Untimed and Misrepresented: Connectionism and the Computer Metaphor

Inman Harvey

CSRP 245

November 1992

ISSN 1350-3162

UNIVERSITY OF

Cognitive Science Research Papers

Untimed and Misrepresented: Connectionism and the Computer Metaphor

Inman Harvey

November 1992

School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH, England
tel: (+44 1273)/01273 678524
email: inmanh@cogs.susx.ac.uk

Abstract

The computer metaphor for the mind or brain has long outlived its usefulness, being based on Cartesian ideas. Connectionism has not broken free from this metaphor, and this has stunted the directions connectionist research has taken. The subordinate role of timing in computations has resulted in networks with real-value timelags on signals passing between nodes being ignored. The notion of representation in connectionism is generally confused;

program is waiting for input, which it then processes near-instantaneously. In general it is a *good thing* for such a program to continue for ever, or at least until the exit command is keyed in.

The cognitivist approach asserts that something with the power of a Turing machine is both necessary and sufficient to produce intelligence; both human intelligence and equivalent machine intelligence. Although not usually made clear, it would seem that something close to the model of a word-processing program is usually intended; i.e., a program that constantly awaits inputs, and then near-instantaneously calculates an appropriate output before settling down to await the next input. Life, so I understand the computationalists to hold, is a sequence of such individual events, perhaps processed in parallel.

3 ime in Computations and in Connectionism

One particular aspect of a computational model of the mind which derives from the underlying Cartesian assumptions common to traditional AI is the way in which the issue of *time* is swept under the carpet — only the sequential aspect of time is normally considered. In a standard computer operations are done serially, and the lengths of time taken for each program step are for formal purposes irrelevant. In practice for the machine on my desk it is necessary that the time-steps are fast enough for me not to get bored waiting. Hence for a serial computer the only requirement is that individual steps take as short a time as possible. In an ideal world any given program would be practically instantaneous in running, except of course for those unfortunate cases when it gets into an infinite loop.

The common connectionist assumption is that a connectionist network is in some sense a parallel computer. Hence the time taken for individual processes within the network should presumably be as short as possible. They cannot be considered as being effectively instantaneous because of the necessity of keeping parallel computations in step. The standard assumptions made fall into two classes.

1. The timelag for activations to pass from any one node to another it is connected to, including the time taken for the outputs from a node to be derived from its inputs, is in all cases exactly one unit of time (e.g. a back-propagation, or an Elman network).

2. Alternatively, just one node at a time is updated independently of the others, and the choice of which node is dealt with next is stochastic (e.g. a Hopfield net or a Boltzmann machine).

The first method follows naturally from the computational metaphor, from the assumption that a computational process is being done in parallel. The second method is closer to a dynamical systems metaphor, yet still computational language is used. It is suggested that a network, after appropriate training, will when presented with a particular set of inputs then sink into the appropriate basin of attraction which appropriately classifies them. The network is used as either a distributed content-addressable memory, or as a classifying engine, as a module taking part in some larger-scale computation. The stochastic method of relaxation of the network may be used, but the dynamics of the network are thereby made relatively simple, and not directly relevant to the wider computation. It is only the stable attractors of the network that are used. It is no coincidence that the attractors of such a stochastic network are immensely easier to analyse than any non-stochastic dynamics.

It might be argued that connectionists are inevitably abstracting from real neural networks, and inevitably simplifying. In due course, so this argument goes, they will slowly extend the range of their models to include new dimensions, such as that of time. What is so special about time — why cannot it wait? Well, the simplicity at the formal level of connectionist architectures which need synchronous updates of neurons disguises the enormous complexity of the physical machinery needed to maintain a universal clock-tick over distributed nodes in a physically instantiated network. From the perspective advocated here, clocked networks form a particular complex subset of all realtime dynamical networks ones need be, and if anything *they* are the ones that should be left for later (van Gelder 1992).

A much broader class of networks is that where the timelags on individual links between nodes is a real number which may be fixed or may vary in a similar fashion to weightings on such links[2]. A pioneering attempt at a theory that incorporates such timelags as an integral part is given in (Malsburg and Bienenstock 1986).

In neurobiological studies the assumption seems to be widespread that neurons are passing information between each other 'encoded' in the rate of firing. By this means it would seem that real numbers could be passed, even though signals passing along axons seem to be all-or-none spikes. This assumption is very useful, indeed perhaps invaluable, in certain areas such as

what is representing something *to what* is made. Yet the chapter can be sensibly interpreted as implicitly taking different layers in a network to be the different *whats*. When a more abstract, philosophical approach to discussion of connectionist representation is taken, as for instance in a collection of papers in (Ramsey *et al.* 1991), the absence of any clarification or specification of the *whats* makes it difficult, from my perspective, to work out what, if anything, is being said.

The gun I reach for whenever I hear the word *representation* has this engraved on it: "When P is used by Q to represent R to S, *who is* Q *and who is* S?". If others have different criteria for what constitutes a representation, it is incumbent on them to make this explicit. In particular I am puzzled

Returning briefly to the first issue raised, that of real-valued timelags within networks; the decomposition of a network by *divide and conquer*, into modules thought of as operating sequentially, is made far trickier if processes are going on concurrently in a way that is not globally clocked. It is no doubt this complexity of analysis that has helped to put people off investigating the broader class of networks.

7 Sketch of an Alternative

If one abandons the computer metaphor, the problem of how to make an intelligent machine becomes:

[Brooks 1991] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[Brooks 1992] Rodney A. Brooks. Artificial life and real robots. In *Proceedings of the First European Conference on Artificial Life*. MIT Press/Bradford Books, Cambridge, MA, 1992.

[Dreyfus 1972]