Cognition is Not Computation; Evolution is Not Optimisation

Inman Harvey

Centre for Computational Neuroscience and Robotics
School of Cognitive and Computing Sciences
University of Sussex, Brighton BN1 9QH, UK

Abstract. An overview is given of the role and relevance of Artificial Neural Nets (ANNs) as control systems for autonomous agents. Though ANNs can be used as computational input/output devices, cognition requires not this but rather some method of implementing dynamical systems. A wider class of ANNs incorporating temporal dynamics and feedback is presented, as one way to achieve this. These are difficult to design, and evolutionary approaches are a possible approach. Since evolving complex robot controllers inevitably takes a long time, one cannot afford to start afresh with each new problem, and an incremental adaptation approach will be necessary in the long term. This means that standard off-the-shelf optimising genetic algorithms are not appropriate unless adjusted to their new role.

1 Evolutionary Robotics

The creation of adaptive autonomous agents suffers from fantasy targets set for us by science fiction films; real achievements fall very far short of the expectations of the general public. Nobody really knows how to do it, there are well-known flaws in classical approaches, and the field is wide open to innovative ideas. Neural nets sound vaguely biological, genetic algorithms are flavour of the month, so it is tempting just to combine some off-the-shelf versions of these. Notwithstanding this rag-bag approach to evolutionary robotics, I argue that there are some principled analyses to the question of 'what is cognition — for autonomous agents'; and 'how can we design really complex artefacts, once we have exhausted any short cuts'. This leads to the use of a particular class of neural nets and a particular style of genetic algorithms.

I will argue that neural nets are in many cases used as a classical, computational form of agent or robot control system; and that genetic algorithms are in many cases used as a classical form of optimisation technique. But the long term future of Evolutionary Robotics lies in a much more radical agenda: that the cognition we wish autonomous agents to have has nothing to do with computation (even with neural nets), and that evolution, even artificial evolution, can be taken out of the narrow shortsighted framework of optimisation. I will refer to examples where this approach has been used in practice; but in this short paper the arguments will be limited to those of general principles which have significant practical consequences.

2 Cognition and Computation

In the attempt to create adaptive autonomous agents in our own image, people inevitably reveal their philosophical stance on what it is to be a cognitive being. I shall assume that we can for engineering purposes treat agents as machines; the type of machine held up as a paradigm has gone through fashions, usually based on the prominent technology of the day. Whereas the mind or brain used to be a type of hydraulic machinery, or a type of telephone exchange, for some decades now the most pervasive, unexamined, metaphor has been the computer.

Artificial Neural Networks (ANNs) have in recent years been promoted as a parallel form of computation, or of information processing. Many applications of ANNs are indeed just this, but the danger is that when they are proposed either as a model of the mind or as a technique for producing adaptive autonomous agents, the computational metaphor still lies unsaid in the background.

2.1 What are Computations?

Until the 1930s computers were human beings who carried out a predetermined set of calculations on figures, for instance long division. After Alan Turing and others demonstrated that machines could be designed to carry out manipulations on symbols according to rules, the word 'computer' has changed its primary meaning to the machines. Technological and theoretical developments have produced the amazing capabilities of these machines we see today, where such a computer can defeat the best chess grandmaster.

Underlying all this is the simple idea of a *computation*:

- Take some input data, such as a set of numbers, or a chess position and the history that led up to it, or the sensory inputs to a robot or animal.
- Carry out a specific algorithm on the input data, until the algorithm halts (subject to the Halting Problem).
- Present the output data that resulted, e.g. the long division, the next chess move, or the agents next motor movement.

There is no mention here of *time*. A slow computer and a fast one can perform the same computation; the lengths of time taken for each program step are for formal purposes irrelevant. Whereas von Neumann computers perform serially, a notion of computation can be extended to parallel computations done in ANNs.

This attractive explanation conceals a fundamental assumption which I and many others would challenge. The brain is not doing any computations at all in the sense spelt out above. There is a regrettably tendency to use the word 'computation' to refer to the workings of any complex system, but this results in the word losing any useful sense, and such silly claims as a thunderstorm, or indeed the universe, is *computing* its next state. In denying that the brain performs computations I am not challenging this vacuous use of the word, but rather the sense that Alan Turing used when defining a universal computer: an algorithm performed on input data to generate appropriate output data.

This challenge has significant practical consequences, including moving away from feedforward atemporal networks, where input nodes are presented with input data, and after a sequence of parallel processing through successive layers the output data emerges from the far end. Historically, even when ANNs include feedback, the same computational perspective has often been applied; for instance with Hopfield nets which are full of feedback loops, after an input pattern is presented the network is allowed to settle down, sink into a basin of attraction, which thus specifies the output of this 'parallel computation'.

Yes these do indeed count as computations, in the sense spelt out above. Such ANNs may be appropriate for some computational purposes. But these are only a small subset of the ways in which ANNs may be used, and it is the non-computational forms that are appropriate models of cognition.

3 Non-computational Cognition

Characterising cognition as computation reduces it to a series of snapshot decisions: given my current sensory inputs, plus perhaps some trace of my previous history, what should be my output action *now*? The alternative view taken here is that cognition, for animals or machines, is something that can only be attributed to the behaviour that arises from the conjunction of an organism and the world that it inhabits. Hence it would be a category error to treat cognition as something 'done' by the brain, or a part of the brain. The behaviour of an organism arises from the dynamics of its interaction with its world, and from our perspective as external observers we can best describe this as the interaction between two dynamical systems (the agent and 'the rest'), coupled together through sensors and actuators. This is the 'dynamical systems view of cognition', dating from the early cyberneticists and reemerging in recent years [1, 5, 9].

Why is an agent in such a coupled set of dynamical systems not an input/output device, and hence not performing computations to generate appropriate outputs from snapshots of its sensory inputs? One could analyse the dynamical system that constitutes the agent — 'brain and body' — and correlate every output with the snapshots, the inputs that generated them, but in general this is still insufficient to give an understanding of how it will behave when coupled with its environment. The changing actuator outputs from the agent themselves influence the way in which the patterns of sensory inputs change, and if this is on a similar or a faster timescale than the time it takes sensory inputs to influence outputs, then the snapshot model breaks down.

4 Neural Nets as Dynamical Systems

A dynamical system can be specified by a number of variables which change over time in a regulated fashion. We are here excluding field systems and only considering those with a finite number of variables; the rate of change of any one variable is governed by a function of some or all of the total set of variables. If we

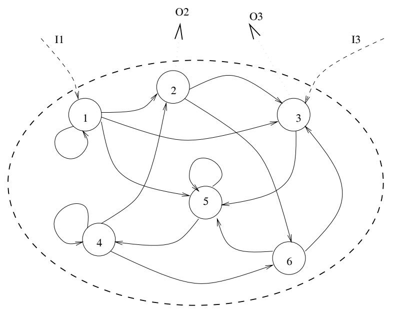

Fig. 1. *A sketch of a dynamic system with 6 variables, influenced by two parameters shown as inputs, and conceptualised as a dynamic recurrent neural net.*

conceptualise each variable as a node, and draw directed lines from each node onto those others whose rate of change it influences, then we have a diagram looking very much like an ANN (Figure 1); the functions correspond loosely to 'activation functions'. Two features to note about such ANNs are firstly, that in general there may be any amount of feedback loops; and secondly that crucial to their definition is their temporality, the time constants associated with their rates of change.

It follows that dynamic recurrent ANNs (DRNNs) are one convenient way to visualise dynamical systems, and this has been made use of by Beer [1, 2] and in Sussex evolutionary robotics work [6]. Beer uses a set of differential equations

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N} w_{ji}\sigma_j(y_j) + I_i(t)$$

where y_i is the activation of the ith node and $\sigma_j(\xi) = (1 + e^{(\theta_j - \xi)})^{-1}$ is a sigmoidal function biased by a threshold term θ_j. Inputs to the system from outside, I_i, would be parameters if they were unchanging; however with an agent in its environment these I_i vary according to the way the system is coupled with the rest of the world. A slightly different approach has been used in much of the Sussex work, where discrete changes in activations at nodes take place at discrete times, knockon effects are transmitted down interconnections taking a finite time to influence later events. There is a close resemblance here with Brooks' subsumption architecture [3] using Augmented Finite State Automata (AFSM). These latter can be seen as nodes in a DRNN, where the temporal qualities are provided by the timers within the AFSMs. Continuous time recurrent neural networks can be shown to be a class of dynamical systems capable *in principle* of replicating to an arbitrary degree of accuracy the dynamical behaviour of any other dynamical system with a finite number of components [4].

So DRNNs can do everything that formal computational systems can do and (crucially) much more; trivially, you cannot time the boiling of an egg with a formal computational system. Many people embedded in the computationalist frame of mind have difficulty in realising what a significant difference the deliberate recognition of temporal attributes of cognitive agents makes, the modelling of them as dynamical systems; perhaps a fair analogy would be to say that building an agent out of formal computational systems is akin to building an aeroplane with only knowledge of statics, no dynamics. DRNNs are however fiendishly difficult to analyse and understand, which is why evolutionary methods have often been used. DRNNs can of course have plastic behaviour at many different timescales, and hence agents built out of them can display learning behaviour; but if an evolutionary approach is used for the design of systems that behave in desired ways, then 'ability to learn in particular contexts' is *just* another constraint on the evaluation function, and need not be treated differently.

5 Artificial Evolution

Genetic algorithms (GAs), based on Darwinian ideas on natural evolution, are often called on as an optimisation technique in a high-dimensional search space, particularly where there is relatively little *a priori* knowledge to guide the search. I will suggest here that some problems should not be treated as optimisation problems, and in the long term much robotics development will fall into a different framework. Crucial properties of an optimisation problem are:

1. It is *one specific* problem.
2. The search space of possible solutions is typically well-defined in terms of a fixed number of parameters.

Now this is not in fact what goes on in natural evolution, where the 'problems' different organisms face were not predetermined at the origin of life on earth. Natural evolution can be thought of as a method for adaptive incremental improvement to organisms who are facing, over geological time, problems that vary with the environment which significantly includes other varying organisms. Engineers should only draw on principles from nature when they suit their purposes, but there is a broad class of problems where we should be looking for the adaptive-improving properties of evolution rather than optimisation properties. The crucial question is:

– If we substitute for a solved problem a slight variation on it, does it make sense to start a new search process *from scratch*, or should we be looking for adaptive improvement?

The quest for adaptive autonomous agents is inherently an ill-defined one, where we may set short-term goals but will continually want to move the goal-posts as each goal is achieved. So the longterm future of evolutionary robotics is in incremental evolution rather than GAs-as-optimisers. This requires a change of emphasis in the use of GAs, which has been reflected in the development of SAGA or Species Adaptation Genetic Algorithms [6]; this will be briefly summarised here, to emphasise the use of of-the-shelf GAs may be inappropriate for long term evolutionary robotics. The practical consequences of SAGA are that one works with a genetically converged population, the degree of convergence maintained by a balance between mutation and selection. Recombination is less important; the notion of 'premature convergence' becomes an irrelevance. A GA paradigm such as Genetic Programming may use a population of many hundred thousand members evolving for less than 100 generations; in this alternative paradigm a population which may be many orders of magnitude smaller evolves for many thousands of generations, indeed in principle indefinitely.

6 Summary

The dynamical systems approach to understanding cognition requires a system which no longer treats time as an afterthought, and dynamical systems can be conceptualised as dynamic recurrent neural nets, DRNNs, which have significantly different properties from many conventional ANNs. These are inherently difficult to analyse and design, and evolutionary methods are one way to attack such difficult problems. For problems such as evolutionary robotics, in the long term GAs as optimisers will be ineffective. The different framework of GAs as incremental adaptive improvers requires something other than standard off-the-shelf algorithms.

References

1. Beer, R. D.: A dynamical systems perspective on agent-environment interaction. Artificial Intelligence, v. 72, pp. 173–215, 1995.
2. Beer, R. D.: The dynamics of adaptive behavior: A research program. Robotics and Autonomous Systems. In Press 1997.
3. Brooks, R.: A robust layered control system for a mobile robot. IEEE J. Rob. Autom., v. 2, pp. 14–23, 1986.
4. Funahashi, K. and Y. Nakamura: Approximation of dynamical systems by continuous time recurrent neural networks. Neural Networks, v. 6, pp. 1–64, 1993.
5. Harvey, I.: Untimed and misrepresented: Connectionism and the computer metaphor. Cognitive Science Research Paper CSRP245, University of Sussex, 1992.
6. Harvey, I., P. Husbands, D. Cliff, A. Thompson and N. Jakobi: Evolutionary Robotics: the Sussex approach. Robotics and Autonomous Systems. In Press 1997.
7. McCulloch, W. S., and W. Pitts: A logical calculus od the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, v. 5, pp. 115–133, 1942.
8. Thompson, A.: An evolved circuit, intrinsic in silicon, entwined with physics. To appear: Proceedings of The First International Conference on Evolvable Systems: from Biology to Hardware (ICES96), Tsukuba, Japan, October 7-8 1996. Springer-Verlag LNCS, 1997.
9. van Gelder, T.: What might cognition be if not computation. Technical Report 75, Indiana University Cognitive Sciences, 1992.

This article was processed using the LaTeX macro package with LLNCS style