Untimed and Misrepresented: Connectionism and the Computer Metaphor

Inman Harvey

CSRP 245

November 1992

ISSN 1350-3162

UNIVERSITY OF

SUSSEX

AT BRIGHTON

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; this can be clarified when at all times it is made explicit who or what Q and S are in the formula "P is used by Q to represent R to S". Frequently they may be layers or modules within a network, but the typical confusion is symptomatic of the computer metaphor which in practice favours feedforward and militates against arbitrarily connected networks.

Rejecting this metaphor, an alternative paradigm is suggested of a brain as a complex dynamical system; investigating the dynamics of arbitrarily connected networks with real-valued timelags, specified so as to produce appropriate behaviour when they act as a nervous system for an organism or machine in continuous longterm interaction with its environment. The practical differences a change of metaphor makes are pointed out, and possible techniques for pursuing this line are indicated.

1 Introduction

Metaphors for the mind or the brain go through fashions, usually based on the prominent technology of the day; hydraulic machinery, telephone exchanges, computers. The computer metaphor has in recent decades been so all-pervasive that its tenets have ceased to be made explicit. It is all the more dangerous when it is taken for granted, and left out of any debate. These assumptions have affected the directions taken in connectionist research, which could be (but rarely are) fitted into a different metaphor.

Connectionism (or Artificial Neural Networks, or Parallel Distributed Processing) is frequently promoted as a parallel form of computation, or of information processing[1]. Many applications are indeed just this, but the danger is that when connectionism is proposed either as a model of the mind, or as a technique for producing an 'artificially intelligent' machines, the computational

[1] Indeed, in the early years of the current resurgence in connectionism, a lot of effort was spent in trying to convince people that these networks *were* doing computation, in order that the field could gain respectability — personal communication, G.E. Hinton

metaphor still lies unsaid in the background. This paper is based on the assumption that such a cognitivist approach is flawed.

The alternative view taken here is that cognition, as ascribed to animals or potentially to machines, is something that can only be attributed to the conjunction of an organism and the world that it inhabits. From this it follows that it would be a category error to treat cognition as something 'done' by the brain, or a part of the brain. This view is to a large extent shared by a significant number of people who have in the past been regarded as radical, or more commonly been completely ignored by mainstream cognitive science and AI. The time has surely arrived, by now, when such views can be assumed to be recognised (even if not accepted) by a cognitive science audience.

Just as the mainstream Cartesian paradigm is generally not argued for in papers that assume it, the alternative view will not be argued for here. Arguments for versions of it, expressed far better than I could, can be found in, e.g. (Dreyfus 1972, Gibson 1979, Agre 1988, Maturana and Varela 1987, Varela *et al.* 1991) — which should not be taken as implying that these authors would agree with each other.

What will be suggested in this paper is that the mainstream Cartesian paradigm has gravely restricted the class of connectionist models that have in practice been investigated. Two particular consequences will be studied in depth: the use of *time* in connectionist networks, and the practical effects of an unconsidered notion of *representation*. A sketch of a broader class of connectionist networks will be given.

In order to avoid many of the traditional mind-body questions, rather than looking at the human mind in this paper we will primarily concentrate on connectionist approaches to producing an 'artificially intelligent' machine that can behave autonomously within an environment. The 'brain' or 'nervous system' of the machine can be considered as a Black Box connected to sensors and actuators, such that the behaviour of the machine plus brain within its environment can be seen to be intelligent, or sensible; at a minimum, that it maintains its viability over a period of time.

The question then is, 'What to put in the Black Box?'. The computationalists will say that it should be computing appropriate outputs from its inputs. Or possibly they may say that whatever it is doing should be *interpretable* as doing such a computation. In contrast to this, a 'dynamical systems' metaphor will be here advocated. For those imbued with the mainstream Cartesian paradigm, these ideas may be so foreign that it is difficult to visualise how to set about designing such a system; an evolutionary technique will be advocated.

2 What is the Computer Metaphor?

The concepts of computers and computations, and programs, have a variety of meanings which shade into each other. On the one hand a computer is a formal system with the same powers as a Turing Machine (...assuming the memory is of adequate size). On the other hand a computer is this object sitting in front of me now, with screen and keyboard and indefinite quantities of software.

A program for the formal computer is equivalent to the pre-specified marks on the Turing machine's tape. For a given starting state of this machine, the course of the computation is wholly determined by the program and the Turing machine's transition table; it will continue until it halts with the correct answer, unless perhaps it continues forever — usually considered a *bad thing*!

On the machine on my desk I can write a program to calculate a succession of co-ordinates for the parabola of a cricket-ball thrown into the air, and display these both as a list of figures and as a curve drawn on the screen. Here I am using the machine as a convenient fairly user-friendly Turing machine.

However most programs for the machine on my desk are very different. At the moment it is (amongst many other things) running an editor or word-processing program. It sits there and waits, sometimes for very long periods indeed, until I hit a key on the keyboard, when it virtually immediately pops a symbol into an appropriate place on the screen; unless particular control keys are pressed, causing the file to be written, or edits to be made. Virtually all of the time the

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 Time 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 early sensory processing. Yet it is perverse to assume that this is true throughout the brain, a perversity which while perhaps not caused by the computational metaphor is certainly aided by it. Experiments demonstrating that the individual timing of neuronal events in the brain, and the temporal coincidence of signals passing down separate 'synfire chains', can be of critical importance, are discussed in (Abeles 1982).

A sketch of a broader class of networks in which the time dimension is not suppressed by the computer metaphor will be given in the penultimate section. For now, we move on to consider the influence of computer-oriented notions of representation on connectionism.

4 What is a Representation?

The concept of symbolic reference, or representation, lies at the heart of analytic philosophy and of computer science. The underlying assumption of many is that a real world exists independently of any given observer; and that symbols are entities that can 'stand for' objects in this real world — in some abstract and absolute sense. In practice, the role of the observer in the act of representing something is ignored.

Of course this works perfectly well in worlds where there is common agreement amongst all observers — explicit or implicit agreement — on the usages and definitions of the symbols, and the properties of the world that they represent. In the worlds of mathematics, or formal systems, this is the case, and this is reflected in the anonymity of tone, and use of the passive tense, in mathematics. Yet the dependency on such agreement is so easily forgotten — or perhaps ignored in the assumption that mathematics is the language of God.

A symbol P is used by a person Q to represent, or refer to, an object R to a person S. Nothing can be referred to without somebody to do the referring. Normally Q and S are members of a community that have come to agree on their symbolic usages, and training as a mathematician involves learning the practices of such a community. The vocabulary of symbols can be extended by defining them in terms of already-recognised symbols.

The English language, and the French language, are systems of symbols used by people of different language communities for communicating about their worlds, with their similarities and their different nuances and clichés. The languages themselves have developed over thousands of years, and the induction of each child into the use of its native language occupies a major slice of its early years. The fact that, nearly all the time we are talking English, we are doing so to an English-speaker (including when we talk to ourselves), makes it usually an unnecessary platitude to explicitly draw attention to the community that speaker and hearer belong to.

Since symbols and representation stand firmly in the linguistic domain, another attribute they possess is that of arbitrariness (from the perspective of an observer external to the communicators). When I raise my forefinger with its back to you, and repeatedly bend the tip towards me, the chances are that you will interpret this as 'come here'. This particular European and American sign is just as arbitrary as the Turkish equivalent of placing the hand horizontally facing down, and flapping it downwards. Different actions or entities can represent the same meaning to different communities; and the same action or entity can represent different things to different communities. In Mao Tse-Tung's China a red traffic light meant *GO*.

In the more general case, and particularly in the field of connectionism and cognitive science, when talking of representation it is imperative to make clear who the users of the representation are; and it should be possible to at a minimum suggest how the convention underlying the representation arose. In particular it should be noted that where one and the same entity can represent different

[2] For a simple model without loss of generality any time taken for outputs to be derived from inputs within a node can be set to zero, by passing any non-zero value on instead to the links connected to that node.

things to different observers, conceptual confusion can easily arise. When in doubt, always make explicit the Q and S when P is used by Q to represent R to S.

In a computer program a variable `pop_size` may be used by the programmer to represent (to herself and to any other users of the program) the size of a population. Inside the program a variable i may be used to represent a counter or internal variable in many contexts. In each of these contexts a metaphor used by the programmer is that of the program describing the actions of various homunculi, some of them keeping count of iterations, some of them keeping track of variables, and it is within the context of particular groups of such homunculi that the symbols are representing. But how is this notion extended to computation in connectionist networks?

5 Representation in Connectionism

When a connectionist network is being used to do a computation, in most cases there will be input, hidden and output nodes. The activations on the input and output nodes are decreed by the connectionist to represent particular entities that have meaning for her, in the same way as `pop_size` is in a conventional program. But then the question is raised — 'what about internal representations?'.

If a connectionist network is providing the nervous system for a robot, a different interpretation might be put on the inputs and outputs. But for the purpose of this section, the issues of internal representation are the same.

All too often the hidden agenda is based on a Platonic notion of representation — what do activations or patterns of activations represent in some absolute sense to God? The behaviour of the innards of a trained network are analysed with the same eagerness that a sacrificed chicken's innards are interpreted as representing ones future fate. There is however a more principled way of talking in terms of internal representations in a network, but a way that is critically dependent on the observer's decomposition of that network. Namely, the network must be decomposed by the observer into two or more modules that are considered to be communicating with each other by means of these representations.

Where a network is explicitly designed as a composition of various modules to do various subtasks (for instance a module could be a layer, or a group of laterally connected nodes within a layer), then an individual activation, or a distributed group of activations, can be deemed to represent an internal variable in the same way that i did within a computer program. However, unlike a program which wears its origins on its sleeve (in the form of a program listing), a connectionist network is usually deemed to be internally 'nothing more than' a collection of nodes, directed arcs, activations, weights and update rules. Hence there will usually be a large number of possible ways to decompose such a network, with little to choose between them; and it depends on just where the boundaries are drawn just who is representing what to whom.

It might be argued that some ways of decomposing are more 'natural' than others; a possible criterion being that two sections of a network should have a lot of internal connections, but a limited number of connecting arcs between the sections. Yet as a matter of interest this does not usually hold for what is perhaps the most common form of decomposition, into layers. The notion of a distributed representation usually refers to a representation being carried in parallel in the communication from one layer to the next, where the layers as a whole can be considered as the Q and S in the formula "P is used by Q to represent R to S".

An internal representation, according to this view, only makes sense relative to a particular decomposition of a network chosen by an observer. To assert of a network that it contains internal representations can then only be justified as a rather too terse shorthand for asserting that the speaker proposes some such decomposition. Regrettably this does not seem to be the normal usage of the word. While claiming that my usage of the word representation as outlined above is the careful and principled form that underlies the confused and careless way in which the word is frequently used, I am aware of the dangers of claiming to be the only person in the platoon that is in step. Nevertheless, until I see an alternative formulation clearly laid out, I shall continue to be puzzled by much of what is written on the subject.

In (Hinton *et al.* 1986), reprinted in (Boden 1990), an attempt is made to make sense of distributed representation in connectionist networks. No acknowledgment of any necessity to specify

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 as to how they can reconcile (if they believe it is not necessary to specify Q and S) the same symbol representing different things to different communities.

6 Are Representations Needed?

With this approach to the representation issue, then any network can be decomposed (in a variety of ways) into separate modules that the observer considers as communicating with each other. The interactions between such modules can *ipso facto* be deemed to be mediated by a representation. Whether it is useful to do so is another matter.

Associated with the metaphor of the mind (or brain, or an intelligent machine) as a computer go assumptions of functional decomposition. Since a computer formally manipulates symbols, yet it is light waves that impinge on the retina or the camera, surely (so the story goes) some intermediate agency must do the necessary translating. Hence the traditional decomposition of a cognitive system into a perception module, which takes sensory inputs and produces a world model; this is passed onto a central planning module which reasons on the basis of this world model; passing on its decisions to an action module which translates them into the necessary motor actions. This functional decomposition has been challenged, and an alternative behavioural decomposition proposed, by Brooks in, e.g., (Brooks 1991).

In particular, the computationalist or cognitivist approach seems to imply that communication between any such modules is a one-way process; any feedback loops are within a module. Within for instance back-propagation, the backward propagation of errors to adjust weights during the learning process is treated separately from the forward pass of activations. This helps to maintain the computational fiction, by conceptually separating the two directions, and retaining a feed-forward network. But consider the fact that within the primate visual processing system, when visualised as a network, there are many more fibres coming 'back' from the visual cortex into the Lateral Geniculate Nucleus (LGN) than there are fibres going from the retina to the LGN in the 'correct' direction. How does the computationalist make sense of this?

Marr (in (Marr 1977), reprinted in (Boden 1990)) classifies AI theories into Type 1 and Type 2, where a Type 2 theory can only solve a problem by the simultaneous action of a considerable number of processes, *whose interaction is its own simplest description*. It would seem that type 2 systems can only be decomposed arbitrarily, and hence the notion of representation is less likely to be useful. This is in contrast to a Type 1 theory, where a problem can be decomposed into a form that an algorithm can be formulated to solve, by *divide and conquer*. Type 1 theories are of course the more desirable ones when they can be found, but it is an empirical matter whether they exist or not. In mathematics the 4-colour theorem has been solved in a fashion that requires a large number of special cases to be exhaustively worked out in thousands of hours of computation (Appel and Haken 1989). It is hoped that there were no hardware faults during the proof procedure, and there is no way that the proof as a whole can be visualised and assessed by a human. There is no *a priori* reason why the workings of at least parts of the brain should not be comparably complex, or even more so[3]. This can be interpreted as: there is no *a priori* reason why all parts of the brain should be in such a modular form that representation-talk is relevant. The answer to the question posed in the title of this section is *no*. This does not rule out the possibility that in some circumstances representation-talk *might* be useful, but it is an experimental matter to determine this.

[3] For the purposes of making an intelligent machine or robot, it has in the past seemed obvious that only Type 1 techniques could be proposed. However evolutionary techniques need not restrict themselves in this fashion (Husbands and Harvey 1992).

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: what sort of physical system should be put inside the Black Box of its nervous system so that it behaves appropriately in its environment? A cogent argument for a dynamical systems perspective has been independently put forward by Beer in (Beer 1992). Of course, abandoning the computer metaphor does not prevent one from using a computational model of a physical system and calculating its behaviour, just as one can calculate the parabola of a cricket ball without claiming that the ball itself is some form of computer.

The computational approach would imply that the 'brain' of a machine has access through its sensors to information about the machine's world, which it can then reason about. We are dismissing this notion, and instead have to rely on processes whereby the physical system within the Black Box can adapt itself according to some given criteria. This plasticity of behaviour is, on the individual scale, what we call learning, but how could such a learning system be devised?

The model proposed for this is Darwinian evolution as we understand it in the biological world. The incremental adaptation of cognitive structures, through a process of selection, alteration and addition, requires interaction of successive generations of a species with its own developing world (Harvey 1992). Adaptation need not be on an individual scale alone, as accumulative change over generations can be thought of as adaptation on a longer timescale. But what class of physical system should be evolved in this fashion? Why not some programming language, as advocated by Brooks (Brooks 1992)?

There are good grounds for thinking that a generalised form of connectionist network could be one very appropriate class. Let us start with three basic axioms:

1. The 'brain' should be a physical system, occupying a physical volume with a finite number of input and output points on its surface.

2. Interactions within the brain should be mediated by physical signals travelling with finite velocities through its volume, from the inputs, and to the outputs.

3. Subject to some lower limit of an undecomposable 'atom' or node, these three axioms apply to any physical subvolume of the whole brain.

A justification for the third axiom is that of the incremental development of the whole by alterations and additions over evolutionary timescales. The consequence of these axioms, as can be seen by shrinking in any fashion the surface containing the original volume, is a network model where internal nodes are the undecomposable atoms, and connections between inputs, internal nodes and outputs are through directed arcs by signals taking finite times. Such a network can be arbitrarily recurrent. The assumption of only a finite number of input/output points on any surface rules out of this model such more general methods of physical interaction as might be assumed to be involved with, e.g. chemical neurotransmitters in the human brain.

No assumptions about the operations of the nodes have yet been made. The simplest assumptions would be those of standard connectionist models. Input signals are weighted by a scalar quantity; all output signals are identical when they leave the node, being calculated from the weighted sum of the inputs. If this weighted sum is passed through a sigmoid or thresholding function then we have the non-linear behaviour we have learnt to know and love. So far the only generalisation this model has when compared with the picture given in (McClelland and Rumelhart 1986) is that timelags between nodes need to be specified. But a whole new universe of possible dynamical behaviours is opened up by this extension.

Apart from the philosophical blinkers which have contributed to the neglect of this generalisation, such networks are more difficult to analyse — but still possible to synthesize. With an

evolutionary approach it may not be necessary to analyse *how it works*, but rather one should be able to assess *how good is the behaviour it elicits* (Harvey 1992, Husbands and Harvey 1992). This is no short-cut recipe, but requires that the internal complexity of the 'brain' (of an organism or a machine) be dependent on the history of interactions with its world; the more the complexity that is required, the longer the history that is needed to mould it.

8 Conclusions

As a preamble, nothing said herein should be taken as denying that connectionist networks can be used for doing a form of parallel computation. Nor should this paper be taken as claiming that it is impossible that any part of the human brain could be usefully interpreted as performing some such parallel computation — or that such techniques should never be used in an intelligent machine. In addition, it should be noted that this paper does not purport to here justify its underlying stance against the mainstream Cartesian paradigm.

What *is* being asserted is that the exclusive interpretation of connectionist networks within the computational metaphor — a pervasive practice grounded within the mainstream paradigm — has severely limited the types of networks investigated. Two particular consequences of this misdirection have been followed up.

Firstly, the irrelevance of time in serial computation, except as a dimension for ordering program steps, means that for the most part only two impoverished subclasses of networks have been analysed; where all internal interactions take a unit time step, or where individual nodes are updated in stochastic order. This does of course make analysis easier, at the expense of avoiding the complexities of behaviour possible when timelags between nodes have individual real values in the same way that weightings usually have.

Secondly, the carry over to connectionist networks of the often inappropriate notion of representation is associated with a desire to decompose networks into modules, often layers, which can be seen to be communicating interpretable messages between each other; in Marr's terms, a Type 1 decomposition. This has militated against the investigation of arbitrarily recurrent networks (with the exception of Hopfield-type nets using stochastic updates, which themselves tend to only make sense as a component within some larger computational system).

Reasons have been given for investigating a broader class of connectionist networks for use in 'intelligent machines', and a possible technique for doing so indicated.

Acknowledgments

This work is supported by a grant from the SERC. I thank Shirley Kitts for philosophical orientation, and Richard Dallaway for comments on an earlier draft.

References

[Abeles 1982] M. Abeles. *Local Cortical Circuits, An Electrophysiological study.* Springer-Verlag, 1982.

[Agre 1988] P.E. Agre. The dynamic structure of everyday life. Technical Report 1085, M.I.T. AI-LAB, 1988.

[Appel and Haken 1989] K. Appel and W. Haken. Every planar map is four colorable. *American Mathematical Society, Contemporary Mathematics*, 98, 1989.

[Beer 1992] R.D. Beer. A dynamical systems perspective on autonomous agents. Technical Report CES-92-11, Case Western Reserve University, Cleveland, Ohio, 1992.

[Boden 1990] M.A. Boden, editor. *The Philosophy of Artificial Intelligence.* Oxford University Press, 1990.

[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] H.L. Dreyfus. *What Computers can't do: a Critique of Artificial Reason*. Harper, New York, 1972.

[Gibson 1979] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979.

[Harvey 1992] Inman Harvey. Species adaptation genetic algorithms: The basis for a continuing SAGA. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 346–354. MIT Press/Bradford Books, Cambridge, MA, 1992.

[Hinton *et al.* 1986] G.E. Hinton, J.L. McClelland, and D.E. Rumelhart. Distributed representations. In D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 77–109. MIT Press, 1986.

[Husbands and Harvey 1992] P. Husbands and I. Harvey. Evolution versus design: Controlling autonomous robots. In *Integrating Perception, Planning and Action, Proceedings of 3rd Annual Conference on Artificial Intelligence, Simulation and Planning*, pages 139–146. IEEE Press, 1992.

[Malsburg and Bienenstock 1986] C. von der Malsburg and E. Bienenstock. Statistical coding and short-term synaptic plasticity: A scheme for knowledge representation in the brain. In E. Bienenstock, F. Fogelman Soulie, and G. Weisbuch, editors, *Disordered Systems and Biological Organization*. Springer-Verlag, 1986.

[Marr 1977] D.C. Marr. Artificial intelligence: A personal view. *Artificial Intelligence*, 9:37–48, 1977.

[Maturana and Varela 1987] H.R. Maturana and F.J. Varela. *The Tree of Knowledge: The Biological Roots of Human Understanding*. Shambhala Press, Boston, 1987.

[McClelland and Rumelhart 1986] J. L. McClelland and D. E. Rumelhart, editors. *Explorations in Parallel Distributed Processing*. MIT Press/Bradford Books, Cambridge Massachusetts, 1986.

[Ramsey *et al.* 1991] W. Ramsey, S.P. Stich, and D.E. Rumelhart, editors. *Philosophy and Connectionist Theory*. Erlbaum, 1991.

[van Gelder 1992] Tim van Gelder. What might cognition be if not computation. Technical Report 75, Indiana University Cognitive Sciences, 1992.

[Varela *et al.* 1991] F. Varela, E. Thompson, and E. Rosch. *The Embodied Mind*. MIT Press, 1991.