elements [NICE 15], and multiple IEEE Rebooting Computing Workshops [IEEE 15]. However, a Grand Challenge would increase the scale and breadth of interactions to raise the likelihood of a major new breakthrough. Recently, the Semiconductor Industry Association (SIA) and the Semiconductor Research Corporation (SRC) issued a document entitled “Rebooting the IT Revolution: A Call to Action” [SRC 15] in which they propose a Grand Challenge similar to that described here, which they called the “National Computing and Insight Technology Ecosystem” (N-CITE) initiative. Clearly there is a broad recognition within the technical community of the need and opportunity for Rebooting Computing.

Brains’ advantage over CMOS-based Turing machines

What might be possible? A typical comparison is the human brain. With an order of magnitude uncertainty, it is usually estimated that there are $\sim 10^{15}$ synaptic operations in the brain per second. If these are simplistically equated to ‘bit operations,’ and guessing the brain expends about 20 W of power on computation, one can estimate that the energy per synaptic operation in the brain is about 20 femtoJoule, or 1000 times the bit operation estimate for the 2025 exascale computer. Does that mean that a 50 exaflop computer could outperform 1 billion humans? On certain types of computation, e.g. floating point arithmetic, the answer is yes. Thus, it is not just the number of operations that the brain performs per second that makes it special, but rather what is accomplished by those operations. Nearly all computers today are Turing machines and Turing believed that human brains are more capable than Turing machines. To replicate the brain’s learning, Turing considered supplementing a Turing machine with an Oracle that could periodically provide guesses or hints for the Turing machine to check. He nicknamed this combination an O-machine, but never got around to describing how the Oracle could be built. The community has refined Turing’s ideas in the intervening years, but a problem remains. The community has figured out how to write software for some learning tasks, such as software experiments in Deep Learning [Hinton 06] and Hierarchical Temporal Memory [Hawkins 11]. However, it takes a large CMOS-based compute cluster days to run a program that merely learns about cats in Youtube videos [Le 13] or prepares for Jeopardy matches [Ferrucci 10]. While these have been remarkable achievements, the brain provides an existence proof that learning could be more sophisticated and the compute platform smaller. The modern thinking is that it would be possible to write a program that could learn much more like a human and start it running on a Turing machine, yet with no assurance that it would finish running in a reasonable time.

We argue that an OSTP-based Grand Challenge could make the necessary advance for learning computers to be practical. With the community essentially concluding that a CMOS-based Turing machine would get bogged down trying to learn, the question should be whether a more capable non-CMOS, non-von Neumann computer could be devised that would be efficient enough for practical learning. The new technology would satisfy Turing’s definition of an Oracle to assist learning, yet in modern parlance it would provide a physically realistic computer that could accelerate learning beyond what is possible with a Turing machine – an idea where many consider the quantum computer to be

precedent. There will be some types of computing for which such a computer would not be appropriate, but it would be ideal for many of the new types of computing that are emerging, such as search and identify.

Many researchers believe that ‘neuromorphic computing’ represents the future, but brains are very different from computers. Presently there are at least two large brain science initiatives in the world: the European ‘Human Brain Project’ and the US BRAIN Initiative (Brain Research through Advancing Innovative Neurotechnologies), also referred to as the Brain Activity Map Project. However, the authors do not see these projects as strengthening the connection of brain function to computational theory. There are several small efforts around the world that are beginning to address this issue, and also one major collaborative effort. The Chinese Brain Inspired Computing Research (CBICR) program at Tsinghua University in Beijing has been running for three years with a total of 35 affiliated faculty from seven different departments. The structure of this program, which includes the research areas of connecting neural function to computational theory, modeling and encoding brain-inspired computing systems, software systems and programming, chip design, chip processing and integration with traditional hardware, and materials and device research, provides a template on which any comprehensive program could be built. This is where an OSTP Grand Challenge can provide the leadership to make the critical scientific discoveries and technical innovations to create super-fast and energy efficient ‘Sensible Machines’ a national priority, This is not and should not be a sentient machine, but rather a tool that enables humans to makes sense of petabytes and more of data that are constantly being generated in the world around us.

At the highest research level is the connection of brain function to the theory of computation, yet we believe this research alone will not bring run time or energy consumption into reasonable ranges. Next is to raise the performance of computers. The fact that brains are extremely nonlinear dynamical systems whereas current computers are based on Boolean logic gates opens the door to raising power efficiency via applying nonlinear network theory to manmade computers. Prof. Leon Chua of UC Berkeley has shown that biological neurons are ‘poised on the edge of chaos’ [Mainzer 13], in other words in their resting state they are biased very near chaotic behavior, and a small perturbation, such as a thermal fluctuation, can push a neuron to display chaos. This makes a significant fraction of individual neuronal firing look random, although it is probably not in reality. This contrasts with the approach used in computer simulation today. Simulations tend to abstract away details of the underlying physics in order to create efficient software subroutines for a von Neumann computer, with the abstraction process risking loss of important behaviors in the system being simulated and high power consumption of the computer doing the simulation providing a motive for shorter and potentially incomplete simulations. These attributes of simulation may well be discarding critical issues for understanding actual brain function. However, being poised on the edge of chaos may be the equivalent of Turing’s random number generator for an O-machine. Perhaps pseudo-random connections are constantly being made between states in the brain that are formally computationally intractable, and thus, according to George Dyson, the brain is continuously creating uncomputable answers to problems it has not yet been asked [Dyson 12], and the main trick is a search strategy to locate those states when they are needed. We suggest an OSTP-backed Grand Challenge further develop expertise in non-linear dynamics and apply this to new devices and new models of computation to increase power efficiency across the board.

The Sensible machines Grand Challenge

The OSTP RFI asks submitters to answer “What is the audacious yet achievable goal proposed?”

What research would be appropriate for an OSTP Grand Challenge? The goal should NOT be to build, emulate, or simulate a human brain, but rather to learn more about the nature of computation and how the brain performs some of its tasks in order to build Sensible Machines. The central thesis of this white paper is that although our present understanding of brains is limited, we know enough now to design and build circuits that can significantly accelerate certain computational tasks compared to existing technology; and as we learn more about how brains communicate and process information, we will be able to harness that understanding to create a new exponential growth path for computing technology. Just as modern airplanes do not have feathers or flapping wings, it is likely that the most important technological lesson we learn from brain function is how to encode more information per energy expended by utilizing nonlinear signals and circuits. What types of computing primitives are enabled by nonlinear dynamics and chaos? Brains are excellent at absorbing huge data streams from a multitude of sensors and comparing that data with a lifetime of stored information to make real-time decisions. That is where the bulk of computing is headed in the future – what has come to be known as ‘Big Data.’ In many cases, the best strategies to save computational time and energy may be to 1.) make multiple guesses at answers to a problem and then work backwards to verify the best solution(s) (instead of simply voting among several possibilities, which is what is done today), 2.) ‘remember’ that the same or similar problem has been solved previously and recall the answer, and/or 3.) ask an expert for help (e.g. perform a web search). These strategies may apply to a wider variety of problems than finding uncomputable states of tricky questions – for example identifying problems that are isomorphous to known systems, and could dramatically exceed any thermodynamic or other limits to computational efficiency by sidestepping actually solving the problem and developing analogies to data in memory.

The application in Figure 1A is an example Grand Challenge ‘deliverable’ to illustrate the interaction between the problem space and the technology solution. The diagram illustrates a spacecraft operating in an environment largely inaccessible to humans yet sensing and storing enormous amounts of data through a photodetector array coupled with a Sensible Machine to continuously observe the earth. The task is to identify significant changes over a wide range of time scales in the presence of a varying background. Over time, the Sensible Machine would collect and store raw pixels in a huge memory, such that even with modern communications technologies it would be impractical to download them all. Incoming observations are compared to historical observations in real time to identify issues of concern, such as shifting weather patterns and tornado formation. The Sensible Machine could recognize that something important was happening and download flagged data to earth for the attention of human analysts.

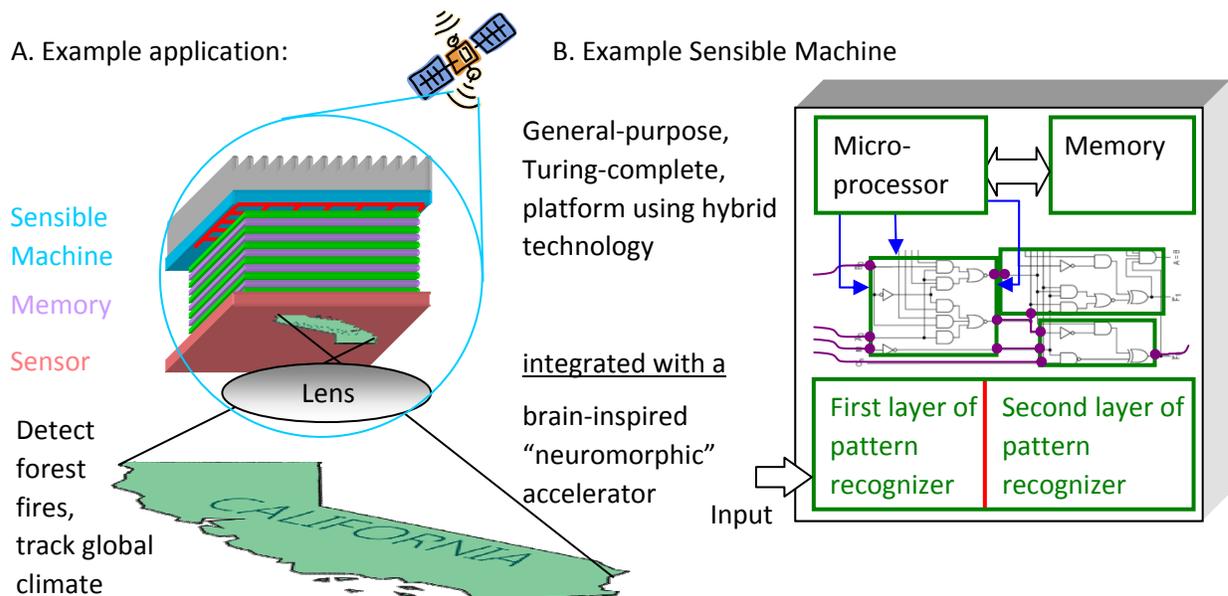


Figure 1: Example of an outcome from the Grand Challenge

While the need for making sense of observations is ubiquitous, this specific example has fundamental value:

- a. The system could follow dynamically changing, yet natural features of the earth, like normal cloud formations, but then flag issues of concern such as funnel clouds, abnormally large waves and incipient forest fires.
- b. It could monitor sea and air traffic, i.e. ships and airplanes, and spot new patterns.

Figure 1B is an example of the technologies in Sensible Machines, illustrating the intended scope of the Grand Challenge. The diagram depicts neural networks in green boundaries, yet showing some of the neural networks embodying computer logic gates to show the integration of brain-inspired computing and standard logic. The system adapts ideas from brain science, but will not necessarily duplicate the human brain in either structure or function.

Sensible Machines will likely include many different types of general-purpose or Turing-complete processors running fairly standard software. However, these decidedly familiar components

will be accompanied by other circuits and networks that use more brain-inspired operating procedures to learn, recognize patterns, and so forth. A new computer science discipline must be developed to enable Sensible Machines to learn from their environment, ultimately yielding optimized computer circuits and architecture that execute the learned behavior with extremely high efficiency.

Sensible Machines will require new types of electronic and photonic nanodevices that emulate some of the aspects of neural ‘circuit components,’ e.g. neurons and synapses, to complement scaled CMOS transistors in order to achieve the speed and energy performance envisioned in this white paper. Since a brain is a dynamical network of highly nonlinear components, the computing primitives in the brain are likely based on the nonlinear dynamics and possibly chaos. Thus, research into materials with extremely strong nonlinear responses to stimuli, such as phase transitions, and the physics of chaos, complexity and emergent behavior that arise from connected nanostructures, will be essential for creating these devices

There are at least four main research areas to explore:

1. Connect brain function to computational theory Collaborate with existing brain mapping projects, but confront the complexity of the dynamical nonlinear neural networks in brains. Do not rely on the ‘linearize, then analyze’ paradigm of the past, which will yield incorrect results for extremely nonlinear systems such as neurons and nanodevices. Such collaborations would yield understanding of the structure of the systems that engineers want to emulate, and they would almost certainly feed back understanding of the information theoretic mechanisms of communication and computation in biological neural networks. Use the knowledge gained to create new approaches to computing.
2. Devices and circuits in support of learning Build subcircuits with inorganic and electronic components that emulate basic synaptic and neural functions. Such circuits will provide breadboards for researchers to understand complex behavior and chaotic dynamics, and will enable them to reproducibly and deterministically alter, tune, and tweak the circuits in ways that deepen their understanding but cannot be done with probes in living brains. Such physical systems will be able to provide experimental data on highly nonlinear and dynamical systems that no computer program running on any existing computer could simulate in a finite time.
3. Task-specific hardware and systems software Using the above components and the understanding gained from their study, architect and construct CMOS-compatible systems that perform specific functions far more efficiently than CMOS alone, and use them as the basis for ‘accelerators’ to be employed in conjunction with more standard processors and memory for a hybrid System on Chip (SoC). Such an SoC could actually be a Sensible Machine, with the standard processors running defined algorithms and the neuro-inspired accelerators providing oversight, hints and guesses to speed up the computations. Different types of accelerator cores built for specific functions, such as image recognition, would form the basis for intermediate deliverables for the Grand Challenge, and would be technologically and economically valuable in their own right.
4. Applications Collaborate with researchers in quantitative psychology to understand the algorithms and ‘software’ that reside on biological neural wetware so that it can be adapted

efficiently for Sensible Machines. Such a system needs to learn how to identify anomalies, solve new problems, search its own memory for connected and unconnected states that contribute to a problem solution, and when, what, or who to ask for help. The ultimate in performance improvement comes when new neural algorithms run in an appropriate architecture built from dynamical components to address real world problems.

Such a Grand Challenge will harness the creativity and excitement of a multidisciplinary set of scientists and engineers to build a research community that can help crack what may very well be the most complex challenge and opportunity faced by science today. Much of the physical infrastructure for such a program is already in place at the five DOE Nanoscale Science Research Centers (NSRCs) and other National Nanotechnology Initiative- (NNI-) related facilities. Building exploratory devices and circuits will require the skills and facilities at the NSRCs, as will analyzing complex material structures with extremely fine spatial and temporal resolution. The neuro-inspired devices can be fabricated from CMOS compatible structures and materials. The next step up the hierarchy involves designing circuits and systems, and here the theory of nonlinear dynamical systems and how to model them computationally is critical. Finally, there is the issue of understanding how brain-like subsystems can be integrated into a computational platform, such as an SoC, which will both rely on and extend existing computing architectures. One of the major questions to be answered is whether the four orders of magnitude in energy per bit operation that will still exist between standard Exascale computers and thermodynamic limits are actually limits, or are there ways of finding solutions to problems that completely avoid much of traditional computation.

2. Applications of Sensible Machines

To focus the effort, the community needs to define a specific series of tasks leading to the Grand Challenge that will be representative of a large class of problems of interest to industry, government, and science. The ability to solve a growing range of increasingly complex problems within a specified energy budget would define a new growth path for computation, similar to Moore's Law. The general rule and example application areas are described below.

Restoring "Moore's Law"

Moore's Law created tremendous opportunities, efficiencies and economic growth. The goal of the Sensible Machines challenge is a new exponential improvement path based on devices, circuits, networks, and architectures that are dramatically more power efficient than scaled CMOS alone, thereby enabling new types of applications and saving substantial energy in executing today's applications. The opportunities from new types of applications are outlined below. Augmenting existing computing approaches with the highly nonlinear and dynamical operations that are the basis of brains is the most likely route to achieve this goal through the creation of new computing primitives.

Sensible Machines may allow a Moore's Law-like process with different parameters. The paper that established Moore's Law [Moore 65] is most readily interpreted as an exponential shrinkage of devices on a nearly fixed-size integrated circuit. This required a scaling relationship between feature size on the X-Y plane, power per device, and device speed. The Grand Challenge should establish another improvement path based on different advances. While details will emerge over time, it appears that scaling will occur through an increase in the number of features in the Z dimension coupled with device power reduction through improved manufacturing of new learning devices. These new devices would have higher theoretical limits on energy efficiency than binary electronic switches, giving room for practical improvements. While the previous era rated microprocessors by clock rate or numerical throughput, a learning computer would be more appropriately rated by speed and accuracy of learning.

Sensible responses in computer-computer interactions

The amount of data being collected and transmitted today far exceeds the capability of people to monitor, recognize, understand, and respond. Along with all of the data needed by human enterprises, computers also send information to each other, both in business and counterproductive enterprises such as malware and other national defense tasks. Humans are still "in the loop," as they write the software that is subsequently used in computer-computer interactions at lightning speed. A further rise in capability could occur by Sensible Machines learning and adapting their behavior more quickly than humans can write code.

An example of a goal for the Sensible Machines Grand Challenge might be the identification of malware at computer speeds the first time a specific instance is encountered [Mountain 15].

Enabling more effective human-computer interactions

Computers have raised productivity across the world, with Sensible Machines possibly extending this trend. For computers to further increase productivity will require them to interact more intuitively with users, requiring that the computer communicate on human terms. Years ago, this involved greater

use of pictures and touch (as opposed to keyboards). The frontier of this research area is for computers to process concepts instead of just moving bits and pixels from one place to another.

The IBM “Watson” Jeopardy competition [Ferrucci 10] is an example of this paradigm, but Sensible Machines would be needed for the essential real time operation, small form factor, and affordability needed in a consumer marketplace. IBM programmed a server cluster called Watson to play Jeopardy. Watson was trained over a period of days with knowledge that could be used to answer Jeopardy questions in real time. Watson then won a Jeopardy match up, answering questions posed in English within a few seconds and with more accuracy than a human competitor. While the Jeopardy game play is a compelling vision for a human-computer interaction, the days of preparation time on a server cluster limited its applicability.

Figure 2 [IEEE 13] is a representation of a human-computer interface that could be enabled by Sensible Machines. A user interacts with a personal Sensible Machine by asking questions just as in Jeopardy – or alternatively giving commands in English. However, the interaction is immediately fed back into the Sensible Machine and incrementally indexed or learned to create an up-to-date body of information for interacting with the user. The vision of this human-computer interaction paradigm has been widely explored; for example Apple produced a concept video in 1987 [Apple 87] of people interactively conversing with a tablet computer this way. While parts of this 1987 video clearly inspired today’s tablet computers, we now know that the computer’s conceptual understanding illustrated in a video with actors and simulated computers requires a server cluster to drive its computation even without real time responses. To be feasible as a widely used tool, a personal Sensible Machine would be needed that is not only fast enough for real time operation, but the purchase cost plus energy cost must be affordable by individuals.

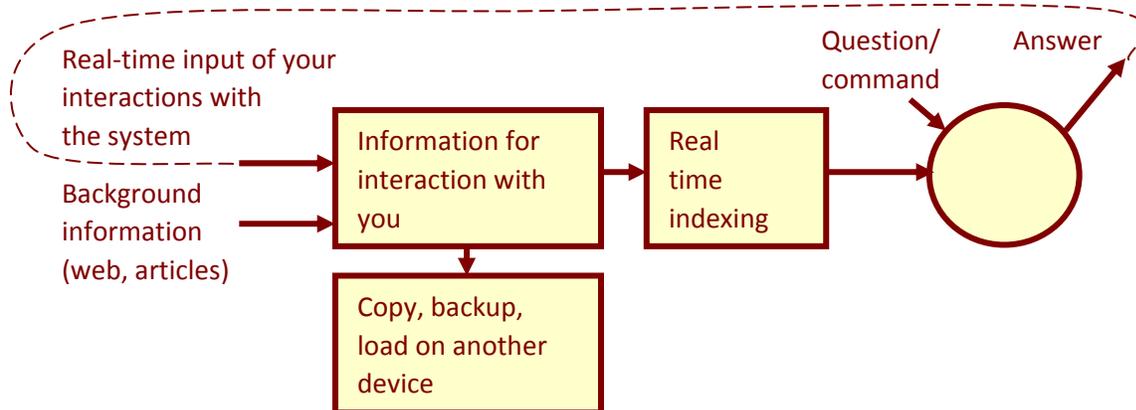


Figure 2: Sensible Machine data flow for facilitating human-computer interaction

Facilitating interactions with supercomputers

In the area of high performance computing, the numerical capability of supercomputers sometimes gets ahead of their operators’ ability to set up problems and make sense of the answers. A large deployment of a Sensible Machine could form an adjunct to a supercomputer. Input data from a particle accelerator, for example, could be prescreened for unexpected or hidden data patterns before being presented to the supercomputer for detailed physics simulation. Output from the simulation could be further processed by the Sensible Machine to learn new patterns.

3. Example of technical plan

Table 1: Summary of research activities

Topic	Topic areas	Lead orgs
Task 1: Connect brain function to computational theory		
(Task 1.1, Task 1.2, Task 1.3) Use concepts from the structure and function of the brain to design networks for recognition and learning (explicit bio-mimicry not required). Correspondingly, study the mechanisms for information encoding, storage, and retrieval in biological systems as a model for the state manipulation and data transmission in artificial nonlinear dynamical systems. Task 1.1 is an architecture entirely independent of implementation, Task 1.2 includes device and circuit implementation, and Task 1.3 further includes a scaling rule for application classes that specifies the evolution of manufacturable circuit parameters, managing heat flow, and electrical connectivity.	Brain science, computer science	IARPA, NSF, DoD (DARPA)
Task 2: Devices in support of learning		
(Task 2.1) Study the role of nonlinear dynamics and chaos in biological information processing systems and develop nonlinear dynamics as a new information processing primitive for computers. In analogy to the semiconductor industry search for alternatives to charge as a state variable, we suggest looking to nonlinear dynamics as a complement and/or alternative to digital information processing.	Brain science	NSF, IARPA, DOE, DoD (DARPA)
(Task 2.2 and Task 2.3) Research highly nonlinear material properties (e.g. phase transitions) and devices that are fast and power efficient for the mix of performance and learning encountered in various applications groups. Task 2.3 is specific to devices exhibiting local activity and chaos.	Physical science	NSF, DOE
Task 3: Task-specific hardware and systems software		
(Task 3.1) Develop standard computer algorithms (sort, search, etc.) and learning methods (back propagation, etc.) for a computer that learns. This would repeat the process of developing numerical libraries for supercomputers in past decades with libraries for the new activity of learning.	Computer science	DOE
(Task 3.2) Software and tools for controllable brain-inspired systems.	Computer science	IARPA, DoD (DARPA)
Task 4: Applications		
(Task 4.1) Develop one and subsequently more applications that solve a problem without being trained in the solution or storing it, using the preceding technologies to assure the result is as efficient as a human brain.	Applications domain	Varied

Overview – learning and energy efficiency

The introductory material described two types of objectives for this Grand Challenge. The first was the development within a decade of a computer that learns and the second was an improvement in energy efficiency both immediately and on an ongoing basis.

The highest research level proposed in this Grand Challenge is the connection of brain function to the theory of computation. The challenge being offered by Sensible Machines is to merge ideas from both brains and current computers in a way that has not received much attention to date. The idea is to use lessons from the brain to become a basis for computing, yet integrate this basis into a “software stack” like the ones on current computers. This would not become a sentient machine, but rather a tool that enables humans to makes sense of petabytes and more of data that are being generated in the world around us.

The next research priority is to address the low throughput of current simulations of brain behavior. The challenge is to find new physical devices that have optimal energy efficiency for the learning functions that are most common in Sensible Machines, because the energy efficiency can be higher than any structures based on logic gates.

Computers that learn

The authors propose developing a learning computer by reconciling the brain’s learning functions with computational theory, thus distinguishing this Grand Challenge from efforts in machine learning. The two large brain science initiatives in the world (the Human Brain Project and the US BRAIN Initiative) are each organized as shown in Figure 3A and Figure 3B and have the principle goal of mapping out the connection network of the brain, although it includes other brain studies. Both of these projects will provide extremely valuable data that can be used to help in the design of neuron-based circuits and systems. Yet the objective of this Grand Challenge is not to duplicate a brain.

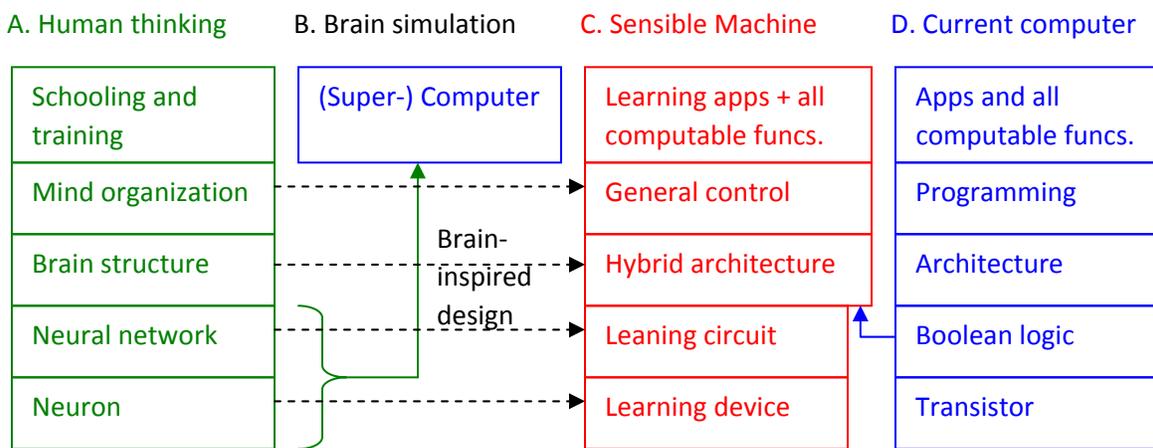


Figure 3: The Grand Challenge in context of other efforts

The computer industry uses a technology stack illustrated simplistically in Figure 3D that is different from the one in a brain. The computer industry has people doing programming, manufacturing, improving computers through physics and engineering R&D, etc., as well as technology for algorithms, programming languages, code libraries, etc. The proposed Grand Challenge would expand the computer

industry by developing a comparable technology stack, such as illustrated in Figure 3C. This would neither require nor preclude using the von Neumann architecture or existing source code, but rather creating a technology stack that enables many people to participate in the computer industry.

Developing Sensible Machines would require a computational theory derived from both conventional technology and ideas from brains. This is envisioned to be more than using the technology stack for current computers illustrated Figure 3D to simulate brains, as in Figure 3B. Proving neural networks to be equivalent to Turing machines [Killian 93] is a good first step, but the Grand Challenge objective would be to go beyond an “existence proof” and actually demonstrate a general purpose computer that learns. If there is to be success at this goal, it should be strikingly obvious that a C compiler could be produced for Sensible Machines, Linux ported, and then have all existing programs run on Sensible Machines. If these things are not obvious, the computational theory should be deemed incomplete. As a practical matter, it may be expeditious to use some (CMOS) components common to current computers in this hypothetical exercise. However, the Grand Challenge would not be considered successful if the computational theory is all due to technology that exists today.

There are also “neuromorphic” efforts to create a new type of computer based on neurons. The authors believe many of these efforts are heading in the right direction and should be increased in scope. However, the authors are unaware of efforts to alter the design of conventional computers to use brain-inspired methods as shown in Figure 3C (except for [Mountain 15], which is associated with the authors). For example, the authors are not aware of a neuromorphic effort where algorithms based on multiple neuron firings can compile and run an arbitrary C program. Support from an OSTP Grand Challenge could help support this objective.

Connecting brain function to computational theory

The discussion below shows a way to connect brain function to computational theory as in Figure 3C, preserving learning. The approach has minimal elegance and is merely intended as a starting point for research under this Grand Challenge, as opposed to a practical solution or a requirement. The method is to embed logic gates into neural networks of various types, while still preserving learning. General purpose computers can be built from the logic, on which all conventional software can run. The illustration will start with a specific computational task, expanding over the course of the exposition to include learning new tasks or variants of the original task.

Figure 4 is an example neural network for categorizing diseases using the very standard notation in the neural network literature of left-to-right information flow along lines, where the information is essentially the “strength of a concept.” For example, the “clump thickness” legend on the upper left would be a real number representing a physical dimension. The circles represent neuron bodies, and their function is to form a weighted average of the inputs. The weights are equivalent to synapse values and are the principal mechanism defining the function of the network. At an abstract level, this form of neural network applies to both brain function and artificial neural networks.

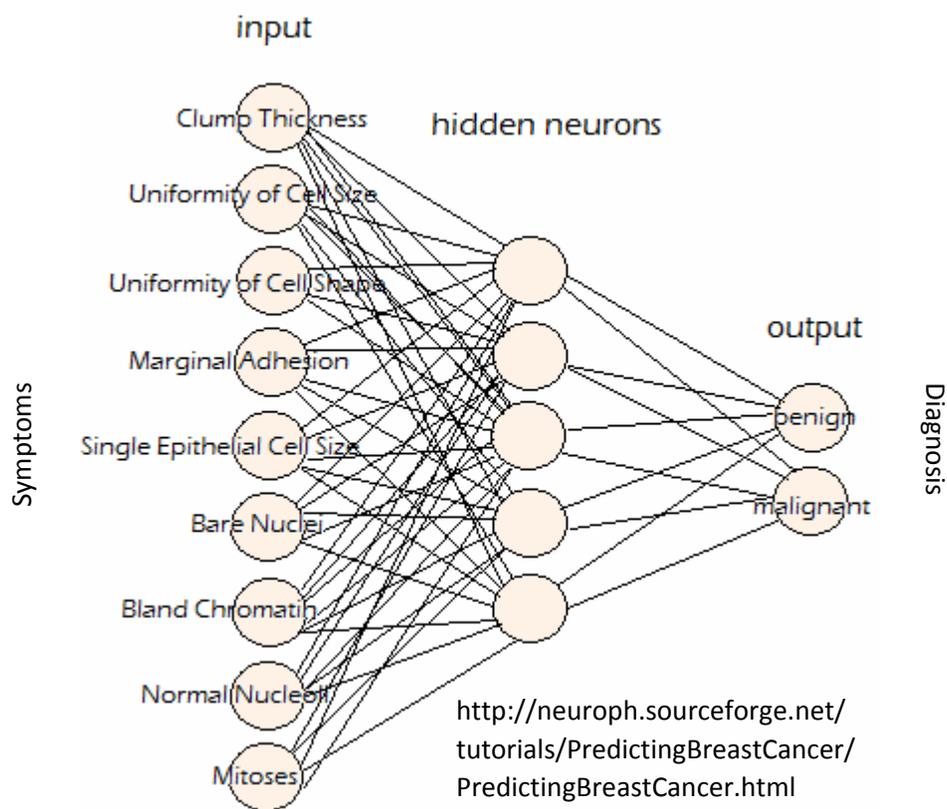


Figure 4: Neural network for disease categorization

However, the visual depiction used in neuroscience is somewhat unhelpful to an engineer trying to construct a work-alike using nanotechnology. We therefore deform (and condense) Figure 4 into the representation of Figure 5A, which replaces the clutter of lines with a rectangular array. In the new depiction, weights are at the intersections of rows and columns. Functionally, each neuron body computes the dot product of the inputs and weights. This method of drawing makes it obvious how to implement the circuit as an electrical array.

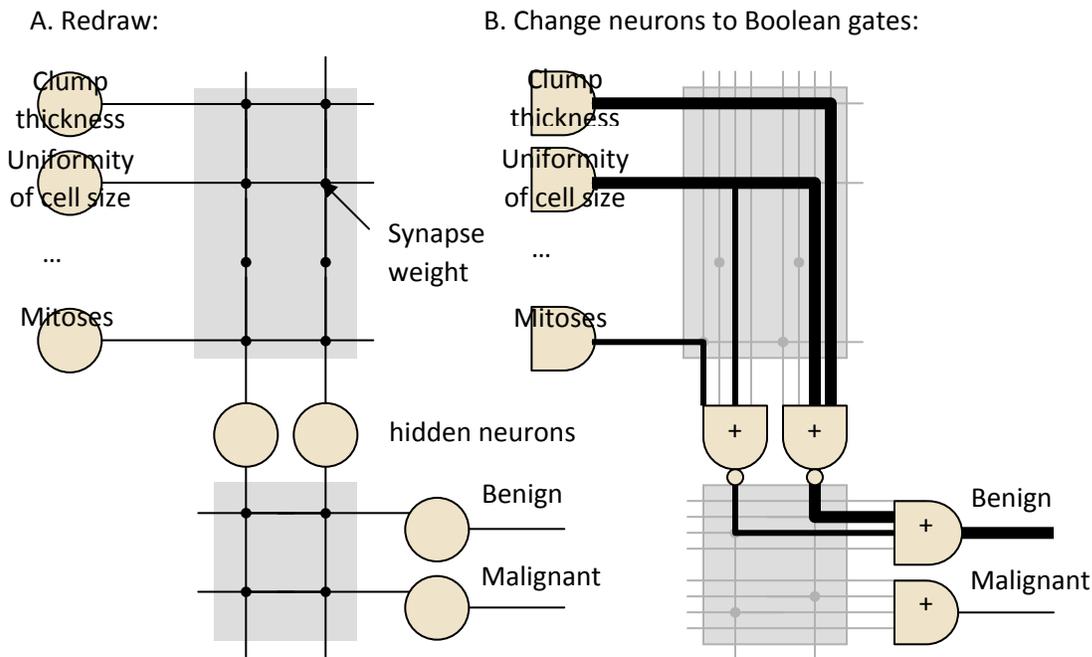


Figure 5: Computer logic in a non-learning neural network

Figure 5B makes a final change where the neuron bodies are replaced by Boolean NAND gates, a change that also mandates a different drawing convention. The drawing convention in Figure 5A is to draw gates with one input wire with multiple connections on the wire, whereas the convention in Figure 5B is to have multiple wires. We also replace the neuron bodies with NAND gates, which would duplicate the neural network behavior if the signaling conventions were chosen in a certain way and the NAND gate had a specific implementation. If the Boolean values are defined as 0 and 1, the Boolean gates AND, OR, and NOT become multiplication, addition, and subtraction. Switching the definitions of the Boolean values to 1 and 0 reverses the roles of AND and OR, which is the convention used here. If the intersections between rows and columns in Figure 5B multiply the input signal by a weight, and the NAND gates add, the circuit will perform the original disease categorization.

While the signals in the original neural network in Figure 4 were real numbers, we are free to restrict those real numbers to just the Boolean values 1 and 0. If we do this, the circuit in Figure 5B will implement a Boolean logic circuit. If we made a larger version of Figure 5B and put the logic diagram for a microprocessor into the diagram, we would have embedded a microprocessor into a neural network. A microprocessor is a general purpose, or Turing complete, computer, so we have just shown a method of making a neural network that can also perform general-purpose computing [Mountain 15].

However, Figure 5B has another interpretation based on the wiring of the logic network. A weight of 0 at an intersection effectively disconnects the row and column, whereas a weight of 1 creates a wire shown in Figure 5B as a thick black line. Weights between 0 and 1 are deemphasized by being drawn as thinner lines; their function will be discussed below. Interpreted this way, weights are similar to synapses in HTM [Hawkins 11]. Thus, a neural network can become any logic network with a suitable

setting of weights, provided that the neural network has enough layers and intermediate (or hidden) neurons.

Neural networks are not generally restricted to Boolean inputs and weights. In fact, the disease recognition network in Figure 4 will not function correctly with Boolean variables. When neural networks are trained through algorithms like back propagation [Widrow 60], weights take on real values in fine gradations over the course of learning. In the context of Figure 5B, there will be strong wires representing the logic of the circuit as well as weak wires that result from training data that was inconclusive. Weak wires may not impact the behavior of the circuit right away, but weak wires could become strong wires after a small amount of additional learning.

This section shows a path for adding new features to computer technology that will allow computers to learn different behaviors. A neural network can contain logic or even a whole computer. However, a sufficiently large neural network could contain more than just one specific instance of logic, as Figure 5 shows that the specific logic function is determined by weights that are learned. A neural network is also more than just a container into which a person can load or train a specific logic function (like an FPGA or one computer programmed to simulate another). As shown in Figure 5B, the strong and weak wires in Figure 5B show how a neural network contains many potential behaviors. The strong wires are the best logic diagram based on previous learning, but alternative networks with weak wires are also present. One of these alternatives may become the best solution after additional learning. Maintaining all the alternatives consumes resources, but discarding the alternatives would eliminate information essential to future learning.

(Task 1.1) Brain science has multiple models that fit the diagram in Figure 4, such as the level-based perceptron and spike trains. One area of research activity would be to devise an artificial neuron behavior that has an understandable connection to brain activity. Then, identify how the brain uses the information processing capacity of the neuron to do things beyond the capability of computers. The state variables of neurons are ion currents and molecule concentrations, which are extremely slow and require significant energy. How then do brains perform so much computation using so little power?

Computational complexity and Sensible Machines

A logic network embedded in a neural network will lead to a computing model that has more throughput than a one of today's computers, providing the additional computational resource to support learning. The discussion below will be in the language of computational complexity. Computational complexity is a branch of computer science that deals with solving problems of size N , such as sorting a list of N numbers or inverting an $N \times N$ matrix. For a problem of size N , computational complexity would quantify the number of computational operations, number of time steps, amount of energy, and other factors when run on a computer of a specific type. Computational complexity is usually stated in "Big-O notation," which describes the limiting behavior of a function when the argument tends towards infinity, usually in terms of simpler functions. We will see that changing from a von Neumann type computer to a Sensible Machine produces a big boost in throughput.

Figure 6A illustrates the computing model for either a Turing machine or a von Neumann computer. Both these computers have a memory holding N symbols, such as bits. Both sides of Figure 6 have been drawn so that the volume of the computer is $O(N)$, where volume represents cost or number of devices. Having a memory of size $O(N)$ is a very basic requirement for solving a problem of size N , as it

is needed merely to hold the problem. The mathematically abstract Turing machine uses a tape (illustrated) that can be reeled back and forth one symbol per unit time. A von Neumann computer can access any location in unit time, which is faster in a practical sense. However, both a Turing machine and a von Neumann computer have a CPU that performs one operation per unit time. This is illustrated by the small tape head or CPU. Both systems have $O(1)$ processing capacity per time step for a system with $O(N)$ memory. Since each memory location requires a device, like a transistor, they will have $O(N)$ cost.

A. Turing/von Neumann computers:

B. Sensible Machines:

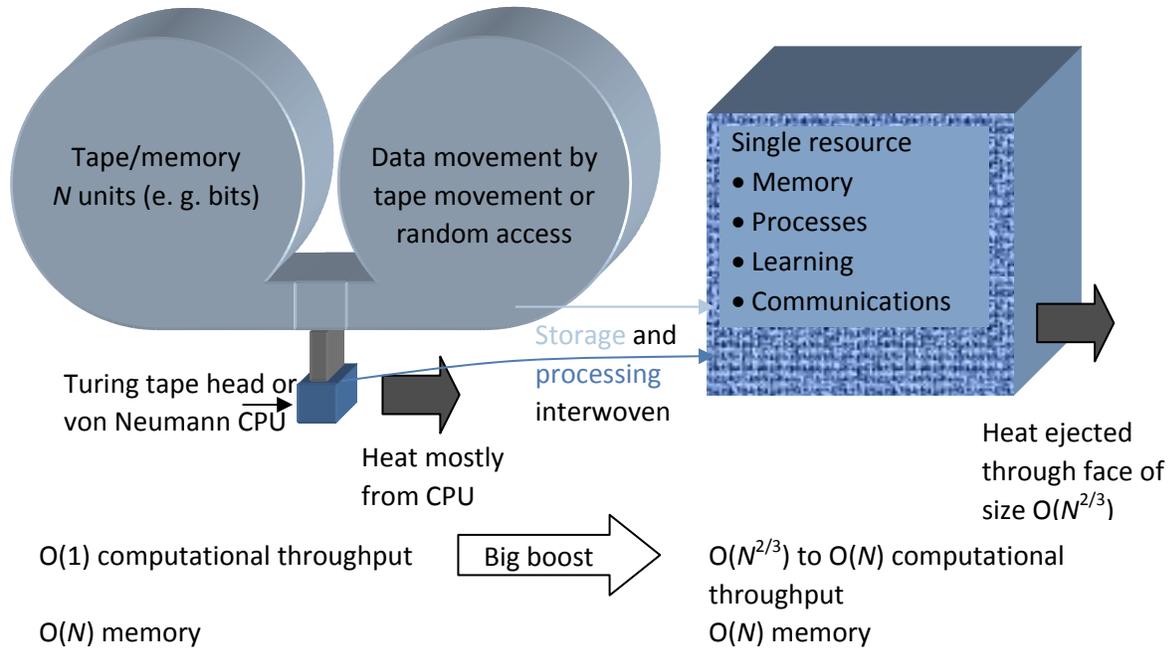


Figure 6: Sensible Machines complexity model

Like brains, Sensible Machines are expected to have a single resource that both stores data and processes it, as illustrated by the woven fabric of Figure 6B. The system can store N symbols just like a Turing machine or von Neumann computer, but could have many neurons processing at once.

A heat issue must be considered for both brains and the Sensible Machines. A hypothetical 3D computer could generate heat anywhere inside a volume of size $O(N)$, but can only dissipate the heat through a surface of size $O(N^{2/3})$. Brains concentrate neurons on their surface, leading to similar properties. We will claim a Sensible Machines could have between $O(N^{2/3})$ and $O(N)$ throughput unit time for a system with $O(N)$ memory and hence $O(N)$ devices and cost.

(Task 3.1) We can embed a microprocessor into a neural net and thereby show how to compute any computable function with at most a polynomial slowdown over the optimum running time or some other resource consumption. While this is an existence proof, a polynomial slowdown when N is in the Exascale range (Exa = 10^{18}) would be catastrophically inefficient. In contrast, there is a field of study for algorithms such as searching, sorting, numerical algorithms, data structures, and so forth that seeks to find effective algorithms for different computer structures. The research area is to co-develop the Sensible Machines architecture in conjunction with a theory of algorithms so that the overall computer

will perform well on both the standard sets of algorithms on today's computer and on new algorithms that include learning.

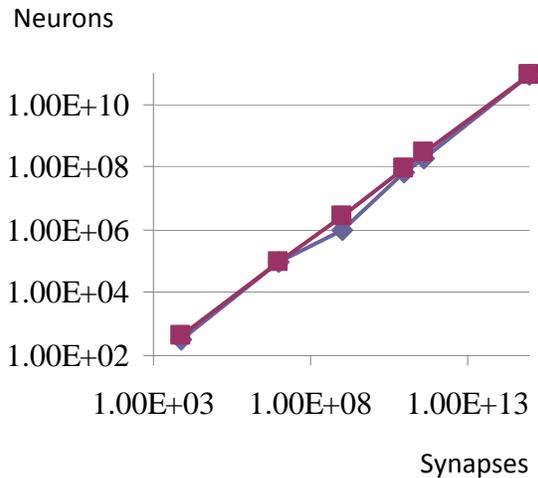
Scaling and Sensible Machines

The semiconductor industry builds logic and memory from nearly the same technology, but a mixed 2D/3D arrangement may be more effective for systems like Sensible Machines that learn. The amount of computation per unit of memory usually falls as applications become more sophisticated. For example, the evolutionary sequence of organisms created by nature includes a scaling rule for the sequence of brains that control these organisms. Table 2 shows the number of neurons versus synapses along this sequence [Wikipedia], where synapses are essentially memory and neurons are the computation devices. The tabular data is graphed in Figure 7A along with the curve $\text{neurons} = \text{synapses}^{3/4}$, showing a close fit to the polynomial relationship. Figure 7B shows Kryder's Law [Walter 05] for disk drive capacity and computer clock rate, showing that computer system storage scaled nearly twice as fast as computer throughput.

Table 2: Synapses versus neurons in evolutionary sequence (Wikipedia)

	Synapses	Neurons
Roundworm	7.50E+03	3.02E+02
Fruit fly	1.00E+07	1.00E+05
Honeybee	1.00E+09	9.60E+05
Mouse	1.00E+11	7.10E+07
Rat	4.48E+11	2.00E+08
Human	1.00E+15	8.60E+10

A. Synapses vs. neurons across evolutionary sequence:



Left: Neurons vs. synapses vs. curve of $y = x^{2/3}$ for evolutionary sequence.

Right from Wikipedia, which states the diagram as © Creative Commons.

B. Kryder's Law:

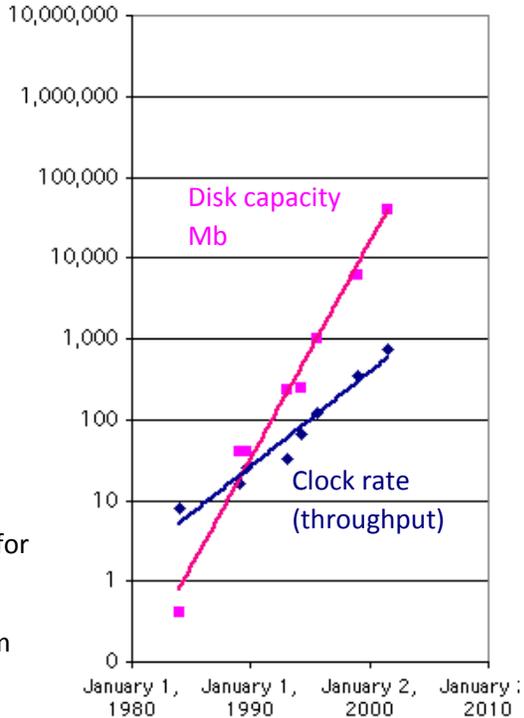


Figure 7: Scaling of real systems, a probable model for Sensible Machines

The position in this white paper is that Sensible Machines will require a lot more memory than non-learning computers for the same function. They will not only need memory for the behavior they have in common with a non-learning computer, but also for the alternate realizations that were part of the learning process and which might be activated by a small amount of additional learning (the thin wires in Figure 5). The amount of additional memory is expected to be more than a multiplier, but rather a ratio that scales, or grows over time. Figure 7 shows that real systems have this behavior. A 3D system with memory distributed in its volume but computation only on the surface will have throughput scale as memory^{2/3} (or alternatively, memory would scale as throughput^{3/2}) which is pretty close to the behaviors in Figure 7.

(Task 1.3) The computer industry has been successful due to applications that scale up over time, thereby solving a greater range of problems and generating economic growth. If the underlying hardware technology scales at a different rate from applications, the resulting computers will become unbalanced and inefficient over time. An area of research would be to define the Sensible Machine computer architecture in terms of structure, function, and applications where the asymptotic scaling rates match.

Architecture

The architectural challenge is to find a new computing approach, quite likely to be a combination of design principles illustrated in Figure 8. This includes synapses that both store

information and create links in a network, as illustrated in Figure 8A using the brown color to represent the synapses that store information in brains.

A. Natural neural network deformed to show equivalence:

B. Artificial neural network:

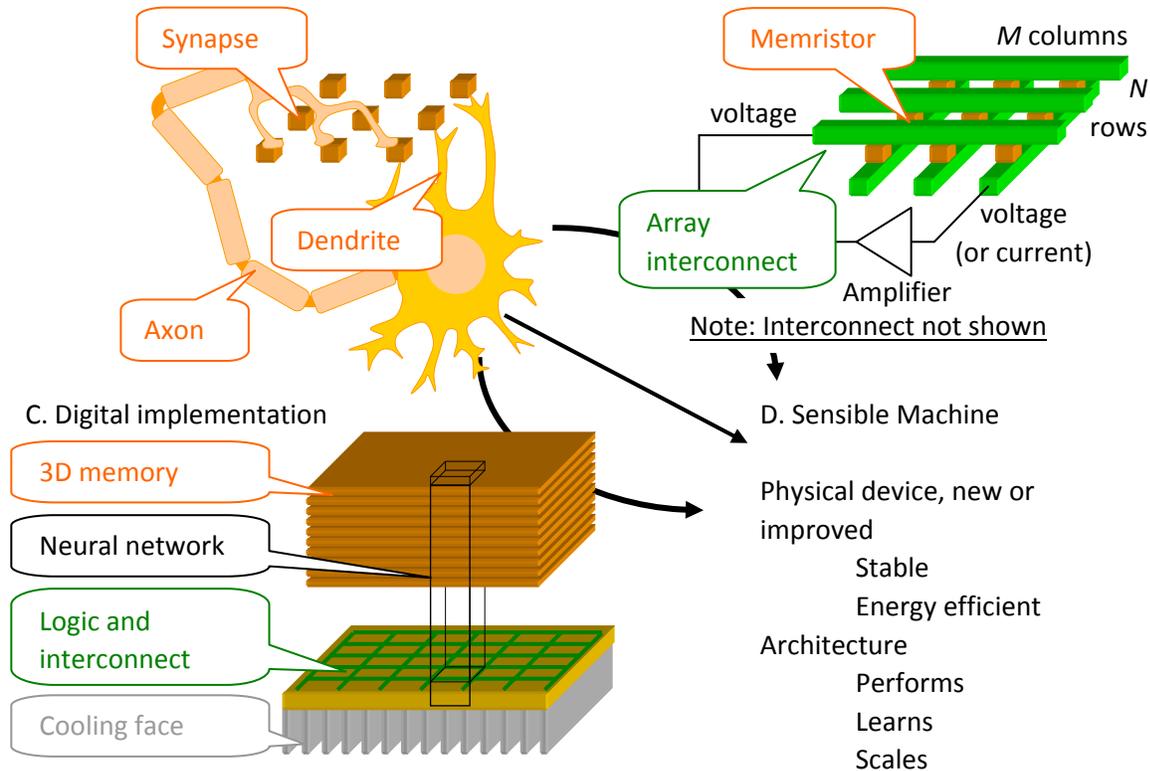


Figure 8: Routes to a Sensible Machine architecture

There is current research on a class of electrical circuits depicted in Figure 8B that use one of a variety of electrical devices (nonvolatile memristors such as ‘RRAM,’ Phase Change Memory, and Spin Torque Transfer – shown in brown in the diagram) as the storage devices in an array similar to Figure 5. These systems are promising at small scale, and there are concepts that allow them to scale to very large sizes. Large scale test beds are readily available today in the form of supercomputers or Graphical Processing Unit (GPU) clusters, and these are being used to test software ideas under such names as Deep Learning. These test bed systems could evolve into a new digital system as represented graphically in Figure 8C. The limitation is that current approaches do not yield systems that come very close to theoretically expected performance levels. The Grand Challenge is to produce architectures such as in Figure 8D that have high performance at scale.

(Task 1.2) The result of the previous sections would be a family of parts that performs both logic and learning, like a universal set of logic gates except that it is not only universal for logic but for learning the logic as well. The new research in this section is to figure out how to implement these universal primitives in actual circuits. Just as electronic logic families are engineered to restore digital signals faster than imperfections accumulate, the research required in this area is to understand the engineering properties of the new computational primitives – not only the behaviors that we want but other behaviors due to noise, parasitics, manufacturing variance, and so forth.

The physics of computation and learning

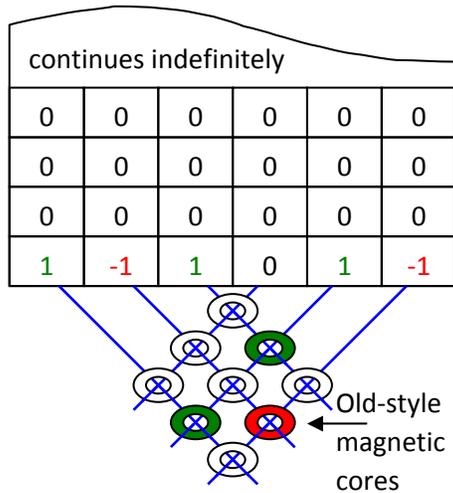
We will attempt to reconcile the apparent high power efficiency of brains with the technology in computers, showing how devices with both state and state-dependent non-linear dynamics could use brain-like principles to raise the energy efficiency of manufactured systems. The necessary theory was developed in the 1960s, yet the industry's separation of computers into Boolean logic gates and memory caused essential degrees of freedom to be taken away from designers. It was stated earlier that there is a widely accepted minimum energy limit of $kT \ln 2$ joules per irreversible logic operation [Landauer 61]. While the $kT \ln 2$ limit applies irrespective of materials and devices, the limit depends on the statistics and context of the function being performed. Sensible Machines will have a constrained usage pattern for learning that is different from the one apparently considered by Landauer, opening the door to finding new devices that have lower minimum energy consumption for learning.

While learning is essential, most experiences do not cause a given synapse to change state. For example, most readers of this white paper will have learned the alphabet as a child. By now, there is nothing more to learn by seeing the letter "L" for the millionth time. However, seeing the letter "Л" may invoke learning and cause synapse changes for readers unfamiliar with the letter equivalent to "L" in Russian (Cyrillic).

When a von Neumann computer updates storage devices used as synapses, a CPU typically runs a floating point subroutine to compute updated values for every synapse [Hinton 06] [Hawkins 11]. If the synapses are already set to their correct values, this calculation simply regenerates the existing values. However, electrical energy will have been dissipated for thousands to millions of gate operations to essentially verify that the system had already learned what it needed to know.

Figure 9A shows an approach to obtaining lower energy, yet implementing the behavior with the Boolean logic gates in Figure 9B has a higher minimum energy. While minimum energy of logic may be due to crude devices, we show in this section that organizing devices into gates imposes another minimum energy limit due to the circuit. Figure 9A is the simplest special purpose computer the authors have been able to devise to illustrate the low energy limit for learning, but it is not intended to be complete or even to use relevant devices. The environment provides an input data stream $2n$ symbols wide to be learned via a minor variant on the delta learning rule [Widrow 60]. The symbols have values -1, 0, and 1. The machine has an $n \times n$ array of synapses, but the synapses are implemented as old-style magnetic cores that flip when exposed to a magnetic field above a threshold. The machine divides the input symbols into two groups or vectors, using an update rule described below to effectively set or reset symbols in response to the outer product of the two vectors. As the input stream flows downward, the values on the bottom row are translated into current in the blue wires. Let's say a symbol value of 1 becomes a downward current, -1 an upward current, and 0 no current. The system would be engineered to flip magnetization at ± 1.5 units of current flowing through each core. Thus, a core exposed to 1, 1 on the two wires will turn green indicating 1 and when exposed to -1, -1 the core will turn red indicating 0. Magnetic cores dissipate energy when they change state, but essentially zero energy otherwise. Other combinations of symbols create a current below the threshold and there will be no state change and no energy dissipation. The state of cores shown in white is not relevant to this discussion.

A. Physical computer model



B. Equivalent gate-level synapse behavior

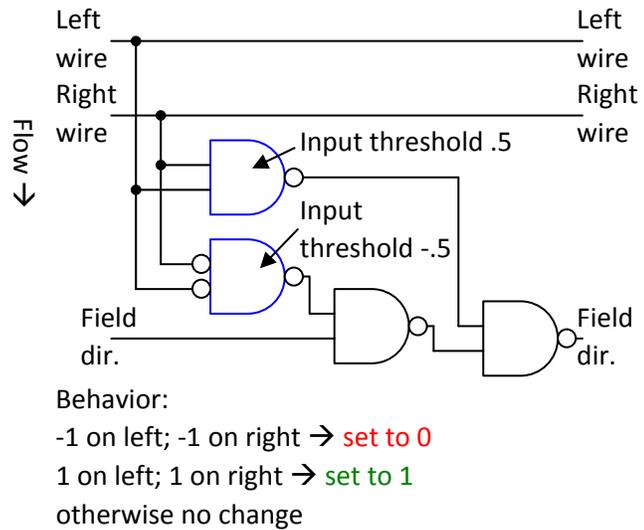


Figure 9: Delta learning rule computer

The average energy per learning cycle can be arbitrarily close to zero if we make a reasonable assumption about the input. Let us require that the input data will not cause something to be learned and then subsequently unlearned. The machine may be attentive to learning new things for an arbitrarily long time, but this requirement avoids synapses cycling between two values and consuming energy. Irrespective of the length of the input stream, there can be no more than n^2 flips. The low energy limit in the worst case that all n^2 synapses flip will be $n^2 \ln 2 kT$, due to each flip erasing one bit of data. Since the input stream could be of infinite length but the dissipated energy is finite, the average energy per step would asymptotically approach zero as the input stream gets longer.

This discussion is completely compatible with theory established in [Landauer 61], but the entire theory must be applied. A careful reading of [Landauer 61] reveals that the energy dissipation per machine cycle depends on the weighted average of the information processed or “erased” over all operations. Landauer assumed input patterns would be evenly distributed, an assumption clearly stated in the paper and reasonable in an era when companies sold logic gates for buyers to integrate into their computers and feed any distribution of input patterns they chose. This implicit tie to stand-alone logic gates is visible in the machine Landauer used in the 1961 paper (reproduced in Figure 10A as an embedded Excel spreadsheet, specifically the portion with white text on a black background like in the original paper). We have augmented Landauer’s chart with the probability of each input in orange and the entropy per output state in purple. This leads to the spreadsheet calculating the entropy change $S_f - S_i$ in **boldfaced red** of **.823959** k, which is on the order of k, or kT^1 joules.

¹ There is an error in Landauer’s paper: The paper reports this difference as 1.18 k.

A. Landauer's paper figure 5

prob	p	q	r		p1	q1	r1	Si (k's)	State	Sf (k's)	
0.125	1	1	1	→	1	1	1	0.25993	α	0.25993	
0.125	1	1	0	→	0	0	1	0.25993	β	0.25993	
0.125	1	0	1	→	1	1	0	0.25993	γ	0.367811	
0.125	1	0	0	→	0	0	0	0.25993	δ	0.367811	
0.125	0	1	1	→	1	1	0	0.25993	γ	0	
0.125	0	1	0	→	0	0	0	0.25993	δ	0	
0.125	0	0	1	→	1	1	0	0.25993	γ	0	
0.125	0	0	0	→	0	0	0	0.25993	δ	0	
									2.079442	Sf (k's)	1.255482

B. Sensible Machines white paper figure 9

probability of a learning event:										Si-Sf (k's)	0.823959
0.001											
	left wire	right wire	field dir.		left wire	right wire	field dir.	Si (k's)	State	Sf (k's)	
0.0624375	-1	-1	-1	→	-1	-1	-1	0.173176	A	0	
0.0624375	-1	0	-1	→	-1	0	-1	0.173176	B1	0.173176	
0.0624375	-1	1	-1	→	-1	1	-1	0.173176	C1	0.173176	
0.0624375	0	-1	-1	→	0	-1	-1	0.173176	D1	0.173176	
0.0624375	0	0	-1	→	0	0	-1	0.173176	E1	0.173176	
0.0624375	0	1	-1	→	0	1	-1	0.173176	F2	0.173176	
0.0624375	1	-1	-1	→	1	-1	-1	0.173176	G1	0.173176	
0.0624375	1	0	-1	→	1	0	-1	0.173176	H1	0.173176	
0.0005	1	1	-1	→	1	1	1	0.0038	I	0.174061	
0.0005	-1	-1	1	→	-1	-1	-1	0.0038	A	0.174061	
0.0624375	-1	0	1	→	-1	0	1	0.173176	B2	0.173176	
0.0624375	-1	1	1	→	-1	1	1	0.173176	C2	0.173176	
0.0624375	0	-1	1	→	0	-1	1	0.173176	D2	0.173176	
0.0624375	0	0	1	→	0	0	1	0.173176	E2	0.173176	
0.0624375	0	1	1	→	0	1	1	0.173176	F2	0.173176	
0.0624375	1	-1	1	→	1	-1	1	0.173176	G2	0.173176	
0.0624375	1	0	1	→	1	0	1	0.173176	H2	0.173176	
0.0624375	1	1	1	→	1	1	1	0.173176	I	0	
									2.778417	Sf (k's)	2.772585
										Si-Sf (k's)	0.005831

Figure 10: Minimum energy of the system in [Landauer 61] and an exemplary synapse

To show that $kT \ln 2$ is not a limit for learning machines, we extended the Excel spreadsheet to Figure 10B for the minimum dissipation of one “core” or synapse of the machine in Figure 9A. In accordance with the discussion above, an important attribute of the learning process is the (small) probability that a synapse actually changes (flips) due to an input – as opposed to merely “verifying that it has learned what it needs to know.” The spreadsheet includes the parameter shown in green for this value (taken as .001), which is used by Excel to compute the left hand column of probabilities. Figure

10B is now in the same form as the 1961 analysis, but with a different state transition pattern and unequal input probabilities. Observe in the entropy change $S_i - S_f$ in **boldfaced red** of **.005831** k, which is about a factor of 100 less than $kT \ln 2$. A generalization of Landauer's approach, which can provide fundamental device-agnostic dissipation bounds for more complex computing scenarios—and reveal efficiency advantages of conditioning behavior on memory—is described in [Anderson 13]. (By the way, the factor of 100 is not fundamental but is nearly proportional to the learning probability parameter in **green**. The authors will attempt to provide the spreadsheet among supplementary materials to this document, from which the reader can observe the functional dependence.)

Why is the result in Figure 10B so much lower than $kT \ln 2$? Landauer created a process 55 years ago for assessing minimal heat generation per machine cycle that he claimed (in the abstract of [Landauer 61]) would yield results “typically on the order of kT .” The current authors reverse engineered Landauer's process and intuitively figured out a way to make an atypical system that would dissipate substantially less than kT . This involved merging functions that are usually performed by multiple gates into a single system to avoid each gate dissipating heat that cannot be recovered and then exploiting unequal probabilities in the data set being processed. In essence, the cores in Figure 9A and recently discovered devices (e. g. memristors) have both logic and state in the same device. A core will only flip if both wires carry current and the current threshold is exceeded, implementing the NAND gates in **blue** in Figure 9B. A core will not flip twice to the same state, implementing the remaining gates in Figure 9B. All the logic just described comes for free if the core doesn't flip. The authors arranged the logic in Figure 9B so the input pattern that occurs with 99%+ probability happens to be the pattern where the logic is free. The method just described is likely to apply to more than just learning, yet the theory for this remains to be developed.

The rationale above is part of the justification for research on devices with state and state-dependent nonlinear dynamics. In order to perform the energy reducing process in the paragraph above, the device must have state, and the state has to affect the way its behavior changes over time. If the device has both state and state-dependent behavior, it can be pushed through a dynamic path to perform logic (albeit perhaps in an unfamiliar form). The challenge will be to explore devices whose function in a learning circuit analyzed like Figure 10 shows exceptionally low energy dissipation, then devise a method of manufacturing the device on a chip or perhaps 3D structure.

There is possibility, albeit controversial, that the energy efficiency improvement from Figure 9B to Figure 9A could be substantial enough to allow a learning computer to be a Turing O-Machine or a Super Turing machine for learning. Being poised on the edge of chaos may be the equivalent of Turing's random number generator for an O-machine.

(Task 2.2) We expect Sensible Machines to use underlying devices differently than CMOS and von Neumann computers. The devices will be expected to both learn and perform, the former activity not occurring in current computers. Learning is heavily skewed towards the operation of “verify that the device has learned what it needs to know,” which is an operation that should not need to consume energy according to established theories. The research activity is to devise new devices or optimize existing ones that perform very well for the usage patterns in Sensible Machines.

Nonlinear dynamical devices and systems

There has been an extensive search by physical scientists for new materials, devices, and principles suitable for computers, but this search has mainly focused on improving transistors or finding a faster and/or lower energy state variable than electrons for computing on a gate. However, this approach overlooked the issue that the brain, which is considered to be very efficient, actually uses ion currents and molecular concentrations to conduct and process information, and that data tends to be stored in a form more like real numbers than bits. Brains are highly nonlinear dynamical systems, and the information content in such systems is enormous, as anyone who tries to simulate them with a von Neumann machine knows. The previous section gave a theoretical basis for the combination of state and state-dependent dynamical behavior to raise energy efficiency. However, an understanding of how information is processed by nonlinear dynamics will be essential to discovering actual devices that approach the theoretical limits. One notable property of nonlinear dynamics is chaos, which has long been associated with aspects of brain function. A significant property of a system that is 'poised on the edge of chaos' is an extremely rapid and thus low energy response to a stimulus, which for instance leads to the sharp spikes of the action potential of a neuron. There are classes of nonlinear nanoscale memristors that display a similar sensitivity, as illustrated in Figure 11A, which shows a chaotic spike train emitted when a small DC bias voltage is applied across a thermo-electric material that undergoes a phase change on Joule self-heating. The particular material used for this example is NbO_2 , which is a well-known correlated electron material that undergoes a Mott insulator to conductor transition with increasing temperature. This leads to a negative differential resistance, or local activity, which enables the memristor to have gain and amplify a signal for a particular range of DC biases. Such materials were considered in the past for electronic switches, but at micron size scales the energy required to cause them to change state was prohibitive. However at the nanoscale, only tens of attoJoules are required, which makes them very attractive for electronic circuits. These devices have been used to construct a 'neuristor' [Pickett 13], an electronic circuit that emulates the action potential of a neuron, but is 1000 times faster and utilizes only 1% of the energy of a neuron in producing a spike (see Figure 11B). Thus, using such circuit elements has the potential for creating extremely efficient information process systems, if they are utilized to express the appropriate computation primitives in the right architecture.

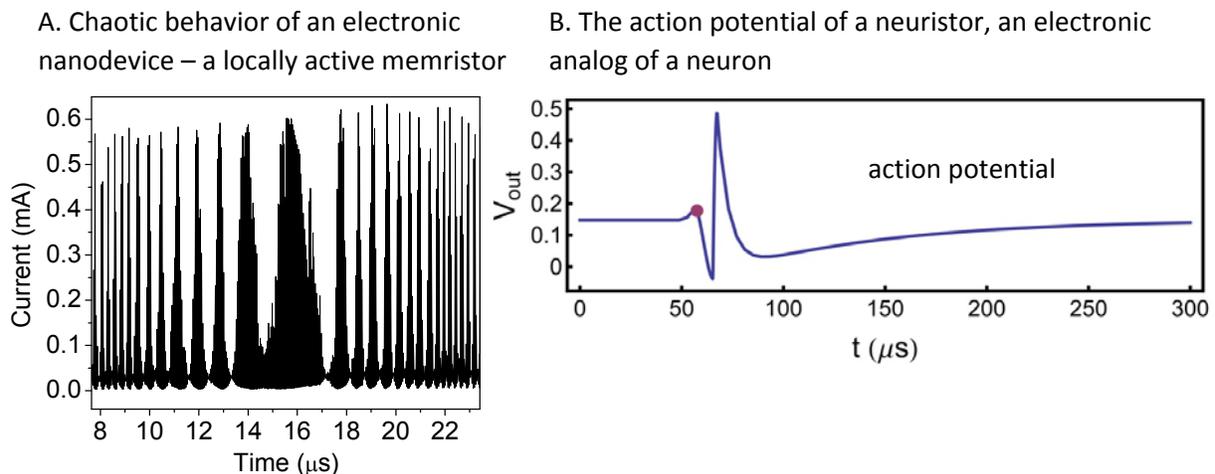


Figure 11: Nanodevices

(Task 2.1) Explore nanoscale materials, device physics, and circuits. Examine nanostructures that change state rapidly with minimal energy cost and hold that state indefinitely (nonvolatile examples: spin-transfer torque, phase change, ion drift) or release it instantly (volatile example: Mott insulator).

Emphasize systems with strong nonlinear response to temperature, voltage, etc. to find deep subnanosecond and subpicoJoule digital and analog switching/state change. Expect that the nanoscale will produce and/or enhance such nonlinear behavior as oscillations, gain and chaos in familiar materials. Analyze the dynamical behavior of networks of nanodevices and observe the information flow and processing. Leverage CMOS foundries for materials and fabrication technology. From the circuits, work with computational theorists to create a computational model for the behaviors, including the ability to shift devices across the edge of chaos as a computational primitive.

(Task 2.3) Understand the function of nonlinear dynamics and the relation to information flow and processing in biological neural networks. Develop a theory of computational primitives based on nonlinear dynamics. Elucidate the role of chaos in these systems. Involve the applied mathematics community to reinvigorate the theory of chaos and nonlinear dynamics developed over 30 years ago, as exemplified by the 1987 publication of the popular book *Chaos: Making a New Science* [Gleick 87], but not fully applied to the neural or computational domains in the interim. Confront data from real experimental measurements that contain measurement uncertainties, limited bandwidth and all sources of noise rather than idealized toy models.

Software tools and applications

If Sensible Machines are to become a new class of computer, there must be a path to create versions for different problem domains. An engineer would specify a Sensible Machine in terms of the required memory and computing capacity and the configuration of processing resources – in a method perhaps similar to the way engineers set up multi-level neural networks for Deep Learning. However, a Sensible Machine would achieve its performance and efficiency by a customizing a hardware design instead of just setting up software parameters.

Two examples are shown in Figure 12; Figure 12A illustrates a typical setup task for a convolutional neural network for Deep Learning. The analogy to programming is the engineer setting up a processing pipeline, specifying the number of levels, the number of neurons at each level, and the behavior at each level, such as autoencoders, perceptrons, etc. The engineer would also create training sets and testing sets.

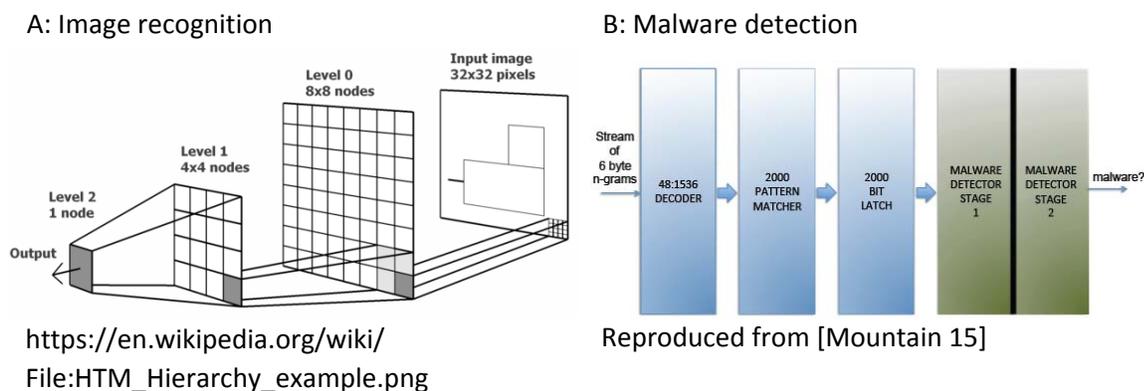


Figure 12: Example of problem setup

The example in Figure 12B is a Sensible Machine prototype set up for malware detection. The engineer sets up a processing pipeline like Figure 12A, but the first three levels in blue are actually digital logic embedded in neural networks; the last two levels are traditional neural networks. The whole structure is similar to Figure 1B.

(Task 3.2) Invent a software system that learns something and then synthesizes a variant of the Sensible Machine to perform what was learned.

(Task 4.1) Collaborating with researchers in quantitative psychology, develop an understanding of the algorithms and 'software' that reside on biological neural wetware, including how these systems learn to identify anomalies, solve new problems, search their memory for previously unconnected information that contributes to a problem solution, and whether to ask for help. Figure out how to use this theory to write algorithms that will address real world problems using the devices and architecture from other parts of the project. Note that this will not be today's machine learning, because the machine is different. Figure out the performance of these computers on real world problems, as the new neural algorithms can be expected to scale differently on the new architectures

4. Conclusions

We outlined in this white paper a technical plan for a ‘Sensible Machines’ Grand Challenge, or a project to create computers that learn. This will require advances in both the functional design of computers and more power efficient technology to compensate for the increased compute load of learning.

Sensible Machines would be of value to society in two ways, first by increasing the scope of applications suitable for computers to include activities that involve learning, and also by creating an ongoing growth path for computing.

Sensible Machines are proposed as general-purpose computers and not artificial brains. If Sensible Machines are truly general-purpose, it ought to be possible to run any and all existing software on them. As a starting point for discussion, the white paper presented a scenario of the logic diagram for a microprocessor being trained into a neural network and then operated. The microprocessor could then run any and all software, demonstrating generality. Such a neural network would not only contain the logic originally designed for a regular computer, but many alternative implementations that are natural result of the learning. Deleting information about these alternatives would result in comparable cost and energy efficiency to today’s computers, yet keeping the extra information would allow behavioral adaptation through additional learning.

However, learning requires additional resources. Information about alternative implementations increases memory requirements. Updating the larger memory during incremental learning raises throughput requirements. The need for increased memory strongly suggests 3D manufacturing approaches for memory, yet 2D may be sufficient for processing, provided more power-efficient devices can be found.

This white paper also asserts that our understanding of the thermodynamic limits of computation is incomplete. While theory developed some 55 years ago correctly identified that circuits of that era would have “typically on the order of kT ” minimum energy dissipation, reversible computing, quantum computing, and now perhaps computers that learn have unexpectedly circumvented what was initially proposed as a limit. By updating the calculation based on a carefully designed subsystem representative of the crucial learning circuit in Sensible Machines, the analysis yields a result $100\times$ lower. While the example using a specific circuit is likely not fully generalizable, we only need it to work for a learning subsystem to make Sensible Machines practical and realizable.

Sensible Machines based on the theoretical advances above will require a more power-efficient computational substrate that can only come about through device physics research on state-containing devices with state-dependent nonlinear dynamical behavior, including chaos. This descriptive phrase matches methods used in biology for synapses and spike generation, yet it also enables the crafting of circuits that have lower energy limits. What are these devices? The class does not include transistors, which have no state, or traditional memories, which have little behavior. Some of the new memory devices, such memristors or RRAM have the necessary properties. Random and chaotic behavior is common in biological systems, yet random numbers can be generated with greater power efficiency with these devices than in software.

5. References

[Anderson 13] Anderson, Neal G., Ilke Ercan, and Natesh Ganesh. "Toward nanoprocessor thermodynamics." *Nanotechnology, IEEE Transactions on* 12.6 (2013): 902-909.

[Apple 97] Apple Knowledge Navigator concept video. See https://www.youtube.com/watch?v=QRH8eimU_20

[Dyson 12] Dyson, George. "Turing's Cathedral: The Origins of the Digital Universe." Vintage, 2012.

[Ferrucci 10] Ferrucci, David. "Build Watson: an overview of DeepQA for the Jeopardy! challenge." *Proceedings of the 19th international conference on Parallel architectures and compilation techniques*. ACM, 2010.

[Gleick 87] Gleick, James. "Chaos: Making of a new science." New York: Viking-Penguin (1987).

[Hawkins 11] <http://numenta.com/learn/hierarchical-temporal-memory-white-paper.html>

[Hinton 06] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507.

[IEEE 13] Rebooting Computing Summit Event Summary (RCS 1) <http://rebootingcomputing.ieee.org/rc-summits/rcs1>

[IEEE 15] <http://rebootingcomputing.ieee.org>

[Kavli 13] 9th Kavli Futures Symposium: The Intersection of Nanoscience and Neuroscience, Jan. 2013, <http://www.kavlifoundation.org/neuroscience>

[Killian 93] Killian, Joe, and Hava T. Siegelmann. "On the power of sigmoid neural networks." *Proceedings of the sixth annual conference on Computational learning theory*. ACM, 1993.

[Landauer 61] Landauer, Rolf. "Irreversibility and heat generation in the computing process." *IBM journal of research and development* 5.3 (1961): 183-191.

[Le 13] Le, Quoc V. "Building high-level features using large scale unsupervised learning." *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.

[Mainzer 13] Mainzer, K., and L. Chua. "Local Activity Principle." (2013).

[Moore 65] Moore, Gordon E. "Cramming more components onto integrated circuits, *Electronics*, Volume 38, Number 8, April 19, 1965." Also available online from <ftp://download.intel.com/research/silicon/moorespaper.pdf> (1965)..

[Mountain 15] Mountain, D., Krieger, C., et. al. "Ohmic Weave: Memristor Based Threshold Gate Networks." *Computer*, December 2015.

[NICE 15] <http://www.niceworkshop.org>

[Pickett 13] Pickett, Matthew D., Gilberto Medeiros-Ribeiro, and R. Stanley Williams. "A scalable neuristor built with Mott memristors." *Nature materials* 12.2 (2013): 114-117.

[SRC 15] Semiconductor Industries Association (SIA) and Semiconductor Research Corporation (SRC), "Rebooting the IT Revolution, a Call for Action", (2015)

[Turing 39] Turing, Alan Mathison. "Systems of logic based on ordinals." *Proceedings of the London Mathematical Society* 2.1 (1939): 161-228.

[Walter 05] Walter, Chip. "Kryder's law." *Scientific American* 293.2 (2005): 32-33.

[Widrow 60] Widrow, B. "Adaptive switching circuits." IRE WESCON Convention Record. 1960.

[Wikipedia] http://en.wikipedia.org/wiki/List_of_animals_by_number_of_neurons