22 Computation and Cognition: Four distinctions and their implications

W. Tecumseh Fitch
University of St. Andrews, Scotland

Introduction

In this paper I discuss four computational distinctions at the heart of natural computation, and thus relevant to the central and most interesting question of cognitive science: "how the brain computes the mind". I assume that we can think of cognition as a form of computation, implemented by the tissues of the nervous system, and that the unification of high-level computational theories of cognitive function with detailed, local-level understanding of synapses and neurons is the core goal of cognitive (neuro)science. Thus I am concerned here with how the brain *computes* the mind, following Alan Turing's seminal gambit (Turing, 1950), and much of subsequent cognitive science, in thinking that intelligence is a kind of computation performed by the brain. By thus asserting that the brain is a kind of computer, I must immediately clarify that the natural computations performed by the brain differ dramatically from those implemented by modern digital computers (Richards 1988). Computation (the acquisition, processing and transformation of information) is a more general process than the serial, binary computation performed by common digital computers. From this viewpoint, the assertion that the brain is a kind of computer is a mild one. It amounts to nothing but the everyday assumption that the brain is an organ responsible for acquiring, remembering, processing and evaluating sensory stimuli, and using the knowledge thus acquired to plan and generate appropriate action.

Maintaining this more abstract concept of computation is critical, because an overly literal application of the concepts of contemporary serial computer technology, such as the hardware/software distinction, can be deeply misleading. In the brain, memories and plans are stored by modifying its physical form and connections ("hardware") continually. The crucial distinction is between a neuron's morphology, through which it stores relevant aspects of past experience, and its current activation, through which it participates in the myriad natural computations the brain is performing at any moment in time. Nor can neural computation be adequately captured by current connectionist simulations. Despite the value of each of these as metaphors, neither is adequate as a model of the vertebrate brain. One goal of this paper is to make clear why. The other goal is more prospective and thus inevitably more speculative. I will introduce four well-established distinctions in computation, and then explore their implications for some critical unsolved problems in cognitive science (neural coding, consciousness, meaning and language evolution), hoping to point the way towards some promising paths to solving them, and thus the central question of cognitive science.

I will discuss "the" brain, but little of my discussion will be limited to the human brain. The vertebrate nervous system is a conservative structure (relative to the respiratory system, for example). Indeed, most basic aspects of cellular neurophysiology and neuronal morphology are common to all animals from worms to mammals. Among vertebrates, the basic groundplan of the brain is common to all vertebrates, from fish to birds and primates, including myriad specific details such as the nuclei and paths of the cranial nerves, or the connectivity and function of pain, thirst or pleasure pathways. There are no neurotransmitters found in humans that are not also found in fish, and no novel neuronal or tissue types in humans not also found in a cat. The key innovations of mammals - an expanded olfactory system and a layered neocortex - are also found in a dog, mouse or any other mammal (Krubitzer, 1995). Furthermore, from a cognitive perspective, all of the basic components of the mind, such as those underlying the senses, motor control and memory, and cognitive states such as sleep, attention, pain, pleasure, fear or anticipation, are shared with other vertebrates.

Thus, when I refer to "the brain" I mean the vertebrate brain in general. Nonetheless, the human mind clearly differs, in qualitative ways, from that of other animals, and a satisfactory neural theory of the mind must explain why. No "magic bullet" (novel neurotransmitter, neuronal morphology, or tissue type) appears to account for these differences. Brain size alone is inadequate to explain them: an elephant, dolphin or whale brain is larger than a human's, but these animals do not have language, complex technology or elaborate cultural and ethical systems like ours. The safest assumption at present is that some relatively subtle aspects of the larger-scale organization of the human brain differ from other vertebrates, probably in a way influenced by but not reducible to brain size or brain/body ratio, and that these organizational novelties underlie the qualitative computational novelties of our species (Deacon, 1997). Understanding these differences is another core problem of cognitive neuroscience.

Four Key Computational Distinctions

Analog vs. Digital

Perhaps the most fundamental distinction in computation is the analog/digital distinction, because it maps onto the fundamental distinction in mathematics between the discrete integers and the continuous real numbers. Although virtually every device termed a "computer" in contemporary parlance is a digital computer, many simple control systems surrounding us are actually analog computers: thermostats, lightbulb dimmers, spark plug distributors or other engine control systems, and many others. These systems have in common their simplicity (the dimmer switch is equivalent to a single "multiply" operation) and their specialization: each is devoted to performing a single restricted type of function. In contrast, the general-purpose digital computer instantiated by the central processing unit (CPU) of a computer is extremely complex (with millions of transistor switches) and general purpose: it is equally well-suited for spell-checking, filtering sounds, adjusting the contrast of a photo or compositing video images. It achieves this flexibility by having a small set of abstract, powerful operations (add, multiply, AND, OR, branch operations and the like) which can be combined into more complex programs to perform virtually any computation conceivable (given a clever enough programmer). The price paid for this flexibility is that the digital computer must always work with discrete values: it must subdivide the continuous world (where any value is possible) into a series of integers where only a finite number of pre-chosen values can be represented. However, though a digital system can only represent a limited number of values, with adequate memory we can chose an arbitrarily large number of these to suit our needs.

In the early days of computing, both analog and digital computers were common, and some fairly sophisticated analog computers were widely used (e.g. analog computers that could solve arbitrary second order differential equations). In those days, memory was very limited and expensive, and the virtues of analog computers were widely recognized. Certain difficulties raised by digital computing were problematic, but the invention of the von Neumann architecture (where data and program are both stored in digital memory to allow virtually any calculation to be performed by a single machine), combined with ever larger and cheaper memory, made the eventual triumph of general purpose digital computing inevitable. An excellent description of the virtues and failings of both styles of computing is von Neumann (1958).

How does the analog/digital distinction apply to the brain? A neuron either fires an action potential, or doesn't, and all its action potentials are essentially equivalent. Thus the output of the neuron can be represented by a single bit, which at any moment in time is either zero or one. The brain can (apparently) represent the world and solve all the problems it does with this digital *lingua franca*. However, this is not the whole story. Although the *output* of a neuron is digital, its *inputs* come in the form of analog graded potentials, and the computations a neuron performs by integrating all of these thousands of inputs are also, for the most part, analog. John von Neumann was aware of this, and his idealization of the brain as a digital computer was thus an educated gambit: How far can we get if we abstract away from the analog aspects of neuronal computation? The answer, if the ever-increasing power of modern, general purpose digital computers is any guide, is "very far indeed". Nonetheless, despite its practical success, it seems clear today that von Neumann's gambit was ultimately unsuccessful as a model of the brain. Problems that are trivially easy for even a simple computer (e.g. dividing two 16 digit numbers) are very difficult for an unaided human. But contemporary von Neumann computers fail at the very tasks the brain excels at: problems easily solved by an infant or a fish (e.g. distinguishing figure from ground) are very difficult for computers. And problems solved by every normal child before the age of five, like deriving the meaning of sentences, still seem hopelessly difficult computationally, despite a half-century of programming effort.

A promising attempt to resuscitate the analog component of neural computation, which I will call the connectionist gambit, has as one core insight the fact that over a longer time scale, the output of a neuron can be seen as continuous, in the sense that the number of spikes per second (the neuron's *activation level*) approximates a continuous value (e.g. between 0 and 1.0). Despite some significant advantages (especially in the realms of nicely handling noisy, distorted or incomplete input) neural nets also turn out to be limited in some critical ways, and have not fulfilled their initial apparent promise (Marcus, 2001). It now seems clear, after five hard-working, well-funded decades of AI and two of connectionism, that neither of these approaches alone is adequate to solve the key problems of neural computation and cognition. Perception and motor control at the level of a fish is still well beyond state-of-the-art robots, and performing vision or language at human levels is a programmer's dream far beyond the reach of existing architectures.

Recent discoveries in neurophysiology suggest a reason for these practical failures. Both gambits agreed on taking the neuron as the key unit of computation. While one approach idealized the neuron as digital and the other as analog, both represented it by a single number (either integer or real). In fact, each individual neuron is a hybrid analog/digital computing machine. Its myriad analog inputs, coming from the synapses that join it to hundreds or thousands of other neurons, are combined and transformed into an all-or-none digital action potential, and apparently both sides of this hybrid computational system are important (Häusser & Mel 2003, Debanne 2004). Exciting recent discoveries in cellular neurophysiology indicate that the voltage-gated channels that trigger action potentials exist in the dendritic trees of some neurons, and play a critical role in the computation that the neuron performs (Wei et al., 2001). Thus, representing a neuron as a simple blob, with multiple synapses attached to it and gated by some threshold function, is inadequate. A typical neuron, with its tree-like form, instantiates a minicomputer irreducible to a single number, whether real or discrete. The future of computational neuroscience lies in systems which have as their core computational primitives less-idealized neurons: hybrid analog-digital devices that transform their data in complex ways.

Serial vs. Parallel

Perhaps the most striking distinction that arises when comparing modern digital computers with the brain is between serial and parallel processing. A kitchen analogy (Churchland, 1995) makes the difference clear. When trimming carrots, you can either trim each one individually (serial operation) or can line them up on the cutting board, and with one knife stroke trim them all at once (parallel operation). The brain is a massively parallel processing system: millions of computations are occurring simultaneously across the cortical surface at any given time. Although a typical laptop has some degree of such independence (e.g. the VLSI chip dedicated to displaying graphics is largely independent of the CPU), the CPU itself is a serial machine: it does one thing at a time. The reason it can do so much is because each operation is executed very rapidly: the 1 Gigahertz machine I'm typing on can perform one billion (10^9) serial mathematical operations per second.

Historically, mathematics has developed along serial lines. The operations and algorithms used in human calculation or theorem proving are virtually always implemented one step at a time. Thus, it was natural when designing and programming early computers to use serial algorithms. However, it was already clear to von Neumann that this is not the way the brain does things. Neurons are sloppy, unreliable, and slow compared to transistors. An average "clock speed" in the cortex is 100 Hz or 100 operations per second (this is both roughly the transit time between cortical layers and a high-end firing rate for typical cells; the auditory system runs faster, e.g. 1000 Hz). A brain process that had to undergo 1000 serial operations with neurons would take 10 seconds, and accumulate so much error that the end result might well be useless. Instead, even very complex neural operations like recognizing a face happen in around a quarter of a second. The brain accomplishes this by performing the millions of operations involved in parallel, with only a few serial steps (say five from ganglion cells in the retina to the fusiform gyrus of cortex). Von Neumann termed this dimension "logical depth": a serial digital computer is suited by its speed and accuracy to deep algorithms involving many steps, while the brain is limited by its biological components to a shallow few.

Why don't digital computers do their operations in parallel, achieving even greater speed? Some do: parallel processing machines (e.g. the Connection Machine) exist but have never fared well, due to the difficulty of programming such machines. The speed attainable in theory is rarely attained in practice because of various annoying practical issues (e.g. many of the parallel processors end up idly waiting for some other computation to finish). Newer parallel systems (e.g. Beowulf) are just getting started and their promise is hard to evaluate at present. Thus, except for a few specialized problems, serial computers rule the silicon world, while in the biological world, parallel systems are king.

The virtues of parallel processing are well known, and already thoroughly catalogued by connectionists (an enthusiastic and accessible introduction (Churchland, 1995). Parallel processing algorithms tend to be robust in the face of noisy input, have an ability to generalize over multiple exemplars, can complete basic patterns and are resistant to losses of computational units (real neurons die unpredictably). All of these are brain-like skills that serial algorithms fare poorly on, and were the primary reason for the enthusiasm in cognitive science in the 1990s for connectionist approaches. Unfortunately, the parallel architectures currently available have important limitations, also well-catalogued: they are slow to learn and their ability to induce general rules or abstract away from known contexts is limited (Marcus, 2001). However, these cannot be limitations of parallel systems in principle (the brain *is* a parallel system, and avoids these problems), but of currently available algorithms and architectures. There is good reason to hope that computer scientists can improve upon the current situation. A new wave of more brainlike parallel processing is beginning, in which hundreds or even thousands of individual microcomputers are networked together via fast intranet, and the promise of this approach is only beginning to be explored.

As this new wave of parallel architectures progresses, it will be important to recognize two distinct levels of parallel processing in the brain. First, at the level of an individual neuron, is the integration of information over synapses. A typical pyramidal cell in cortex has around 1000 separate inputs onto its dendritic arbor and cell body, and the cell's activity at any moment is a complex transformation of these inputs (Häusser & Mel 2003). Each synapse is individually and locally updated, so the cell is a true parallel distributed processor. Furthermore, there is a cell-level economy (based on how often it fires, its uptake of neurotrophins, and myriad other factors) that influences all of its synapses. The limit is in programmed cell death (apoptosis) which plays a critical role in the developing nervous system: when a single cell dies, all of its synapses die with it. However, as far as other cells are concerned, a neuron has a single discrete output: its "decision" to fire (or not), which is distributed via its axonal arbor to all of the cells downstream of it. Thus, there is already massive parallel processing and then compression of information at the single cell level. Further, these two processes interact: the cell-level decisions to fire often propagate back to its synapses, and playing a role in parallel synaptic modification.

The second, global level of parallel processing is at the level of large assemblages of neurons (e.g., all the pyramidal cells in a single region of cortex like V1 or A1). This level, typically highlighted in connectionist discussions of the brain, has thousands of inputs *and* outputs. The end product of a computation at this level is not typically compressed through a single output channel, and thus the effect of the transformation must be "read off" the activity of many neurons. This makes the neurophysiologist's job difficult, since the information present at this global level can only be discerned via multi-unit recordings. Fortunately techniques for acquiring and analyzing multiunit are advancing rapidly. It also complicates the computational theorist's job, since it is the complex, transformed output of the neuronal minicomputer that enters into the more global multicellular parallel processing algorithms. Only at the final output level of the whole nervous system is brain-level parallel processing finally compressed and channeled into the final decisions of motor control and action.

The traditional connectionist metaphor elides this local/global distinction: although connectionist nets are loosely modeled on the higher level (nodes are often called "neurons") they actually better parallel the lower, single cell layer. The adjustable "multiply" units at the heart of a connectionist architecture are computationally equivalent to a single dendritic compartment. Thus a typical connectionist model (with perhaps 1000 such units) is more comparable to a single neuron, than a network of cells. However, because the output of such nets is not typically channeled through a single output, this aspect emulates the global parallel system. The next wave of neural modeling, already well under way and a major current focus of computational neuroscientists, involves more biologically accurate models of single neurons, connected into more realistic networks (increasingly, on parallel systems like Beowulf). Despite the challenges of its complexity, this approach seems to offer hope of solving some of the problems suffered by typical connectionist models.

Summarizing, it is a vast oversimplification to think of the brain simply as a parallel analog machine. Each individual neuron is a complex analog-to-digital converter, processing thousands of synapses of input in parallel and converting them to a single, digital output. The brain is composed of 100 billion such minicomputers, running in parallel, and includes a final output level (of attention, decision and action) that is essentially serial. Progress in understanding the computational problems the brain solves may necessitate models of neural computation that respect this complexity, and are considerably more complicated than those cognitive scientists typically entertain. While this complexity may seem daunting, the apparent failures of both von Neumann's and the connectionist gambit, which have already explored the possibilities and revealed the limitations of the two simplest abstractions, leave us little option. The good news is that the rapidly-advancing field of computational neuroscience is hot on the trail of such models, and neuroscience is generating reams of data to test them empirically. The vast computing power available today with large networks of individual computers means we can implement and test models at this level of complexity without major technological difficulty.

Feedforward vs. Feedback

The next distinction has long been recognized as central in engineering, and goes by several names. Two everyday examples may make this feedforward/feedback distinction intuitive. First, a visual analogy. An image reflected in a tinted mirror, or a rippled lake surface, is transformed: the reflected image is darker, or rippled. These are examples of a feed-forward or "one-pass" transformation: the operation of reflection occurs just once. Feed-forward transformations can be quite simple (like the mirror image) or extremely complex (the time-varying reflections from a wind-rippled lake). If we now artfully arrange two mirrors, one tinted and the other normal, so that the image of one falls upon the other, we suddenly see an infinite receding set of images: mirrors within mirrors within mirrors. A single transformation of "tinting" is suddenly repeated uncountably many times by the simple expedient of having its output reflected back in as a new input. One mirror gives one application of the tinting operation, while two mirrors, when properly arranged, give infinite applications. This is a feed-back or "recurrent" system.

A second analogy is acoustic Facing a cliff in the middle of an open space, you clap your hands and hear a single echo. The sound from your hands travels to the wall, and is reflected back to your ears a bit later, in subtly changed form (usually with high frequencies removed). But if you are standing between two buildings (or in a canyon) and clap your hands, you hear an endless series of echoes, of gradually decreasing intensity ("reverberation"). Like the two mirrors, a reverberant space could theoretically "ring" forever, echoing till the end of time. Practically, of course, losses render the reverberant sound inaudible rather quickly (20 s or so), but we can easily build a system that adds a bit of energy with each pass, say with a microphone, amplifier and speaker. Point the microphone towards the speaker, snap your fingers, and the multiply-transformed sound will hold steady or (more likely) swell to an unpleasant screech. Such "feedback", familiar to any concert-goer, is usually an annoyance, but the same principle can be used to musical effect when properly controlled (e.g. by Jimi Hendrix).

While feed-forward systems have only a single shot at their input, feedback systems are contrived in such a way that their last output becomes their next input, and so have (in principle) infinite opportunities to apply whatever transformation they embody. Any feedback system thus has a feed-forward system at its heart, but differs by including some additional way to rechannel some output back into its input. Engineers characterize the transformation performed by a signal processing system as its "impulse response". Because of the theoretically infinite nature of feedback systems they are called "infinite impulse response", or IIR, systems. In contrast, the output of a feed-forward system, fed a finite signal, is itself finite. Such systems are termed FIR (finite impulse response). Because of the importance of this distinction in signal processing, especially filter design, engineers have fully explored, and mathematically formalized, the advantages and disadvantages of each class of system (e.g., Oppenheim & Schafer, 1989)

The fundamental advantage of feedforward systems is that they are fast, straightforward to understand, and can preserve timing details (they don't distort phase). The fundamental advantage of feedback systems is power: they can do a lot with a rather limited transformation. An engineer building a filter with five multiply operations can do some practically useful things in an IIR filter,. In contrast, a five-multiply FIR filter will, because of its one-pass nature, have trivial power compared to its IIR equivalent. This power does not come without a price: feedback systems fed a complex signal always distort phase, or disrupt timing, and furthermore are difficult to understand in all but the simplest cases. Worse, feedback systems can generate uncontrolled runaway behavior (like the annoying sound system feedback mentioned previously).

Flat vs. Hierarchical Structure

The last distinction I will discuss is critical in modern linguistics, where it was first formalized, but perhaps less clearly recognized in other branches of cognitive science. This is the distinction between what I will call flat and hierarchical structure (the linguist's version, following (Chomsky, 1957), is between finite-state grammars and phrase structure grammars). In flat structures, all of the elements have equal status: a list of words ("juice coffee milk carrots") or numbers (6177769541) has no organization beyond the serial order of its elements. However, as soon the number list is rewritten as a US phone number, (617) 776-9541, a level of organization above this serial order is evident (the first three numbers are the area code for Boston, etc.) Alternatively, if the structure was +61 77 769 541, the first two numbers could be the country code for Australia. The two strings, although sharing the same sequence of digits, have different hierarchical structures. Repeated application of branching algorithms quite naturally generates such tree-like structures, and computer science is full of tree-building and -parsing algorithms (decision trees, search trees, suffix trees, etc., see Skiena, (1998)). Such trees also appear in motor control and in phonology: a word is a higher-order structure made up of its phonemic and syllabic components.

"Chunking" into higher order, abstract components is pronounced in language: a sentence like "I'll trade you some juice and coffee for your milk and carrots" has a complex hierarchical grammatical structure beyond the order of the words. The appreciation of this structure was foundational for modern linguistics because it invalidated behaviorist approaches which sought to portray language as a serial sequencing operation. This point, made forcibly by Chomsky, (1959) in his critique of Skinner, was foundational for modern linguistics, and all of cognitive science (see Jackendoff, (2002) for a more detailed exposition). Phrase structure seems to be critical not just for human language but many other aspects of our cognition: music, mathematics and social reasoning all involve hierarchical structures. Humans both produce and process hierarchically-structured stimuli, and actively prefer such stimuli (Morgan, et al. 1989). What makes this proclivity striking is the lack of evidence for such abilities in nonhuman primates. For example, monkeys appear to hear a melody as just a sequence of notes, rather than a coordinated, interrelated system of related pitches (D'Amato, 1988; Wright, et al. 2000). Monkeys exposed to auditory output from a finite state grammar, with only flat structure, easily learn it, spontaneously generalizing to novel grammatical stimuli, but fail to do so when exposed to a carefully-matched phrase-structure grammar (Fitch & Hauser, 2004). Although too few species have been examined to reach any broad conclusions, and apes may have greater abilities to generate hierarchical structures in the motor domain (e.g., Greenfield, 1991, Byrne & Russon 1998), hierarchical processing does not seem as widespread in animals as in humans. Thus, unlike the first three distinctions, which are equally relevant to neural computation in all vertebrates, the well-developed hierarchical abilities observable in humans may reflect a computational distinction implemented preferentially in the human brain.

Hierarchicality has several distinct meanings, and a key distinction in language is between recurrent and recursive systems. Any feedback system with loops in it (e.g. all of neocortex) is recurrent, even if only a small or very simple component of its output loops back into its output. Recursive systems, although similar, are more restricted and powerful: the entire complex last output can serve as the next input. This difference is moot for very simple multiply operations but quite relevant in language, where the output of a syntactic operation may be a complex phrase-structure tree. When this can feed back to the input, and thus serve as the starting point for a more elaborate tree, true recursion in the linguistic sense results. This naturally generates sentences such as "I know that you want me to think that you are happy" or "John thinks that Mary believes that Hans wants …" and the like, with no obvious upper bound to the number of embedded clauses. Although phone numbers have some hierarchical structure, you cannot embed one phone number into a second and expect to produce another valid phone number, any more than you can embed one syllable into another to get a new valid syllable. Recursive hierarchicality probably does not apply to phonology.

Importantly, structure-preserving recursion allows the creation of long complex phrases from a few simple rules, exhibiting the power typical characteristic of any recurrent system at a more sophisticated level. A fully recursive tree building algorithm, that takes a structured output and passes that entire structure back in as input, is quite demanding computationally. The apparatus to support recursion in computer science (typically a "stack" which preserves intermediate function calls and their results) is complex, and is not implemented in all computer languages. There are only two aspects of language that appear to support fully recursive hierarchicality: syntax and semantics. This dual ability can be illustrated by a single example, in the sequence:

1. Bob likes Mary.
2. John suspects that Bob likes Mary.
3. Susan realizes that John suspects that Bob likes Mary. (etc...)

Each sentence is built, recursively, from its predecessor, by simply adding a new agent and mental action (semantically speaking) or subject and verb (syntactically speaking). The structure of "Bob likes Mary" is still contained in either of its more complex successors. This embedding process can go on indefinitely: there is no limit built in to the generative process (although there are clearly limits on memory and comprehension of the output sentence). The ability to "embed" mental states within other mental states, at the semantic level, and phrases within phrases, at the syntactic level, constitutes fully recursive hierarchicality. This difference, best exemplified by language, has important computational implications: recursive algorithms require a structure-preserving feedback mechanism which ordinary recurrent algorithms do not.

Implications for Cognition and Language

I will now explore some broader implications of these distinctions, in no particular order, starting with an important implication of the feedforward/feedback distinction for the role of time in neural computation. In some feedforward systems (e.g. a delay line), time is spread out over space. A neural example is in the cochlea: because it takes time for an acoustic signal to travel up the basilar membrane, neurons at different locations correspond to different arrival times. By contrast, in a feedback system, e.g. in auditory cortex, time is not "laid out" in space. A feedback-style IIR filter still processes information in the time domain, but does so by folding its output back into input. This fact makes IIR systems harder to understand, but as already pointed out, is a source of their power as well. Many computational neuroscientists and cognitive scientists believe that temporal processing (as opposed to spatial processing of the type familiar in the visual system) plays an important but still poorly understood role in neural computation. Thus there is an important distinction between feedforward systems like the cochlea and feedback systems like the cortex. Recurrent loops like those in A1 can do powerful temporal processing with no need for delay lines, and it would be premature to conclude that such temporal processing is insignificant in auditory computation (Shamma, 2001). On the contrary, given that all neocortical areas are rich in recurrent connections, it seems more likely that models of the visual system focusing solely on spatially-distributed processing are oversimplified. It would be odd if the computational power intrinsic to such feedback systems were not utilized in neural computation.

<u>The Price of Feedback</u>: A second implication of the feedforward/feedback distinction for natural computation is profound (Braitenberg, 1977). Given a single neuron performing some transformation, all we need is to place that neuron in a loop to raise its operation to the nth power. This is easily accomplished by looping part of its axonal arbor back into its dendritic arbor or having a downstream region project back to the upstream region feeding it. Such recurrent loops are a fundamental characteristic of mammalian neocortex. Despite the

tendency to think of information flow in the brain as being one-way, all layers of cortex are heavily back-connected to the regions "before" them in this chain (including their thalamic inputs outside of cortex). Feedback in the brain, as in an amplifier, can get out of control: if inhibition fails to keep excited neurons from overexciting their neighbors in a feedback loop, the entire cortex can blow out of control, and an epileptic seizure is the result. This is an inevitable consequence of the recurrent nature of neocortex: a high price paid for the power of feedback. Interestingly, although the cerebral cortex is basically a feedback system, the cerebellum is almost entirely feedforward. The parallel fibers carrying information through the cerebellum synapse with a Purkinje cell only once, and there are no recurrent loops at all within cerebellar cortex. Because a feedforward system like the cerebellum does not distort phase, it is perfectly suited to computing timing details are crucial to coordinated cognition and action. A price is paid for this phase accuracy as well, however: the cerebellum has as many or more cells then the cerebral cortex, and each Purkinje cell takes about 10,000 synapses (and up to 200,000 synapses, averaging ten times more synapses than an average pyramidal cell in cerebral cortex). These large numbers follow from the information-processing principle already mentioned: more processors must be dedicated to a feedforward system to achieve a desired effect.

<u>Rhythmic Coding</u>: A more speculative implication of the flat vs hierarchical distinction for cognition concerns how the brain can use time to code information. Traditionally, neuroscientists assume that firing rate (a continuous number) codes neuronal activation, an assumption shared by most connectionist models. The long-recognized problem with this idea is that it takes too long to get an accurate reading (e.g. Stein, 1967). A neuron firing at 0.2 Hz (every 5 seconds) might take a minute of continuous reading before an accurate average value was obtained: far too long for most practically useful computations. Because action potentials are expensive, firing at 100 Hz burns a huge amount of energy and simply increasing the firing rate is of limited applicability (one reason the auditory system, with many fast-firing neurons, is one of the most metabolically expensive component of the brain). Thus, while firing rate is undoubtedly an important way the nervous system codes information, its inadequacies as the sole code have been clear for many years, leading theorists to suggest other temporal coding schemes. A different way to code information, one with great computational power and thus considerable theoretical appeal, would be to use hierarchical temporal structure, or rhythm, to code information. In musical parlance,

periodicity or "tempo" is distinct from "rhythm". Any periodic event, recurring at a certain rate, has a tempo. A rhythm is more: a temporal structure *invariant* over changes in tempo. Many different rhythms exist which share the same tempo and the number of events. Thus, a significant amount of bandwidth is left over in a simple rate-coding system (where tempo alone conveys information): specific inter-spike timing regularities could code additional information. We know that, depending on their precise timing, a volley of action potentials can either excite a downstream neuron or not (Rieke et al. 1997). This means that neurons are clearly capable of "recognizing" rhythms (transforming their input differently depending on its temporal structure), suggesting that hierarchical structure in a spike train from a given neuron has a considerably greater potential to code information than if it were organized as simply flat structure, where each spike is equal. Although a search for such hierarchical structure will be complicated by the fact that "neuronal rhythms" might change as firing rate changes (unlike the musician's rhythm, which stays the same), the computational power added by such a coding scheme would render it quite appealing, because it allows more information to be encoded, quickly, with no additional and expensive action potentials.

Consciousness: The serial/parallel distinction has an interesting implication for the function(s) of consciousness. I assume that consciousness is a specific, concrete component of neural functioning, that (like most aspects of neural function such as perception or action) it has both subjective and objective sides. By definition, only the latter is available for scientific study in non-linguistic organisms. There is nothing specific to consciousness about this, nor does it pose a mysterious "hard problem" for neuroscience. For further discussion and defense of this position see (Churchland, 1995; Dennett, 1991). Although there is no doubt that the brain itself is a massively parallel machine, our mental experience is curiously serial. Although we often do several things at once, we typically attend to just one of them. As Dennett has aptly put it, consciousness is like a serial machine running on a parallel architecture. I hypothesize that this serial nature of consciousness is a computational necessity, one that solves an inevitable problem faced by parallel processing systems that can learn. Coherent updating in a parallel system demands a system for credit and blame allocation, so that each of the semi-independent processing units (neurons, or small assemblages thereof) be informed about the final "decision" of the system as a whole. To see why, imagine you are about to engage in some complex novel action (say crossing a dangerous ravine). In parallel, your brain computes, unconsciously and automatically, various possibilities for accomplishing this goal (Dennett & Kinsbourne 1995). Each of these possibilities may be equipotent during the preparatory stage: indeed each might be perfectly good possible solutions to the problem. In the end, however, one must be chosen and implemented: the myriad possibilities must be winnowed down into a single decision. Now comes the problem: the crossing is made, either successfully or not, and the brain needs some way of distinguishing the myriad equiprobable possibilities it just considered, and rejected, from the one actually implemented. It must "brand" the motor program actually chosen to assign credit (or blame, in the case of failure) to the proper neurons, who can then update their synapses appropriately. I suggest that conscious awareness is simply the subjective counterpart of this necessarily serial neural function, a function that must be present if a massively parallel processing system is to learn from its experience.

Meaning. Meaning is a core unsolved problem of cognitive science. The difference between the semantics of music and language provides an interesting contrast for beginning an exploration of the computational structure of meaning. Spoken language and music are both complex, culturally-transmitted, hierarchical systems based on sound. To a first approximation, language is meaningful (in the sense of being capable of conveying an unlimited number of specific propositions), and music is not. However, music does convey *something*, as illustrated by the fact that we can reliably map a piece of music onto non-sonic domains (like dance or emotion). Both systems thus map sound onto something, which is propositional semantic content in the case of language, and something else for music. I suggest that this "something else" can be captured by the analog/digital distinction. In language, sound maps onto discrete, categorical conceptual dimensions, while in music, sound maps onto continuous, analog dimensions. The speed and intensity of notes in music will map onto the speed and intensity of a dancer's movements, but how fast or hard you say the "a" in "cat" has no influence on the categorical meaning of the word. Thus, the basic sound/meaning mapping in language is digital and categorical, while music maps on to the analog and continuous. This is why music is so well suited to linking with the motor movements of dance, or with emotions. Both dance and emotion have an essentially analog, continuous component. The conceptual mapping performed by music seems nicely-captured by Manfred Bierwisch's term "gestural form" (Bierwisch, 1979).

However, this is oversimplified. Some marginal musical styles convey discrete concepts via musical forms (e.g. "tympani roll = thunder" and the like). More importantly, language has a musical, prosodic side. Spoken language manages to get the best of both worlds: using its discrete-to-discrete mapping capability to allow the specification of an unlimited set of specific meanings, but retaining a musical analog-to-analog capability to convey subtle emphasis and emotion via prosody. Nonetheless, analog vs. discrete mapping seems to capture a basic distinction between language and music. This mapping distinction has an interesting empirical implication. If we want to understand the difference between analog and digital interpretive mappings, an experimental comparison of musical and linguistic interpretation provides an excellent place to start. If there are distinctive cognitive and neural processes underlying this difference, a comparison of musical and linguistic processing should be a good way to discover them (e.g., Koelsch et al 2004). Furthermore, the substantial variation among normal individuals in musical talent, which dwarfs the variation in linguistic ability among normal humans, provides a valuable empirical wedge into the question of the neural bases of these types of natural computation. Thus, the application of the analog/digital distinction to the problem of meaning raises some intriguing experimental possibilities.

Language Evolution: Finally, I will explore some implications of the hierarchical vs. flat structure distinction for the evolution of language. Some degree of hierarchical structure, perhaps limited to motor control, may be part of the basic vertebrate cognitive toolkit. In communication systems, some form of phonological phrase structure appears to characterize bird and whale song, and human music. But fully recursive hierarchy, where phrases are embedded within phrases, appears to be unique to human language. Cognitive algorithms for recursively generating complex sentence structures must be implemented with the same neurons and neurotransmitters as any other operations. How could such a novel capacity evolve? Perhaps it represents a modification of some pre-existing ability: three possibilities are a preexisting primate communication system, motor control and Machiavellian social intelligence. While the vocal communication systems of our primate ancestors clearly provided the precursors of many aspects of speech, they seem to be lacking recursive phrase structure. And to the extent that nonhuman primates calls have hierarchical structure at all (e.g., in gibbon "song"), order is not freely permutable but seems fixed in its sequential structure. The interpretive abilities of primates in this domain seem to be much greater than their production capabilities (e.g. Cheney & Seyfarth, 1990; Zuberbühler, 2002). Thus, it is far from clear that the vocal communication system of our common ancestor with chimps had characteristics suitable to provide an evolutionary precursor for this key hierarchical aspect of language. Motor control seems a more promising precursor system for hierarchicality, as recognized by several theorists (e.g., Greenfield, 1991, Byrne & Russon 1998). The kind of flexible hierarchical control implicit in the ability to catch prey while locomoting, or to grasp and manipulate tools, appears a promising precursor of hierarchical structure in the vocal realm. However, nothing in the domain of nonhuman motor control has the recursive, structure-preserving hierarchicality characteristic of language. Thus, while I think it is very plausible that motor hierarchicality is closely linked to phonological hierarchicality (not just in human speech but in music and probably in birdsong as well), it is ill-suited as a precursor to the syntactic and semantic level of hierarchicality in language.

A more plausible precursor ability for recursive self-embedding is provided by the cognitive operations concerned with social intelligence, particularly the Machiavellian intelligence so typical of primates (Byrne & Whiten, 1988). Recursive embedding seems obviously useful in social life, and particularly the ability to conceptualize the thoughts of others. While the concept "John intends to attack me" may be useful, the ability to entertain a higher-order concept that "My friend Joe sees that John intends to attack me" could provide a decisive advantage for coalition-forming primates like chimps or baboons. And once a single such level of mental embedding was widespread, it is easy to see how a further level would become selective: "John sees that my friend Joe will aid me if he attacks now, and will wait till I'm alone to attack". Recent evidence suggests that baboons can apply some degree of hierarchical cognition when interpreting the calls of others (Bergman, et al. 2003). This and other data suggest that some recursive, hierarchical-structure-preserving embedding of "minds within minds" may already have been present in our shared ancestor with chimps. Such mind-reading recursion has the proper computational structure to provide a precursor ability for the recursive hierarchical signature of syntax and semantics. If true, this might explain why birds do not seem to have evolved anything like language, despite the presence of many of the requisite cognitive abilities. They may lack the conceptual abilities associated with Machiavellian "mind-reading", more typical of primates. However, a note of caution is warranted, since the communication systems of songbirds like crows and other corvids, which share the vocal abilities of other songbirds but live in large, complex social groups and thus might be good candidates for complex mind-reading abilities, are still very poorly understood.

Summarizing, I have discussed four distinctions at the heart of computation and explored some of their implications for contemporary cognitive science. I suggest that progress in understanding natural computation will require a more complex model of neural processing, that respects the complexity of the neuron and the complex ways in which the brain implements the analog/digital and serial/parallel distinctions. These distinctions will be especially important for understanding the role of temporal processing in natural computation. I suggested that the serial/parallel distinction has an important implication for the evolution of consciousness, which far from being epiphenomenal seems to be a core computational requirement for successful learning in a parallel architecture. With regards to language, I propose that the analog/digital distinction provides an interesting cut into the problem of meaning, with some testable empirical consequences. Finally, I suggested that the flat/hierarchical distinction so crucial for understanding the evolution of syntax and semantics in human language seems more likely to derive phylogenetically from the "mind-reading" conceptual capabilities of nonhuman primates than the phonological structure of animal communication systems.

Acknowledgements

I thank A. Cutler, M. Bierwisch, T. Deacon, D. Dennett, P. Földiak, M. Kinsbourne, D. Raubenheimer, S. Thompson-Schill and an anonymous reviewer for helpful discussions and/or comments on the manuscript.

References

Bergman, T. J., Beehner, J. C., Cheney, D. L., & Seyfarth, R. M. (2003). Hierarchical classification by rank and kinship in baboons. *Science, 302*, 1234-1236.

Bierwisch, M. (1979). Gestische form als bedeutung musikalischer zeichen. *Jahrbuch Peters 1978*, 161-178.

Braitenberg, V. (1977). *On the texture of brains*. New York: Springer-Verlag.

Byrne, R. W., & Whiten, A. (1988). *Machiavellian Intelligence: Social expertise and the evolution of intellect in monkeys, apes and humans*. Oxford: Clarendon Press.

Byrne, R. W., & Russon, A. E. (1998). Learning by imitation: A hierarchical approach. *Behavioral and Brain Sciences, 21*, 667-684.

Cheney, D. L., & Seyfarth, R. M. (1990). *How monkeys see the world: Inside the mind of another species.* Chicago, IL: Chicago University Press.

Chomsky, N. (1957). *Syntactic Structures.* The Hague: Mouton.

Chomsky, N. (1959). Review of "Verbal Behavior". *Language, 35*, 26-58.

Churchland, P. M. (1995). *The engine of reason, the seat of the soul.* Cambridge, MA: MIT Press.

D'Amato, M. R. (1988). A search for tonal pattern perception in cebus monkey: Why monkeys can't hum a tune. *Music Perception, 5*, 452-480.

Deacon, T. W. (1997). *The Symbolic Species: The co-evolution of language and the brain.*

New York: Norton.
Debanne, D. (2004). Information processing in the axon. *Nature Reviews Neuroscience, 5*(4), 304-316.
Dennett, D. C. (1991). *Consciousness explained*. Boston: Little, Brown & Co.
Dennett, D. C., & Kinsbourne, M. (1995). Time and the observer: The where and when of consciousness in the brain. *Behavioral and Brain Sciences, 15*, 183-247.
Fitch, W. T., & Hauser, M. D. (2004). Computational constraints on syntactic processing in a nonhuman primate. *Science, 303*, 377-380.
Greenfield, P. M. (1991). Language, tools, and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior. *Behavioral and Brain Sciences, 14*, 531-595.
Häusser, M., & Mel, B. (2003). Dendrites: Bug or feature? *Current Opinion in Neurobiology, 13*, 372-383.
Jackendoff, R. (2002). *Foundations of Language*. New York: Oxford Univ. Press.
Koelsch, S., Kasper, E., Sammler, D., Schulze, K., Gunter, T. C., & Friederici, A. D. (2004). Music, Language, and Meaning: Brain Signatures of Semantic Processing. *Nature Neuroscience, 7*, 511-514.
Krubitzer, L. (1995). The organization of neocortex in mammals: are species differences really so different? *Trends in Neurosciences, 18*, 408-417.
Marcus, G. F. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. Cambridge, MA: MIT Press.
Morgan, J. L., Meier, R. P., & Newport, E. L. (1989). Facilitating the acquisition of syntax with cross-sentential cues to phrase structure. *Journal of Memory and Language, 28*, 360-374.
Oppenheim, A. V., & Schafer, R. W. (1989). *Discrete-time signal processing*. Englewood Cliffs, NJ: Prentice Hall.
Richards, W. (Ed.). (1988). *Natural Computation*. Cambridge, Mass.: MIT. Press.
Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1997). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press/Bradford Books.
Shamma, S. (2001). On the role of space and time in auditory processing. *Trends in Cognitive Sciences, 5*(8), 340-348.
Skiena, S. S. (1998). *The algorithm design manual*. New York: Springer Verlag.
Stein, R. B. (1967). The information capacity of nerve cells using a frequency code. *Biophysical Journal, 7*, 797-826.
Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*, 433-460.
von Neumann, J. (1958). *The computer and the brain*. New Haven, Connecticut: Yale University Press.
Wei, D.-S., Mei, Y.-A., Bagal, A., Kao, J. P. Y., Thomspon, S. M., & Tang, C.-M. (2001). Compartmentalized and binary behavior of terminal dendrites in hippocampal pyramidal neurons. *Science, 293*, 2272-2275.
Wright, A. A., Rivera, J. J., Hulse, S., Shyan, M., & Neiworth, J. J. (2000). Music perception and octave generalization in rhesus monkeys. *Journal of Experimental Psychology: General, 129*(3), 291-307.
Zuberbühler, K. (2002). A syntactic rule in forest monkey communication. *Animal Behaviour, 63*, 293-299.