1.4 Why Can't A Computer Be More Like A Brain? Or What To Do With All Those Transistors?

Jeff Hawkins

Founder, Numenta, Menlo Park, CA

1. Overview
Why is it so difficult for computers to perform tasks that humans do quickly and easily? For over fifty years, we have tried to program computers to recognize images, understand language, control robots, and learn on their own. Although we have had some limited successes, what is more remarkable is how little progress we have made. Large increases in memory capacity and CPU performance have not resulted in corresponding improvements in performance on AI tasks. Machine intelligence is in need of new approaches.

In humans, the neocortex is responsible for most high-level perception, language, and motor control. The anatomy and physiology of this large sheet of cells have been studied in sufficient detail to constrain how information must be stored and retrieved in neocortical tissue. The neocortex, indeed the entire nervous system, has a hierarchical organization. Knowledge about the world, whether it is visual, auditory, or tactile information, is stored in this hierarchically-organized memory.

We are beginning to understand how hierarchical representations solve many problems in both neuroscience and machine intelligence, and how to build computer systems that perform in a similar way. Such machines are naturally parallel. At a time when the semiconductor industry is looking for ways to utilize multi-core CPUs, a new hierarchical computing paradigm is emerging, one that holds the promise of productively utilizing large amounts of parallel computing power.

2. Introduction
By the age of five, a child can understand spoken language, distinguish a cat from a dog, and play a game of catch. These are three of the many things humans find easy that computers and robots currently cannot do. Despite decades of research, we computer scientists have not figured out how to do basic tasks of perception and robotics with a computer.

Our few successes at building "intelligent" machines are notable equally for what they can and cannot do. Computers, at long last, can play winning chess. But, the program that can beat the world champion can't talk about chess, let alone learn backgammon. At best, today's programs solve specific problems. Where humans have broad and flexible capabilities, computers do not.

Perhaps we've been going about it in the wrong way: For 50 years, computer scientists have been trying to make computers intelligent while mostly ignoring the one thing that is intelligent: the human brain. Even so-called neural-network techniques take as their starting point a highly-simplistic view of how the brain operates.

It is clear to many people that the brain must work in ways that are different from digital computers. To build intelligent machines, then, why not understand how the brain works, and then ask how we can replicate it? My colleagues and I have been pursuing that approach for several years. We've focused on the brain's neocortex, and we have made significant progress in understanding how it works. We call our theory "Hierarchical Temporal Memory", or HTM.

3. What Neuroscience Tells Us About Intelligent Machines
For over 100 years, neuroscientists have been collecting details about how our brains are structured. For the most part, this research has been done without a strong theoretical framework, so we know a lot about how the brain is put together, but not much about how it works.

From the perspective of building intelligent machines, the neocortex is the part of the brain of most interest. It is a sheet of roughly 30 billion neurons, 3mm thick, and 1,000 cm^2 in area. It wraps around and covers most of the rest of the brain, and it constitutes about 60% of the volume of a human brain. The neocortex is the locus of almost all high-level perception, thought, planning, language, and motor control. It is the location of almost all declarative memory, the knowledge you have about the world that you can recall and state. This knowledge was not in your brain at birth, it had to be learned. Throughout your life, and especially during youth, the cortex is learning, building a model of the world. The key question is how does the neocortex build and store a model of the world? If we can answer that question, then we are well on our way to building machines that can do what the neocortex does.

The neocortical sheet is divided into dozens of regions, each of which plays a different role in cognition. Some regions deal with vision, others with audition, others are responsible for language, etc. A remarkable fact is that although the cortical regions are functionally differentiated, structurally they look almost identical. In 1979, physiologist Vernon Mountcastle proposed that each region of the cortex performs the same function. What makes a visual area visual and a motor area motor, is due solely to what that region is connected to [1]. This was an incredible insight. It meant that we could look for a common algorithm underlying all the things the cortex does. The cortex is not one hundred algorithms solving one hundred problems, but one algorithm applied to many problems. Today, there is a large body of evidence supporting this idea of a common cortical algorithm.

To understand what this neocortical algorithm might be, we need to look at the structure of the neocortex itself. The functionally-differentiated regions of the neocortex are connected by bundles of nerve fibers. A map of the connections between regions has been worked out for a few mammals. Figure 1.4.1 shows a well-known connectivity diagram for the Macaque monkey's neocortex. The boxes in the drawing are regions of neocortex: The lines are bundles of nerve fibers going between regions. (Don't be intimidated by the complexity of this diagram. All mammals have a neocortex; They vary in size and connectivity from species to species, but they all work. We are looking for principles that apply to all neocortexes, not just Macaque, therefore the particular connectivity shown in Fig. 1.4.1 is not important for this discussion.) The most important aspect of the structure of the neocortex is visible in Fig. 1.4.1; It is organized as a hierarchy.

Sensory information enters at the bottom, and ascends from region to region. In regions at the bottom of the hierarchy, cells respond to small patterns in the world at specific locations in the sensory space, for example, a vertical edge in one part of the visual field. As we ascend the hierarchy, cells respond to more and more complex objects, and are less specific about where the object appears. At the top of the visual hierarchy, for example, you will find cells that respond to complex objects, such as specific people, regardless of how and where the person appears. The same principle applies to other sensory inputs such as touch and audition.

Figure 1.4.2 shows the same regions as depicted in Fig. 1.4.1, but now the boxes are drawn to represent the relative area of each region [2]. In addition, the main visual pathway is highlighted. We can learn several things from this figure. One is that sophisticated visual- object recognition only requires a hierarchy six-levels high. Recognition can occur so fast that information only has time to travel through a few regions before the brain knows what it is. This tells us that each region is not doing a lot of processing. Neurons are slow, and there just isn't enough time for them to do much before significant recognition occurs. Another conclusion is that the visual areas at the bottom of the hierarchy are relatively

 978-1-4244-2011-7/08/$25.00

large. In the case of the Macaque monkey, the first two levels, together, represent about 25% of the entire area of the neocortex. If all areas of the neocortex are doing basically the same thing, we therefore know that each and every region is storing some part of our knowledge of the world. The size of each region is likely to reflect the amount of memory dedicated to a function, therefore much of what our neocortex knows about how things look, is low-level information. To know what a dog looks like, requires a lot of low-level visual knowledge, and not as much high-level visual knowledge.

Although neuroscientists refer to each box in Fig. 1.4.2 as a region of neocortex, this is somewhat misleading: Each region contains billions of cells and the cells, that project from one region to another do so with a limited fanout and in a particular topological fashion. A more- accurate depiction of a neocortical hierarchy is shown in Fig. 1.4.3. The neocortex is shown as more like a tree-shaped network than a flowchart.

What does each area of the neocortex do? Very briefly, I believe that every region of the neocortex is learning common sequences of input, common temporal patterns, similar to learning melodies. As input streams-in from below, an area of cortex learns what the common sequences are (like melodies), and it passes a stable pattern representing the sequence (like the name of a melody) to the region above. Another way to describe this is to say that, as we ascend the hierarchy, a region groups multiple input patterns into a single output.

As information ascends the hierarchy, it becomes temporally more stable. As information descends the hierarchy (there are more fibers projecting down the hierarchy than up), sequences unfold in time over and over. A high-level stable concept can create a complex time-varying motor behavior (or prediction) at the bottom of the hierarchy.

Time plays a critical role in instructing each region what patterns belong together, that is, which patterns have a common cause in the world, and should be grouped together. If a sequence of notes occurs over and over again, it is likely it represents a single thing, and should be grouped together under a common name. This basic principle applies to all sensory inputs. Time acts as a teacher to determine what input patterns should be grouped together.

There is much more that we know about the structure of the neocortex, and how it relates to the theory of HTM. The point that I am making here is that the brain's physical structure acts as a constraint on how information must be stored in a brain. If we want to build machines that perform like a brain, we should start by emulating the brain's information-storage paradigm. The most important part of this storage paradigm is its hierarchical structure. Our knowledge of the world is distributed in a hierarchical memory system which is fundamentally different than the linear memory we use in computers.

Hierarchical representations solve many problems that have affected machine learning in the past. The most important fact is that they require dramatically less memory and less training time, primarily because representations at the bottom of the hierarchy are shared among higher regions. A more detailed discussion of the theory of Hierarchical Temporal Memory can be found at www.Numenta.com.

4. Building Machines That Work Like the Neocortex

A company I co-founded in 2004, called Numenta, has built a software platform for exploring and deploying hierarchical memory systems that mimic those found in the brain. These systems are comprised of many memory nodes arranged in a tree-shaped hierarchical network similar to that shown in Figure 3. Numenta's software toolset is freely available, and researchers are currently applying it to a diverse set of problems.

In addition to building a general-purpose toolset, Numenta is working on applying the technology to machine-vision tasks. Although the tools and algorithms are not yet mature, and the memory systems we are building are simple compared to what are found in the human brain, we are encouraged by our early progress. Figures 1.4.4 and 1.4.5 show the results of two stages of our vision-system development. The first stage recognizes 48 different line figures [3]. Although not designed to be otherwise useful, this is a difficult problem, one that we believe would be unsolvable using other methods. You can download a Windows-based front end to this system and experiment with it yourself by drawing figures and seeing how it performs. It can handle large amounts of noise, distortion, and translations. It is fun to experiment with and I believe you will find its performance impressive. The full source code to this application is available on Numenta's Website. Figure 1.4.5 shows early results from a more-sophisticated vision system under development. This one uses higher-resolution gray-scale images. In both examples, the system is trained by exposing a hierarchical-memory system to moving images. Showing the images moving through time is essential for training.

Numenta is building a community of researchers exploring HTM technology, and we welcome anyone who wants to contribute.

5. Performance: Hardware and Software Considerations

Of particular interest to the circuit and system designer may concern the implications of hierarchical-memory systems for future CPU- and memory-chip designs. If HTMs represent a major new machine-intelligence paradigm, then what might we expect the role of silicon to play in the growth of this field?

Today, our HTM systems are implemented completely in software. Because HTMs comprise many nodes performing a similar function, they are naturally parallel and easily divided among multiple CPUs. Our current implementation will run on single-core CPUs, multi-core CPUs, and large computer clusters. HTM software readily takes advantage of multi-core-processor architectures, and adding more cores to CPUs will benefit our work in HTMs. As useful as this is, most of the performance improvements we have made to date are a result of software changes.

For example, through software tuning, alone, we have significantly improved the time it takes to train an HTM; and to do inference. The simple vision system shown in Fi. 1.4.4, at first took several days to train, but now it takes less than 15 minutes. Recognizing a novel image went from a second to only a few milliseconds. However, the larger and more-sophisticated system of Fig. 1.4.5 currently takes two days to train. This is too slow, and we expect that we can speed it up significantly with improved software.

Could custom hardware help as well? Yes, but because HTMs are in a phase of experimentation and rapid change, we don't believe now is a good time to design custom hardware to accelerate the effort. By the time a custom circuit is perfected, it would likely be obsolete, because HTM technology itself is not yet stable. For the moment, we are relying on the flexibility and rapidity of software change, as we test and evolve the HTM technology.

But what about the future? If we are able to overcome the theoretical obstacles before us, and successfully apply HTM technology to many real-world problems, then it will make sense to design custom silicon to reduce cost and improve performance. But, it is too early to know what these circuit designs will look like; Yet, we can speculate that as with current computers, the two main opportunities will lie in memory and CPU design.

HTMs consist of a hierarchy of memory nodes. The computational burden of each node is not great, but in order to train an HTM, it can require exposure to many training patterns. Thus, although each iteration is quick, overall training time can be an obstacle. As our systems get larger, this becomes more critical. Each node has a chunk of memory associated with it. The data stored in a node consists primarily of large sparse matrices. The computations performed by a node are typically vector and matrix operations on these large sparse matrices. Many of the software-performance improvements that we have made to date have come from taking advantage of the sparsity and other unique attributes of the data. The same idea could apply to custom silicon.

I can imagine a time in the future where we create memory and CPU chips that are specifically designed for the type of data HTMs require. It might even be possible to embed some of the matrix and vector operations into the memory chips themselves. Thus, instead of retrieving a matrix from memory, and multiplying it by a vector, we could instead pass the vector directly to the memory chip and the memory chip itself would perform the operation, and return the result. There are many possibilities, but as I said, we believe it is a little too early to pursue these hardware options today. It may take a year, or several years, before HTM theory settles enough to warrant custom hardware.

If the reader is interested in learning more about HTMs, and working with them, I suggest looking at the material on Numenta's Website. There, you will find documentation, videos, papers, tools, source code, and discussion forums. As well, there are people around the world currently using these tools. But as in any new endeavor, it is hard to know what obstacles we will encounter, and where the first major successes will come from.

6. Summary

By using neuroanatomy and neurophysiology as a set of constraints, we believe that we have started to uncover how the brain uses hierarchy and time to create a model of the world, and to recognize novel patterns as part of that model. Hierarchically-organized memory is fundamentally different than the linear memory used in current computers, and therefore offers the potential for new computer architectures. Today, we are exploring and advancing this technology by using traditional computer architectures (benefited by multiple CPU cores) to emulate the hierarchical structure of the neocortex.

The age of intelligent machines may be at hand, and if so, there will be many opportunities to rethink how integrated circuits and systems can play a leading role.

References:

[1] V. B. Mountcastle, "An Organizing Principle for Cerebral Funcition: The Unit Model and the Distributed System", G.M. Edelman and V. B. Mountcastle, eds, *The Mindful Brain*, MIT Press, 1978.
[2] D. J. Felleman and D. C. Van Essen, "Distributed Hierarchical Processing in the Primate Cerebral Cortex", *Cerebral Cortex*, V1 N1, Jan/Feb 1991.
[3] D. George and J. Hawkins, "A hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex," *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2005.

Please click on paper title to view a Visual Supplement.

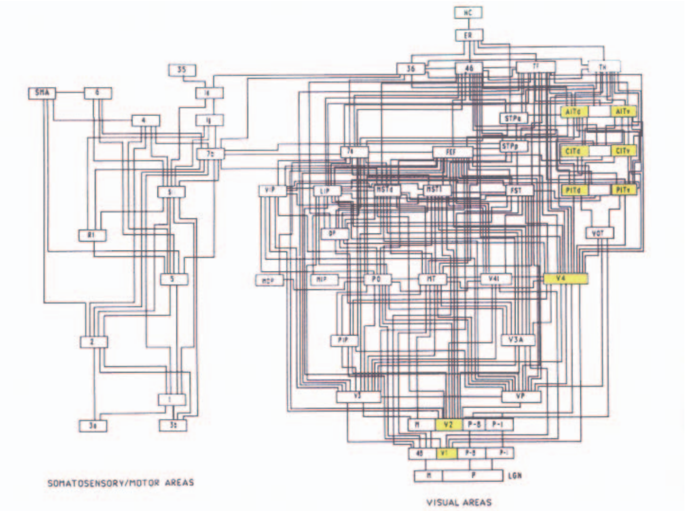

Figure 1.4.1: Hierarchy of neocortical regions in a Macaque monkey. Highlighted regions are the main visual pathway used in recognizing objects [3].

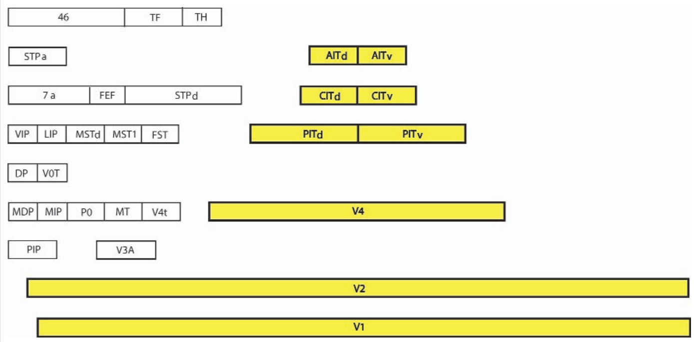

Figure 1.4.2: Visual regions from Fig. 1.4.1, now drawn showing relative size. Most memory is allocated to low-level visual knowledge.

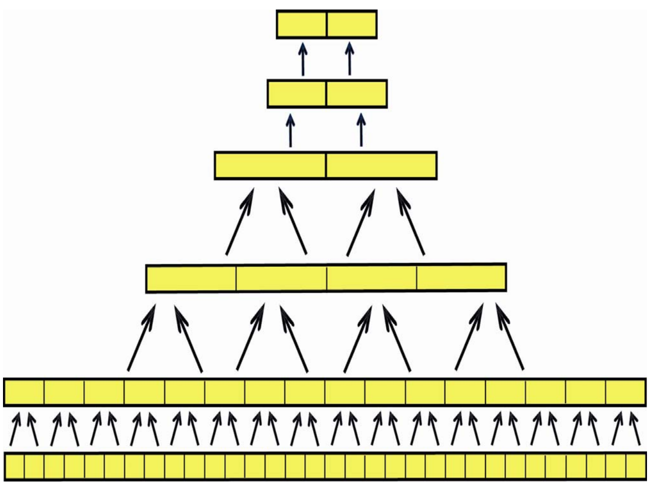

Figure 1.4.3: Highlighted regions from Fig. 1.4.2. Connections between neurons indicates that cortical hierarchy is tree-shaped.

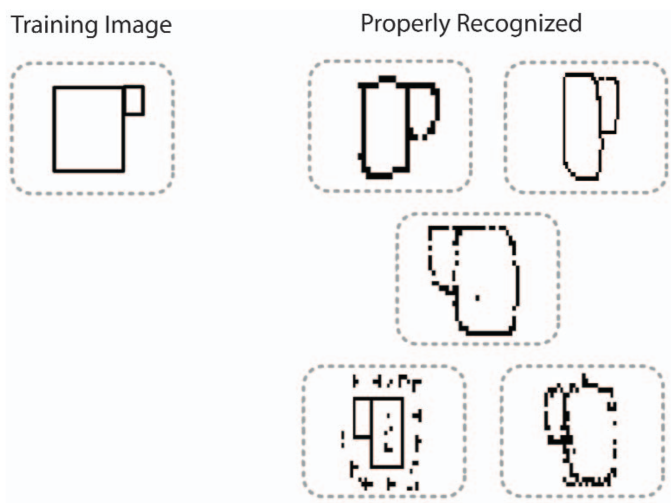

Figure 1.4.4: Results from a simple hierarchical vision system trained on 48 objects. Shown are one training sample and some of the novel patterns it classified as the same.

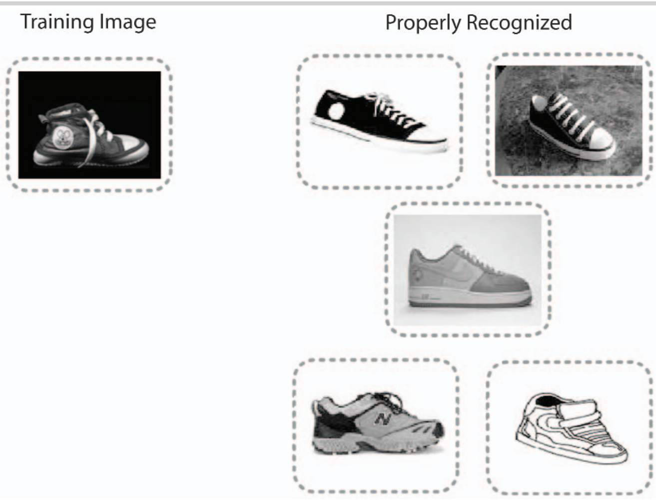

Figure 1.4.5: Results from a more-sophisticated vision system under development. As with Fig. 1.4.4, training is accomplished by showing moving images, an example of one of which is shown on the left. Also shown are novel images that were classified the same as the training image.