ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 39-44

Incremental Evolution of Neural Network Architectures for Adaptive Behaviour

Dave Cliff[1,2] and Inman Harvey[1] and Philip Husbands[1]
[1] School of Cognitive and Computing Sciences
[2] IRC, School of Biological Sciences
University of Sussex, BRIGHTON BN1 9QH, U.K.
davec or inmanh or philh@cogs.susx.ac.uk

Abstract

This paper describes theoretical aspects of our ongoing work in evolving recurrent dynamical artificial neural networks which act as sensory-motor controllers, generating adaptive behaviour in artificial agents. Some shortcomings in mainstream artificial neural network research are identified, and the rationale for our approach is discussed. This involves the use of recurrent networks of artificial neurons with rich dynamics, resilience to noise (both internal and external); and separate excitation and inhibition channels. The networks allow artificial agents (simulated or robotic) to exhibit adaptive behaviour. The complexity of *designing* networks built from such units leads us to use an extended form of genetic algorithm, which allows for incremental automatic evolution of controller-networks. Some recent results are reviewed, using these methods with simple visually-guided robots.

1 Introduction and Rationale

Increasingly, practitioners of artificial neural network research are realising that both the complexity of model neurons, and also the styles of network architecture, need to be extended beyond those employed in the much-cited work of the early 1980's. Certainly, models such as Hopfield networks, or back-propagating multi-layer perceptrons, played an important historical role in making parallel distributed processing an acceptable paradigm of study; but if we are to succeed in either understanding biological nervous systems, or in building artificial neural networks which exhibit intelligent behaviour, it is likely that we will have to move to more complex models.

But what form should this complexity take? The notion of 'complexity' is often highly subjective, and hence problematic. We should avoid introducing unnecessary complications, but (more importantly) we should not be deceived by our own simplifications. In artificial neural network (ANN) modelling, simplifications are made for various reasons. Often, there are issues of mathematical tractability: certain model neurons or network architectures are easier to formally analyse than others. In other cases, the ease with which the models can be simulated or built in available hardware is an important factor, and appropriate simplifications are made. In either case, it is important to note that the

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 39-44

'simplification' is made for *our* convenience: the ANN is easier to construct or understand. The problem with this approach is that in using simplified models, we may actually be making life harder for ourselves as scientists; because the tasks we try to make our models perform may, by their very nature, require greater complexity than is possible without using clever 'trick' techniques, or large and unwieldy modular assemblies of simple networks.

There are two simplifications which are very common in ANN models: most models in the literature have very simple (or non-existent) dynamics; and arbitrary connectivity is often avoided. Networks with many feedback connections and delays between units are much more challenging to either analyse, simulate, or build, than are networks such as the common three-layer back-propagation network. Yet for many interesting and important problems, feedback and intrinsic dynamics are likely to be essential. There is ample evidence in the neuroscience literature from most branches of the animal kingdom, that biological neural networks exhibit rich dynamical behaviour and exploit feed-back connections to great effect.

Many ANNs are developed purely to transform between representations or encodings which have been formulated by their designers. Such networks may be worthwhile engineering artefacts, performing useful computations; but it is important to remember that the primary evolutionary pressure on the development of biological nervous systems (which we seek to understand or draw inspiration from) was whether a particular nervous system helped an animal survive in environments which were dynamic and uncertain. That is to say, nervous systems evolved where they generated *adaptive behaviours* (i.e. behaviours which are likely to increase the chances that the individual animal survives to reproduce). We, in common with a growing number of other researchers, believe that the generation of adaptive behaviours should form the primary focus for research into cognitive systems, and that issues of purely transforming between representations or encodings are, at best, secondary.

It is the above factors that have influenced our recent work, discussed in the remainder of this paper. We have created ANNs which generate adaptive behaviours in artificial "animals" (i.e. robotic or simulated agents). Our agents have tactile sensors and minimal visual systems (two oriented photoreceptors). The ANNs use highly recurrent networks of artificial neurons (called "units"), with propagation delays as signals pass across links between units. The units have separate excitation and inhibition channels, and operate in the presence of noise introduced both internally (i.e. within each unit) and also externally (i.e. in sensory-motor transduction). The transfer functions for excitation and inhibition in each unit are nonlinear with discontinuities in the first derivative.

Naturally, either analysing or designing networks composed of such units is a challenging and difficult task. Nevertheless, we believe that units of the sort used in our work are closer to the minimum complexity acceptable for generating adaptive behaviours than are the simpler units of prior work. For this reason, the problems of design and analysis have to be tackled, rather than avoided by introducing simplifications. Our approach has been to, as far as is possible, *auto-*

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 39-44

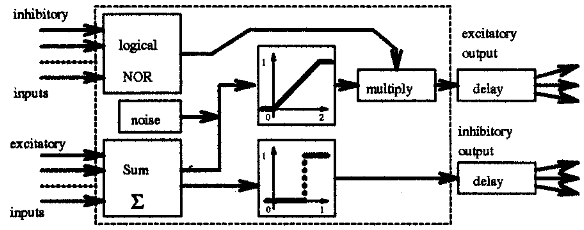

Figure 1: Block diagram showing operations within a single model neuron.

mate the design of the networks by employing our own extended form of genetic algorithm, SAGA. Whereas most genetic algorithms are essentially performing optimisation in a fixed parameter space, SAGA allows for the dimensionality of the parameter space to be under evolutionary control, by employing variable-length genotypes. In terms of the networks, this means we are able to start with a population of agents each of which has a minimal number of units: extra units may be introduced by the genetic operators, and will only be retained if they increase the evolutionary success of the mutated agent: our automatic network generation is truly incremental.

Below, the neuron model and network simulations are discussed. Details of how the networks are encoded as genotypes suitable for use with SAGA are given. Next, we discuss the adaptive behaviour evolved in our simulated agents, and present brief analysis of a resultant evolved network. This paper is necessarily brief: for further details of our rationale, see [1, 5]; for further details of SAGA, see [4]; for full details of the visual sensing employed, see [2]. For a more complete version of this paper, see [3].

2 The Model Networks

Because our networks are recurrent, there is no clear divide between different 'layers' (c.f. input, hidden, and output layers found in back-propagation networks). Nevertheless, for the purposes of generating adaptive behaviour, it is necessary to designate some units as receiving input from sensors, and others as producing outputs to actuators (such as motors). In practice, this designation may be distorted by the opportunistic evolutionary process; for instance, a unit linked to a sensor which is rarely triggered may be recruited as an internal, or 'hidden' unit. The remainder of this section discusses details of the neuron model, and how the networks architectures are encoded as 'genes' which can be operated on by the SAGA genetic algorithm.

2.1 The Neuron Model

The neuron model we have employed in our work to date has separate channels for excitation and inhibition. Values propagate along links between units, and are all real numbers in the range $[0, 1]$. All links are subject to a delay Δt. A schematic of the operations for one unit is shown in Figure 1. The inhibition

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 39-44

channels operate as a 'veto' or 'grounding' mechanism: if a unit receives *any* inhibitory input, its excitatory output is reduced to zero (but it can still inhibit other units). Excitatory input from sensors or other units is summed: if this sum exceeds a specified inhibitory output threshold t_{io}, the unit produces an inhibitory output. Independently, the sum of excitatory inputs has uniform noise (distribution: $[-n, +n] \in \mathbf{R}$) added, and is then passed through an excitation transfer function, the result of which forms the excitatory output for that unit, so long as the unit has not been inhibited.

The excitation transfer function takes the form: $\mathcal{F}(x) = \mathcal{T}[(x - t_l)/(t_u - t_l)]$ where t_l and t_u are lower and upper threshold levels, and $\mathcal{T}[x'] = 0$ iff $x' \leq 0$; $\mathcal{T}[x'] = 1$ iff $x' \geq 1$; and $\mathcal{T}[x'] = x'$ otherwise. In most of our work, we have used: $t_l = 0.0$; $t_u = 2.0$; $t_{io} = 0.75$; and maximum noise $n = 0.1$; for all units.

The dynamic properties of these units are simulated using fine time-slice approximation techniques, with random variations in time-cycling to counteract periodic effects. To date, we have used unit weights and delays on all links in the network, and found that the dynamics produced are sufficiently sophisticated for current experiments. Nevertheless, we are actively investigating the use of variable weights and delays, placing these also under evolutionary control.

2.2 The Genetic Encoding

The network architecture has to be encoded as a 'genotype'. We have used a genetic encoding scheme which stores the "wiring diagram" (connectivity data) for the network as a string of alphanumeric characters. The encoding has been developed to be robust with respect to the mutation and crossover operators, where 'robustness' indicates that, given two parent genotypes encoding valid networks, an offspring genotype formed through crossover and mutation also encodes a valid network. For further details of the encoding, see [5]. A network is valid insofar as all the links in the network connect one unit to another: for each individual agent, the control network is initially randomly connected. It is our evolutionary learning algorithm, SAGA that develops these random networks into useful control architectures. This differs from other genetic algorithms in that it allows for variable-length genotypes, which allow for the *dimensionality* of the search space to be varied under evolutionary control; and the initially random population of individual genotypes converges, over evolutionary time, to a situation where the population is evolving as a *species*. See [4] for full details.

3 Evolving a Visually Guided Robot

Here we briefly present some recent results. We attempted to evolve networks for a simple adaptive behaviour, which was for a simulated[1] visually guided robot to spend as much time as possible in the centre of a circular arena.

[1] The simulations involve accurate physical and kinematic models of a real robot constructed at Sussex. Vision was simulated using ray-tracing with anti-aliasing via 16-fold super-sampling. See [2] for further details.

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 39-44

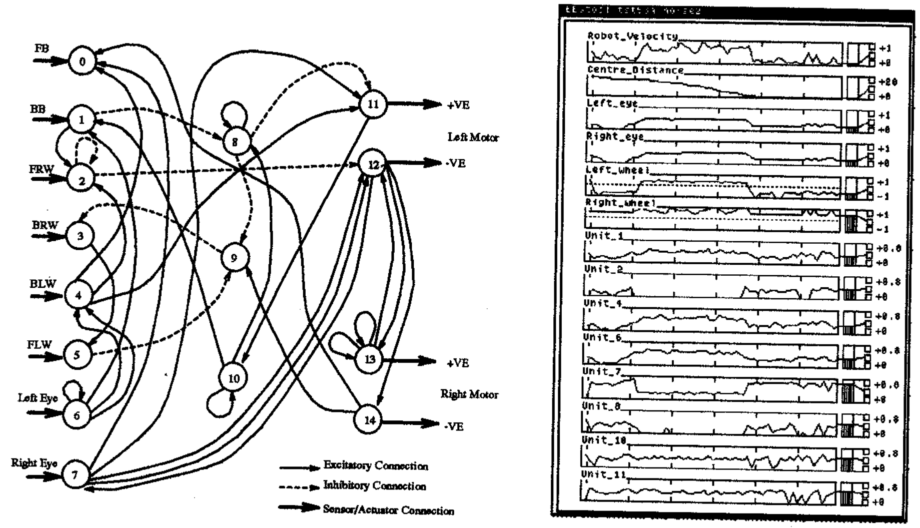

Figure 2: Evolved network and behavioural outputs. On left is final best-performing network, evolved over 100 generations. The left-hand units take inputs from bumpers, whiskers and eyes; the right-hand units control outputs to motors. Some may be redundant, e.g. unit 0 for the front bumper. There are excitatory and/or inhibitory feedback links to all units, including to inputs. Here, all links are of unit weight and unit time delay.
On right are graphs from monitoring the performance of one network over a single evaluation 'lifetime'. As is favoured by the evaluation function for this experiment, the distance from the centre of the arena decreases to 0, and the robot velocity then drops. The right wheel is moving forwards at all times. The left changes in direction; forwards early on, with some variation which steers the robot to the centre, it then reverses so that the robot twirls near the centre. The noise, which is a crucial part of the dynamics, can be seen in the activations of some units shown here.

Each individual robot was positioned at a randomly chosen point near the edge of the arena, in a random orientation. The robot then had a fixed finite 'lifetime', in which it had to get as close to the centre of the arena as it could, and then stay there. The robot's performance was evaluated by taking the gaussian function of a discrete temporal integral of its distance from the arena-centre during its lifetime: the more time the robot spent at or near the centre, the higher the score. The robots have two independent drive-wheels and a third free-wheel. The drive wheels may go at either full or half speed, either forwards or reverse, so the robot is capable of rotating on the spot, or travelling in circles of different radii, or in straight lines.

Each robot had 6 tactile sensors: two 'bumpers' (at front and back), and 4 radially symmetric 'whiskers'. The tactile sensors are primarily of use in detecting collisions with walls of the arena, and appropriately reorienting. The robot also has two directionally-sensitive photoreceptors, which allowed it to visually sense its environment (the walls of the simulated arena are dark, while the floor and ceiling are light).

We created a population of 60 robots with initially random genotypes, and

ESANN'1993 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 7-8-9 April 1993, D-Facto public., ISBN 2-9600049-0-6, pp. 39-44

evaluated each one over 8 'lifetimes'. At the end of the evaluation, we took the robot's *worst* score as a measure of its performance, to encourage robustness in the face of noise. When all 60 robots had been evaluated 8 times, the genotypes of the higher-scoring robots were 'inter-bred' using SAGA principles to create a new generation of 60 individuals. We repeated this process for 100 generations.

Results are shown in Figure 2. As can be seen, the network does not resemble the sort of networks which are traditionally published in the literature, but then this network was *not* designed by a human: it evolved according to Darwinian principles. Yet the graphs in figure 2 clearly indicate that the robot is approaching the centre of the arena and staying there. For full details and further analysis of the network, see [3].

4 Conclusion

This paper has concentrated on theoretical aspects of our work, which is motivated by concerns that prior network models may have been over-simplistic, and have not paid sufficient attention to the generation of adaptive behaviour. We have demonstrated that, using a neuron model with elementary dynamics, recurrent networks can exhibit rich dynamical activity in which noise plays a role, and they can be used for evolving controller networks that generate adaptive behaviour. The evolved networks have a distinctive appearance, in that they do not resemble networks designed by humans. As far as we know, we are the only research group who have successfully employed truly incremental evolution in creating dynamic recurrent networks for the generation of adaptive behaviour. We expect that our techniques will, as time progresses, become standard practice.

References

[1] D. T. Cliff. Computational neuroethology: A provisional manifesto. In J.-A. Meyer and S. W. Wilson, eds, *Proc. First Intnl. Conf. Simulation of Adaptive Behavior (SAB90)*, pp.29–39, Cambridge MA, MIT Press, 1991.

[2] D. T. Cliff, P. Husbands, and I. Harvey. Evolving visually guided robots. In *Proc. Second Intnl. Conf. Simulation of Adaptive Behaviour (SAB92)*. Cambridge MA, MIT Press, in press.

[3] D. T. Cliff, I. Harvey, and P. Husbands. Incremental evolution of neural network architectures for adaptive behaviour. Technical Report CSRP 256, University of Sussex School of Cognitive and Computing Sciences, 1992.

[4] I. Harvey. Species Adaptation Genetic Algorithms: A basis for a continuing SAGA. In F.J. Varela and P. Bourgine, eds, *Proc. First Euro. Conf. Artificial Life (ECAL91)*, pp.346–354. Cambridge MA, MIT Press, 1992.

[5] I. Harvey, P. Husbands, and D. Cliff. Issues in evolutionary robotics. In *Proc. Second Intnl. Conf. Simulation of Adaptive Behaviour (SAB92)*. Cambridge MA, MIT Press, in press.