ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Learning State Prediction Using a Weightless Neural Explorer

Igor Aleksander[1] and Helen Morton[1,2]

[1]Imperial College - Department of Electrical and Electronic Engineering, London SW7 2BT - UK
[2]Brunel University – School of Social Sciences (Psychology), Uxbridge, Middlesex, UB8 3PH - UK

Abstract. A weightless neural state machine acting as an exploratory automaton changes its position in a simulated toy world by its own actions. A popular question is asked: how might the automaton 'become conscious of' the effect of its own actions? Here we develop previously defined iconic learning in such weightless machines so that this knowledge can be achieved. Weightlessness, iconic learning are expressed in terms of state equations. Experimental results that show the conditions under which correct predictions can be obtained on a neural simulator are presented. Issues of information integration and memory implication are briefly considered at the end of the paper.

1 Introduction: Iconic Training then and Now

In an earlier paper (Aleksander and Morton, [1])i t was shown that the 'iconic' method of training weightless neural state machines could transfer perceptual/visual information into the states of the machine. We argued that this results in a system that has a phenomenally conscious state structure where events in the perceptual world form attractors of meaningful (potentially leading to action)states. In this paper we investigate how such a system can learn to *predict* the outcome of its own actions to anticipate the world input.

2 The Scenario

2.1 Finite state machines

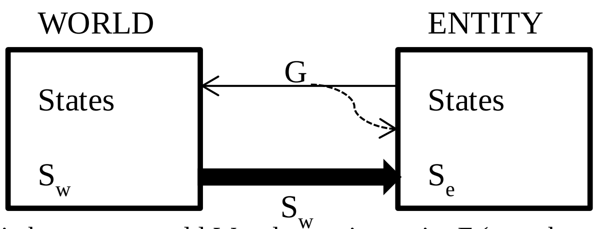

Figure 1: Relationship between a world W and an acting entity E (note that, the dashed line is relevant to section 4.2)

We define a world W as a discrete finite-state machine with the following state dynamic:

$$S'_w = f_w(S_w \times G) \qquad (1)$$

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

where S'_w is the set of states at time $t+1$ (i.e. the 'next' state) and S_w is the set of states at time t (i.e. the current state). Also G is a finite set of actions applied to this world. $\times$ is a Cartesian product. We then define an entity E which is the agent capable of acting on W. E is a learning finite-state machine (learning refers to the updating of f_e from examples), with the state dynamic :

$$S'_e = f_e(S_e \times S_w) \tag{2}$$

Note that this assumes that W is completely observable by E (i.e. there are no hidden internal variables).

The action output is $$G = f_g(S_e) \tag{3}$$

In the material presented below we use the notation

$$s_j \in S_j$$

2.2 A Neural State Machine

A finite state machine such as E becomes a neural state machine when every state variable in S_e is the output of a *weightless* neuron. This is also true of each variable of G. Such machines are generally *iconically* trained. For completeness we summarise below the *weightless* and *iconic* concepts, used in this paper, while details may be found elsewhere (e.g. Aleksander and Morton [2])

2.3 Weightless Neurons

The weightless neurons used in this paper have n binary inputs and one binary output. Training is a process where the input vector and the output bit (b) are stored as an ($n+1$)- bit vector in a binary memory of the neuron. A training set X consists of $/X/$ such distinct vectors, say $\{x_1, x_2 \ldots x_{/X/}\}$ When learning is over and the neuron has to classify an unknown input vector $\boldsymbol{i}_j$ (i.e. the *jth* training vector), this is compared in the neuron to all stored vectors in X. If $\boldsymbol{i}_j$ is closer in Hamming distance to one of these rather than others, the stored b forms the neuron output. This gives the neuron a 'nearest-neighbour' form of generalisation. on what has been learned. If there is a contention in the sense that in the set of nearest Hamming-distance vectors not all b values are the same, the output of the neuron is assigned a 0 or a 1 arbitrarily, with equal probability. This also occurs if the nearest stored vector has a Hamming distance greater than some threshold selected by the experimenter.

2.4 Iconic Training

This applies to neurons acting as state variables in a neural state machine in which it is assumed that the dimensions of the state are the same as the dimensions of the input. Then each state variable is associated with a single variable of the input. This variable controls the b value of one neuron and associates this with the input vector of that neuron. This scheme can be used to create attractors in the state space of the neural automaton such that the

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

attractors represent and, therefore, reconstruct for partial inputs, the input images on which the system is trained. Attractors are created by training an iconically created state to return to itself for a set of inputs (often noise) present at the input during training. That is,

$$((s_e)_t, (s_w)_{t+1}) \rightarrow (s_w)_{t+1} \qquad (4)$$

This becomes one of the entries for f_e considered as a table that defines (2)

3 Prediction

At a minimal level, prediction is defined as E at time t being in a state which predicts the state of W at $(t+1)$, that is:

$$(s_e)_t \rightarrow (s_w)_{t+1} \qquad (5)$$

A problem arises if by iconic training $(s_e)_t = (s_w)_t$ because (from (5)) prediction could happen only were the state of E re-entrant, that is when its action on the world would produce nil change. But, were E 'aware' of its own action associated with $(s_e)_t$, that is, it could form $(s_e, g)_t$ which, because of iconic training would become $(s_w, g)_t$ which, from (1) and further iconic training the network can learn. At this point, iconic training becomes a time-dependent process, and we feel it is best to proceed by illustrating the idea with a simulation.

4 Simulation of a predictive system

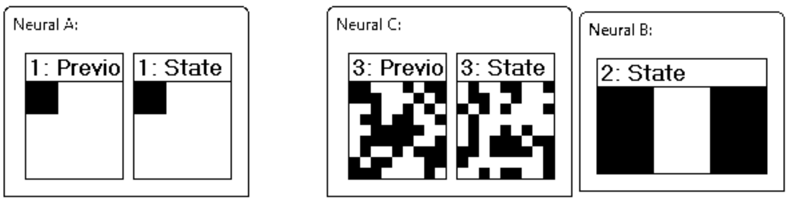

Figure 2: Screen shot of am NRM: Neural A is a model of the world, Neural C is the exploring entity and neural B is the Action output from the entity to the world.

The Neural Representation Modeller (See, I.Aleksander et.al. [3]) was used to simulate a world and an exporing entity. A screen shot of the arrangement is shown in figure 2. Two state machines are involved Neural A which is the world and neural B/C which are the entity. In such machines 'previo' (standing for 'previous') represents $(s_x)_{t-1}$ and 'state' represents $(s_x)_t$. In the entity, C is a 3-bit machine that can be in 8 states. It acts on the world by means of 5 actions, G (as in equation (3)) = {*(u)p, (d)own, (l)eft, (r)ight, (s)tand-still* }. Initially this machine acts arbitrarily and autonomously (that is, f_g is degenerate). That is, The action outputs occur (for system convenience and without implications on the results) with a probability of 1/8 for the first four and ½ for the fifth. The world (Neural A, where A) is a 9x9 array of weightless neurons, The position of the entity (state of the world) is indicated by a 3x3 black square for which there are 9 non-overlapping positions. The world responds in one time step to the action messages listed above. The random action causes the entity (black

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

square) to perform a random walk in the 9x9 space, which is made toroidal. For example, a u control input in the state shown in fig 2 would cause the next world state to have the entity (black square) in the bottom, right-hand corner of the space.

The world (Neural C) is also a 9x9 network of weightless neurons. This is a fully connected recursive network with an as yet unspecified connection to the world either as input or output.

4.1 Iconic learning to predict

The obvious way to predict the next state of the world is for the entity to 'become aware' of its own action as well as the effect that this has on the world. In terms of figure 1 this means that whatever $g \in G$ is being generated by the entity (continuing with arbitrary action selection for the moment) the next state of the entity will be a function of the intended action and the current state of the world. That is, equation (4) becomes modified to

$$((s_e)_t, (g)_t) \rightarrow (s_w)_{t+1} \tag{6}$$

As exploration develops, internal states that are not iconic get trained out and (6) becomes

$$((s_w)_t, (g)_t) \rightarrow (s_w)_{t+1}$$

Which means that the results of the entity's exploration have been internalised indicating that the entity knows what will happen from proprioception of its own actions. In summary, looking at figure 2, training proceeds as follows to obtain the results below i) generate an arbitrary action, ii) let the entity attempt a prediction and record this, iii) let the world take its next step, iv) correct the prediction and return to (i).

4.2 Results

In summary, looking at figure 2, training proceeds as follows to obtain the results below: i) generate an arbitrary action, ii) let the entity attempt a prediction and record this, iii) let the world take its next step, iv) correct the prediction by iconic training and return to (i). The recording of the prediction and the actual state of the world are shown in figure 4.

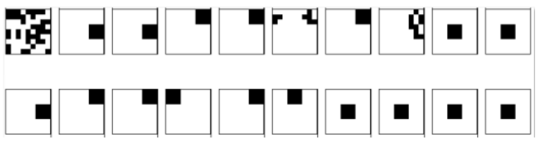

Figure 4: The upper row shows the first 10 steps in which the entity attempts to predict the next world state before learning what it actually which is shown in the lower row. The prediction score is 3/10

The first competence learned by the system is the language in which the world states are represented, that is a black 3x3 square in 9 legitimate positions. The second is the prediction

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

of the world state itself. In the 10 learning steps above, the 'language' performance is correct in 7 out of 10 cases, and remains 10/10 in further trials. The prediction performance is only 3/10 correct in the first run and typically develops as follows:

Run:	1	2	3	4	5	6	7	8	9	10	….
Pred./10:	3	5	7	4	9	8	10	10	10	10	….

We say that 'maturity' has been achieved after 7 runs, that is, after 70 training steps, the entity can predict the effect of its own actions

4.3 Connectedness and 'becoming conscious'

All the above is achieved with a fully connected recursive net. Much current discussion focuses on Information Integration issues in which the ability of less-connected systems to 'become conscious' of anything is shown to depend connectedness [4]. To examine this we have repeated the above experiments with lower percentages of internal interconnection and observed that correct operation still takes place but with much longer times to maturation and with noise entering into the predictive state lowering the predictive confidence. Fig. 5 shows the nature of the prediction. Confidence is 1-(average % noise for the whole run) Connectedness has cost implications in terms of the memory required by the neural state machines. Below we show how performance in terms of training steps to prediction and ultimate prediction confidence and amount of weightless system memory relate. The memory requirement for an n input neuron system is: $(no.\,of\ neurons) \times (n+1) \times (steps\ to\ full\ prediction)$.

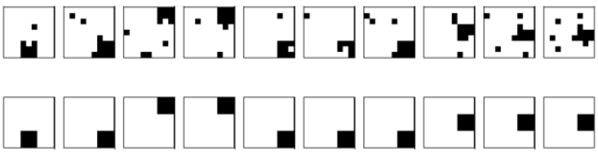

Figure 5: Result after 37 training runs (370 steps). Connectedness reduced from 81 full internal connections per neuron to 10 randomly chosen ones. The confidence level is 0.94

Inputs per neuron	81 (100% conn)	40 (50% conn)	20(25% conn)	10(12.5%conn)
Runs and confidence.	7 , 100%	17, 100%	25, 99.53%	37, 94.32%
Memory (bits)	464,940	564,570	405,000	329,670

At 5 inputs per neuron saturation occurs in the neuron training and the system collapses before reaching full prediction.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

5 Discussion

This exercise is part of a current series of investigations of the way that an artificial entity (in this case a weightless neural sate machine) can become conscious of the effect of its actions in its world. This is a precursor to current work on action selection and emotion-based planning . We have focused on explaining the formal nature and training of the neural state machine without intending to contribute to the existing corpus of work on series prediction or robot exploration. The key issue has been the introduction of proprioception as an influence on the generation of the 'next state' within the entity. We have shown that the collection of next world states can be predicted by merely allowing the entity to generate random action signals and observe the result of its actions entirely autonomously, without the intervention of an experimenter. Starting with a fully connected net acting on a 'toy' world, we showed that the internal feedback in the entity machine, can be reduced to 12.5% of full connection and still maintain performance, albeit by introducing noise into the predictions. We have not attempted a prediction of the measured performance as this is our first publication on this topic. Such analysis is part of current and future work.

References

[1] I. Aleksander and H. B. Morton, Phenomenal weightless machines, *ESANN 2009 proceedings, 22-24 April, 2009, pp307-312.*

[2] I. Aleksander and H. B. Morton: Aristotle's Laptop: Discovering our Informational Minds, *World Scientific Press,* (2012).

[3] I. Aleksander et al. , A brief introduction to weightless neural systems. *ESANN 2009 proceedings, 22-24 April, 2009,*

[4] G. Tononi, Consciousness as Integrated Information: a provisional manifesto. *Biological Bulletin.* 215, (2008) PP 216-242