Self-organising connected dynamical systems for learning temporal dependencies

Karla Parussel

Abstract. A novel architecture for a self-organising agent controller is presented here. It consists of a network of self-organising systems that can interact with one another. Each node in the network is implemented as a dynamical system with a state coordinate that traverses a dynamic energy landscape formed from attractors whose strength and position change over time. Connections exist between attractors of different landscapes allowing the dynamical systems to perturb one another as they settle into a stable state. The agent controller is given cost and reward actions that, if successfully used, change the strength of subsequent input signals. The controller is also given a set of enabler actions that must be used for specific cost and reward actions to have an effect in the next turn. The cost and reward actions have no effect if the specific enabler action they rely upon is not used in the previous turn. The enabler actions that the cost and reward actions rely upon changes randomly over time.

1 Motivation

The brain can be understood as a self-organising system, [15] [25]. The way that a brain functions internally is not directly determined by an external controller. Instead the brain reacts to sensory signals. It is an open system that senses the environment and acts within it, thereby changing it.

Living organisms are made up of cells and their evolution has been shaped accordingly. These systems are radically different to the computer architectures that are used for processing biologically inspired models such as artificial neural networks.

The motivation for the work presented here was to find an abstract model that was more suitable for processing on a computer, whether because it is more efficient or because it is easier to understand and reason about. The aim was to provide an abstraction of the underlying computation or process inherent in many self-organising systems rather than to claim biological plausibility. By modelling our understanding of how self-organising systems function we can verify and clarify our assumptions about how real-world systems function.

The idea of using a single dynamical system for autonomous agents and embedded cognition is not new [8]. But no self-organising system is completely self-contained. Every self-organising system, and also every directed system, is ultimately one of many parts of a larger self-organising system. The theory of autopoiesis [19] holds that there is no real boundary between an agent and its environment because the agent is a part of its environment. Beer describes how the nervous system of an agent, its body and the environment it inhabits can be considered coupled dynamical systems [2] [3].

Olds and Milner were able to influence the actions of a rat in a skinner box by injecting a current into its pleasure centres [12]. It could be argued that the rat's brain was no longer self-organising, but the rat, scientists, lever and apparatus injecting the current together formed a self-organising system. We are all part of a society, world, solar system and galaxy that is each self-organising because they all act in accordance with the laws of thermodynamics [5] [6] [7] [13].

Even though a self-organising system can be made up of constituent components which are themselves self-organising, each component can still be considered a distinct entity, for example, an economy is formed from a network of buyers and sellers. What then is the nature of the connections between self-organising systems?

Real-world open self-organising systems settle into stable states by minimising the input of free energy. Energy leaves the system in a higher entropic state than when it entered [11] [23]. Energy flows can take many different forms. For example money can be considered a currency of free energy in an economy because it enables work to be performed and can be converted to other forms of real-world energy. The term free energy is used here in its broadest sense to refer to any energy that can perform work because of the presence of a thermodynamic gradient.

If each system is ultimately part of a larger self-organising system, then conversely, it is possible to reduce some self-organising systems into networks of smaller systems that are connected by energy flows. For example, distinct regions in the brain interact with each other via connections from pyramidal neurons. A biologically plausible model of a neuron receives inputs of energy which leak away over time, but are expelled as action potentials if it reaches a voltage threshold [26] [17]. Dendritic self-organisation can be explained as the minimisation of free energy [16].

We can consider a pyramidal projection from one area of the brain to another to be persistent whereas the synaptic connections between cells in a neuronal network as transient. This is because the latter connections strengthen and weaken over time due to synaptic drift and weight change. There is effectively little difference between two neurons which are not connected and two joined by a synapse that is unlikely to release a neurotransmitter. A self-organising system containing persistent connections might be better modelled using a network of smaller systems containing only transient connections. This could be a modelled as a network of dynamical

systems that would then both perturb and be perturbed by other systems that they are connected to.

The connections between dynamical systems in a network would act as energy flows between self-organising systems. Light and heat from the sun to the earth, photons reaching the eyes or signals being carried along pyramidal projections from one area to another, or connections between interneurons within the same circuit, are all examples of self-organising systems that perturb each other via the flow of energy.

2 Model

The architecture presented here consists of a network whereby each node is an individual self-organising system modelled by a dynamical system. These systems attempt to settle internally into stable states, but by doing so can disturb other systems external to themselves via connections, which are also attempting to settle into stable states.

The network of dynamical systems is used here as an agent controller. One system is designated the output node and determines which actions are to be performed. Sensory stimuli can be injected into any system as a collection of real values.

2.1 Methodology

A useful working definition of intelligence is the ability to adapt to an unknown environment. If the environment is fully known in advance then an agent can perform equally well, or better, by merely following hard coded rules. It is difficult to argue that such an agent is acting intelligently.

Self organising systems are made up of many components that can be arranged in a myriad of different ways. How they self-organise in practice is determined by the flow of free energy that its environment provides.

Therefore an intelligent self-organising agent must be able to adapt to an unknown function external to itself. This function can be an external environment for an embodied agent, or a particular task that a system is applied to. It cannot be assumed a-priori how an intelligent self-organising agent can most effectively adapt because it is in effect performing a search of which of its internal states is the most stable given a particular input. For the same reason it cannot be assumed in advance which components or mechanisms will be required for a self-organising intelligent agent to most successfully adapt to an unknown task or environment.

148 different versions of the code were tested for how well they performed a variety of different experiments. The code was evolved each time for different experiments and stopped once an initial idea was obtained as to how well it performed. In this way different algorithms or choice of functions could be compared. If they did not show a general increase in performance then they were discarded. Each evolutionary run could take a day or longer to provide an initial idea of fitness, and many weeks to fully complete.

The system was designed so that each dynamical system could settle into a stable state, but by doing so could disturb other systems with the network of systems either eventually finding a compromise, or oscillating between stable states. Consideration was given as to how a system might need to adapt by modulating its own properties. Algorithms and functions that required fewer parameters to be optimised were preferred because it meant that evolutionary runs would be shorter. Algorithms also needed to be consistent with the overall concept to make it easier to understand how the model was adapting. One of the motivations for discarding biological plausibility was to make an adaptive system more easily understood, reasoned about and engineered. If a bug in the code meant that it was not functioning as intended, then it was corrected even if this led to a decrease in fitness. This was because it could have been compensating for other bugs elsewhere in the code. It is expected that there will be ways in which the model can be further optimised and extended to be applied to other tasks. It is also possible that ideas that were previously discarded may find a use when the model is applied to other tasks.

2.2 Energy landscape

Each dynamical system is effectively an energy landscape [11] [10, pp21]. Kauffman uses the landscape concept to describe dynamical systems as consisting of attractors, disjoint from each other in state space, acting as lakes with drainage basins [14, pp 176].

Using an analogy of a ball rolling along a peak, ridge or plateau, given sufficient energy it will roll down a slope and minimise its own potential energy. The ball will not be able to later return unless its kinetic energy is first increased. This process will continue until the ball comes to a stop at the bottom of the landscape, or within a local depression that requires more kinetic energy than the ball currently has for it to escape. Valleys correspond to attractors in a dynamical system, the speed that the system moves into them being determined by the steepness of the slope.

The space in which the energy landscape resides is implemented as a continuous toroidal space of three dimensions. Any location within it can be pin-pointed using a coordinate where each dimension is within the range $[0 : 1]$. Attractors are placed randomly within the space to form the energy landscape. The state of the system is specified using a state coordinate which has a velocity. Attractors influence the trajectory of the state coordinate within the space by pulling it towards themselves. Each cycle, the dynamical systems process input signals, update the position of their state coordinate, send output signals and then update their internal state.

Calculations involving two coordinates within the toroidal space always use the shortest Euclidean distance possible. This is achieved by comparing the absolute difference of each dimension between one coordinate and the other. If the absolute difference is greater than 0.5 then $-/+1$ is added to reduce it.

Each attractor has a strength value and this affects how strongly it can pull the state coordinate. The strength parameter changes over time and can also become negative whereby the attractor will act as a repulsor pushing the state coordinate away. Attractor strength is constrained to be in the range $[-1 : 1]$.

Attractor strength decays over time. It can either be increased by input signals arriving via connections or directly injected in. Directly injected signals modify attractor strength before those from input connections. Signals directly injected I are multiplied by an evolved scaling parameter and applied to attractor strength A_S, either by $A_S = \bar{I}$, $A_S + \bar{I}$

or $(A_S + \bar{I})/2$, the choice of which is evolved.

2.3 Connections

Attractors can send signals out over connections. Connections can be either excitatory or inhibitory. Inhibitory connections invert their signals to be negative. Connections are not weighted. Each output connection originates from a source attractor.

Signal strength is determined by distance of the state coordinate to the centre of the source attractor. The stronger the attractor, the more it can attract the state coordinate and the stronger its output signal. Signal strength is calculated as $1.0 - normalise(m)$, where m is the Euclidean distance between the position of the attractor A_P and the position of the state coordinate $\bullet S_P$, normalised to be in the range $[0 : 1]$. $\bullet S_P$ has a shorter Euclidean distance to A_P than S_P using the method described above.

There are various types of connections between dynamical systems with each type having a different effect. Some types modify a property of a specific attractor in the target system; strength (*StrengthSignal*, *CopyStrengthSignal*) and receptivity to input (*ReceptivitySignal*). Other connection types change the properties of the entire target system; acceleration of the state coordinate (*GainSignal*), the rate at which attractors move (*AttractorMovementSignal*) and whether connections should change the attractors they connect to (*ConnectionLearningSignal*) and whether the output signals of the target system should be inhibited (*OutputInhibitionSignal*). The average signal strength is calculated for each connection type.

There are three forms of connectivity between source and target dynamical systems. One to one connectivity between each corresponding attractor, full connectivity from each source attractor to each target attractor, and sparse connectivity which is like full connectivity but with an evolved probability of a connection being created.

2.4 Updating state

The position of the state coordinate is then updated. The direction to move the state coordinate is the unit vector $\hat{s} = normalise(\bullet A_P - S_P)$.

The unit vector is then scaled up to account for strength of attraction and how close the state coordinate is to the attractor. If the inverse square law were to be used to emulate the effect of gravity then it would produce numbers within the range $[1 : \infty]$, whereas what is required are numbers $[0 : 1]$. So instead attraction of the state coordinate is calculated as $S_a = \hat{s} \cdot (1 - n)^2 \cdot \bar{A_S}$, where $\bar{A_S}$ is attractor strength multiplied by an evolved scaling parameter and n is the Euclidean distance between $\bullet A_P$ and S_P normalised to be in the range $[0 : 1]$.

An evolved decay rate is multiplied by n and applied to the strength of each attractor to act as a form of habituation. The closer the attractor is to the state coordinate the faster its strength decays.

The attraction of the state coordinate to each attractor is added to the state coordinate's velocity as $normalise(\sum S_a) \cdot C_s$, where the sum of attraction is normalised to be a unit vector and C_s is an evolved constant acceleration for the state coordinate. If this results in values outside the range $[-1 : 1]$, then the velocity is constrained by normalising it so that the largest dimension has an absolute value of 1.

The new velocity is then multiplied by a gain modifier set to the mean of input signals of type *GainSignal* if any are present. The velocity is added to the state coordinate before being multiplied by a genetically determined decay rate. The new state coordinate is wrapped to keep it within the toroidal space.

2.4.1 Sending outputs

Dynamical systems then send signals over their output connections to be processed in the next cycle. If the output connection is of type *CopyStrength* then the strength of the source attractor is used instead without relation to its distance from the state coordinate. This is added to the strength of the target attractor as if the connection was of type *StrengthSignal*.

Dynamical systems can inhibit the output signals of other systems if they connect using *OutputInhibitionSignal* connections. After the output force of an attractor is calculated, it is multiplied by the output force inhibition modifier. At the beginning of each cycle the modifier is set to 1 and ordinarily has no effect. Unlike the other connection types which calculate the average signal strength, *OutputInhibitionSignal* connections add their excitatory or inhibitory signal strengths to this modifier before the modifier is thresholded to be within the range $[0 : 1]$.

2.4.2 Updating connections

Once the input and output signals are processed, the internal state of each dynamical system is updated. Even though connections are not weighted, there is a pressure on them to change which attractors they connect to and from. Connections that update a property of the dynamical system cannot adapt in this way. The strength of the signal they send is compared with the strength of the target attractor, 0.5 or above is strong, below is weak.

- If both the target attractor and connection signal are strong then the pressure is reduced.
- If one strength is strong and the other weak then the pressure is increased.
- If both strengths are weak then no change is made.

The change to the connection-change pressure is randomised with a number drawn from $[0 : evolvedmaximum]$. If the connection is inhibitory then the change is inverted. For the purpose of updating connections,‘ when retrieving a signal strength from a connection the value returned is either 1 or 0 depending on whether the it is above or below an evolved threshold.

The connection pressure is multiplied by the connection learning modifier. This modifier is set to the mean of input signals of type *ConnectionLearningModifer* if any are present, otherwise it defaults to 1. If a random number drawn in the range $[0 : pressure]$ is greater than the evolved threshold then the pressure is reset and the connection connects to or from another attractor depending on whether it is an input or output connection. The new attractor to connect to is chosen at random.

2.4.3 Updating attractor position

Each attractor is pulled towards other attractors, or pushed away from attractors that have flipped to become repulsors. The total attraction and repulsion is summed up for each attractor before their positions are updated. Each attractor's position P_i is pulled towards or pushed away by every other attractor P_j. This is calculated to be a unit vector $\hat{a} = normalise(\sum((\bullet P_j - P_i)(R_j + R_i - 1)))$

R is the output force recorder function of an attractor. This decays alongside attractor strength using an evolved decay rate. When calculating the attractor output force when sending a signal over a connection, if it is greater than an threshold then an increase is added to the recorder function. Both threshold and increase are evolved.

The unit vector is added to the attractor's velocity as $\hat{a} \cdot C_a$ where C_a is an evolved constant acceleration. If this results in values outside the range $[-1:1]$, then the new velocity is normalised so that the largest dimension has an absolute value of 1. The velocity is then multiplied by an attractor movement modifier. This modifier is set to the mean of input signals of type *AttractorMovementSignal* if any are present, otherwise it defaults to 1. The velocity is added to the attractor's coordinate before being multiplied by a genetically determined decay rate. The coordinate is wrapped to keep it within the toroidal space.

Attractor strength is used when calculating attraction of the state coordinate, the output force recorder function is used when calculating attraction of other attractors.

2.4.4 Scaling

The range of values used for describing distance, attractor- and signal strength were chosen to be in the range $[0:1]$ for practical reasons. There is nothing inherent in the model that makes these ranges relevant to one another. Therefore scale parameters are evolved so they can be applied to one another. A pair of scale parameters are evolved together, a base $[0:2]$ raised to the power of an exponent. When calculating attraction of an attractor on a state coordinate, the attractor strength is scaled before being applied to the unit vector. Directly injected inputs are also scaled before modifying or setting attractor strength. Connections which modify global parameters of whole dynamical systems (*GainSignal*, *AttractorMovementSignal*, *ConnectionLearningSignal*, *OutputInhibitionSignal*) have the signal they deliver scaled by a single real parameter in the range $[0:2]$. Connections between attractors (*StrengthSignal*, *CopyStrengthSignal*, *ReceptivitySignal*) are also scaled by another parameter in the range $[0:2]$.

2.4.5 Pruning

The network needs to be pruned before first use. Pruning makes the network more efficient and easier to visualise. All systems with output connections that ultimately connect to the output system are marked as active. Inactive systems are then removed from the network along with their input and output connections.

Depending on how it has evolved, some dynamical systems will not receive any sensory signals either by direct injection or via input connections from other systems. These systems may be active and have output connections that affect the average input signal in other systems. It was found that pruning these systems reduced performance but also allowed the agent controller to differentiate more effectively between sub-optimal actions. Only the inactive systems were pruned when obtaining the results in section 4.

2.5 Parameter Optimisation

The parameters of the networks are optimised using an evolutionary algorithm. Once these evolutionary runs are finished the parameters are used to generate and test a population of 450 agents in order to determine the average performance of the model. An average fitness is required because the mapping from genotype to phenotype is stochastic.

The fitness function used during parameter optimisation was $(2 \cdot Resource) + Age$. Age is important for the fitness function during parameter optimisation when agents are more likely to die before the end of their evaluation. Each agent was tested for a consistent number of cycles unless it died prematurely.

3 Experiments

The artificial life animat concept was abstracted to provide the simplest possible context for testing the model. An agent was created that can neither sense an environment nor be affected by one. It only interacts with a body that contains a resource level (see figure 1). An example of a network of dynamical systems can be seen in figure 2.

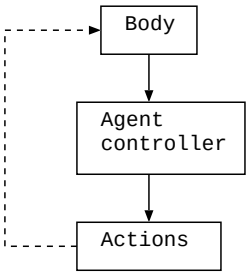

Figure 1. The agent controller receives input signals derived from the state of the body. It then attempts to choose one action to be performed. The action directly alters the body state of the agent. This leads to different input signals being passed to the agent controller in the next turn.

Each change in resource level was passed to the agent controller as an input signal. Before being input, they were scaled to the largest increase and decrease that had occurred to each resource so as to be within the range $[0:1]$. They were then inverted so that desirable changes, such as increases to a resource level, resulted in a reduced signal to the agent controller. This allows the network to act as a minimal disturbance system as it settles upon actions that reduce its total input activation.

Each attractor in the output dynamical system corresponded to a different action. At the end of a cycle, the action

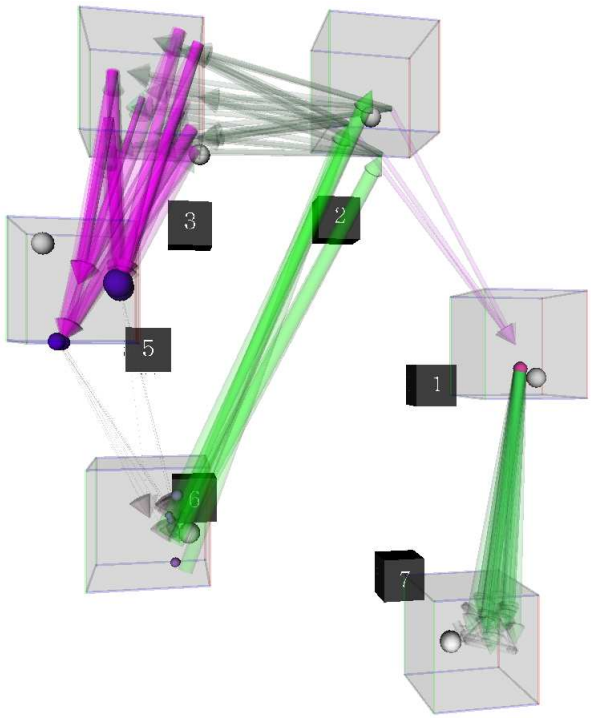

Figure 2. The semi-transparent cubes with edges each represent a single dynamical system. Each is a 3D space containing attractors (coloured spheres) and a single state coordinate (white sphere). Connections between attractors, or attractors and systems, are represented with semi-transparent arrows. The width of the connections and the size of each attractor signifies its current strength. Only the active systems are shown. The dark cubes label each system. The seventh dynamical system has a recurrent connection to and from the same system.

of the attractor that is closest to the state coordinate was performed. If there was a tie then then one of the winners was chosen at random.

How does intelligence differ from other natural self-organising phenomena? It is proposed here that as with all self-organising systems, free energy is minimised so the system can settle into a stable state. But in the case of intelligent systems, unlike a stimulus / response agent, neutral, or even costly actions can be performed if it means that the minimisation of free energy is greater when averaged over time. By this means an intelligent system is more able to escape local minima within an energy landscape. If correct, this would suggest that if it is possible for an intelligent system to develop, then it is likely to given sufficient time because such systems can settle into states that are more stable.

The agent controller was given the task of learning temporal dependencies to test whether it could learn to perform a sequence of neutral actions that subsequently enabled a cost or reward. It was given a set of four reward and four cost actions that increased or decreased the resource levels by 0.5/1.0/1.5/2.0 points respectively. The agent was also given eight enabler actions that had no effect except to allow a cost or reward action to have an effect if it was used in the next cycle. If the corresponding enabler action had not been used then the cost / reward action would have no effect.

There was a 0.1% chance for each cost / reward action at the end of the cycle to change which enabler action it depended upon to be allowed to be successfully performed. Enabler actions were chosen at random so it was possible that some enabler actions could allow more than one other cost / reward action while some enabler actions allowed none.

4 Results

During the evolutionary run, agents were run for 2,000 cycles before a fitness score was added to the running average of its genotype. At the end of the evolutionary run, all the genotypes in the population were evaluated the same number of times and the best one was selected. This was then tested in a run lasting 1,000,000 cycles with the fittest genotype used to create a population of 450 agents. The results are slightly different for each agent because there is a stochastic mapping between genotype and phenotype. The genotype was tested for an extended period because cost and reward actions change which enabler action they rely upon randomly over time. Running it for 1,000,000 cycles means that we can rule out luck when evaluating. Fitness levels plotted for each agent in the population show that all agents were able to adapt successfully 3.

The results for the population of agents expressed from the best genotype are in table 1. The first eight actions are neutral and enable one, some or none of the next eight cost or reward actions. The third column is the frequency with which the action was performed, with all values adding up to 100%. The fourth column specifies the success rate the agents had when performing that action. If an action was successfully performed because the enabler action that it was reliant upon was performed in the previous cycle, then both its success rate and that of the enabler action it is reliant upon were incremented.

The most salient result is that the best reward action was

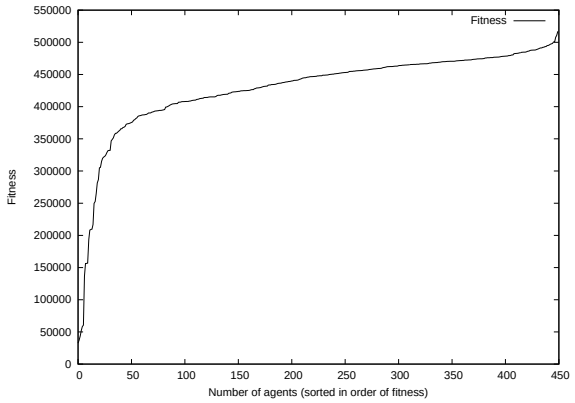

Figure 3. Figure sorted in order of ascending fitness. The run lasted 1,000,000 cycles. The results are slightly different for each agent because there is a stochastic mapping between genotype and phenotype.

attempted more frequently than the other actions (25%) and more often successfully (84% of the time). The enabler actions were attempted more often than the cost and reward actions, and the reward actions marginally more often than the cost actions. But the agents learnt to perform the reward actions successfully more often than the cost actions.

Action	Amt	Freq	Success
Enabler	0	6.67%	48.89%
Enabler	0	6.17%	44.28%
Enabler	0	6.24%	44.86%
Enabler	0	6.08%	43.80%
Enabler	0	6.73%	49.18%
Enabler	0	5.57%	38.69%
Enabler	0	6.40%	46.82%
Enabler	0	6.21%	44.84%
Cost	-2.0	3.45%	4.17%
Cost	-1.5	3.45%	418%
Cost	-1.0	3.44%	4.19%
Cost	-0.5	3.44%	4.16%
Reward	0.5	3.62%	6.82%
Reward	1.0	3.61%	6.84%
Reward	1.5	3.62%	7.00%
Reward	2.0	25.30%	84.49%

Table 1. Average frequency of enabler, cost and reward actions chosen by a population of 450 agents and how successfully they were used.

The experiment was repeated using the same genotype, but this time a base cost of -0.5 was applied when using an action. If the cost or reward actions were used successfully then the base cost was not applied. The two smallest cost and reward actions (-/+ 0.5 respectively) were replaced with two non-sequence neutral actions which never applied a base cost. An agent could settle into a stable state by using only these two actions, but if it were to reduce the input signal even more, it would first have to perform another action that increased it. The results are shown in table 2.

The agents did not perform the neutral non-sequence action any more frequently than any of the other actions. The greatest reward action was performed most frequently and successfully even though this meant that the agents first needed to perform a sequence action that carried a base cost.

Action	Amt	Freq	Success
Enabler	-0.5	6.27%	10.52%
Enabler	-0.5	6.31%	11.60%
Enabler	-0.5	6.25%	10.80%
Enabler	-0.5	6.17%	9.56%
Enabler	-0.5	6.18%	9.91%
Enabler	-0.5	6.20%	10.05%
Enabler	-0.5	6.40%	12.73%
Enabler	-0.5	6.23%	9.89%
Cost	-2.0	5.80%	6.60%
Cost	-1.5	5.79%	6.62%
Cost	-1.0	5.80%	6.59%
Non-sequence	0	5.77%	100%
Non-sequence	0	5.76%	100%
Reward	1.0	5.80%	6.26%
Reward	1.5	5.77%	6.28%
Reward	2.0	9.53%	36.18%

Table 2. Average frequency of enabler, cost and reward actions with a base cost when unsuccessfully used. Also includes two non-sequence neutral actions.

5 Discussion

There is a paradox with life. If life is by nature thermodynamically far from equilibrium [4] [1] then how did it self-organise by minimising free energy? Life reduces the thermodynamic gradient between the hot sun and cold space [23, pp 8]. The existence of a thermodynamic gradient means that there is a pressure to perform work. The more fully the thermodynamic gradient can be exploited the more entropy is produced and the more the gradient is reduced over time.

Shrödinger used the concept of entropy and Gibbs free energy to describe how life increases entropy by the act of feeding. Food is relatively ordered and after being used to produce energy for the organism is returned in a simpler state to the environment [24]. Feeding and breeding allows for more entropy to be increased over time than a single agent that starved to death and decayed.

Maybe intelligence avoids the dark room problem [9] in the same way? Organisms seek to maximise their consumption of resources and chance of breeding rather than hide in a dark chamber avoiding surprises, but in so doing they increase entropy over the life-time of the agent. In both cases a thermodynamic pressure is succeeding in performing work, thereby increasing entropy, until the components on which it acts upon settle into a state stable enough to persist.

Although entropy is not represented in the dynamical systems described here, they do function by settling into stable, or metastable states. The stronger the reward, the weaker the directly injected input signal. In this way disturbance is minimised, which can be seen as equivalent to minimising free energy. The agent controller learnt to perform the correct neutral enabler action that would allow it to successfully perform a rewarding action, but it could also learn to perform a costly enabler action first so as to gain a greater reward later. In doing so the agents successfully avoided a local minima in the energy landscape and minimised more disturbance when averaged over time.

Information theory and thermodynamics share the concept of entropy, and an information theoretic approach to AI has

proven extremely useful. This approach though ignores the fact that brains are examples of self-organised physical systems that change their structure over time. Understanding how and why they do so will help us to understand how they function, and consequently, how artificial versions can be engineered.

For example when discussing the dark-room problem Friston et al. refer to free energy in terms of information theory. It could be argued though that surprise generates neural activity and the difference between expectation and actual sensory input could be understood as disturbance that needs to be minimised. The difference between an information focused and activity- or disturbance focused approach may be considered conceptual rather than actual, but as designers the concepts that we use determine how we reason about and engineer such systems.

Biologically inspired self-organising neural networks developed previously were explained as settling into a stable state by finding the lowest point on an energy landscape [22] [21] [20]. Modelling this understanding explicitly as a network of connected dynamical systems demonstrates that it is a useful concept when engineering self-organising systems.

LeDoux [18] pp16 describes a distinguishing characteristic of cognitive processing as flexibility of response to the environment. Emotions provide a counter-balance to this by narrowing the response of an agent in ways that have a greater evolutionary fitness. Based on this, [20] proposed that emotions can be defined in terms of whether they drive an agent out of a stable state or help enforce it.

An approach similarly inspired by non-equilibrium thermodynamics focused on the idea of activity as disturbance that can be minimised by a system settling into stable states may also be fruitful for modelling cognition.

REFERENCES

[1] John Avery, *Information Theory and Evolution*, World Scientific Publishing Co., 2004.

[2] Randall D. Beer, 'Dynamical approaches to cognitive science', *Trends in Cognitive Sciences*, **4**(3), 91–99, (2000).

[3] Randall D. Beer, 'Dynamical systems and embedded cognition', in *The Cambridge Handbook of Artificial Intelligence*, eds., K. Frankish and W. Ramsey, 128–150, Cambridge University Press., (2007).

[4] E.J. Chaisson, 'A unifying concept for astrobiology', *International Journal of Astrobiology*, **2**(2), 91–101, (2003).

[5] E.J. Chaisson, 'Non-equilibrium thermodynamics in an energy-rich universe', in *Non-equilibrium thermodynamics and the production of entropy*, 21–31, Springer-Verlag, (2005).

[6] Eric J. Chaisson, *The rise of complexity in nature*, Harvard University Press., 2001.

[7] Eric J. Chaisson, *Epic of evolution: seven ages of the cosmos*, Columbia University Press., 2006.

[8] Randy D.Beer, 'A dynamical systems perspective on autonomous agents', *Artificial Intelligence*, **72**, 173 – 215, (1995).

[9] Karl Friston, Christopher Thornton, and Andy Clark, 'Free-energy minimization and the dark-room problem', *Frontiers in Psychology*, **3**, 130, (2012).

[10] John Hertz, Anders Krogh, and Richard G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley Longman Publishing Co., Inc., 1991.

[11] Francis Heylighen, 'The science of self-organization and adaptivity', in *The encylopedia of life support systems*, 253–280, EOLSS Publishers, (2000).

[12] Olds James and Milner Peter, 'Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain.', *Journal of Comparative and Physiological Psychology*, **47**(6), 419427, (1954).

[13] E. Jantsch, *The Self Organizing Universe : Scientic and Human Implications*, Pergamon Press, New York, 1980.

[14] Stuart Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Pres, 1993.

[15] J. A. Scott Kelso, *Dynamic patterns: The self-organization of brain and behavior*, A Bradford book. The MIT Press., 1995.

[16] Stefan Kiebel and Karl Friston, 'Free energy and dendritic self-organization', *Frontiers in Systems Neuroscience*, **5**, 80, (2011).

[17] Christof Koch, *Biophysics of Computation*, Oxford University Press., 1999.

[18] Joseph E. LeDoux, *The Emotional Brain*, Simon & Schuster, 1998.

[19] Humbert R. Maturana and Francisco J. Varela, *Autopoiesis and Cognition: The realization of the living*, D Reidel Publishing Company, 1980.

[20] Karla M. Parussel, 'Emotion as a significant change in neural activity', *International Journal of Synthetic Emotions*, **1**(1), 51–67, (2010).

[21] Karla M. Parussel and Lola Cañamero, 'Biasing neural networks towards exploration or exploitation using neuromodulation.', in *ICANN 2007: Proceedings of the 17th International Conference on Artificial Neural Networks Part II*, eds., Joaquim Marques de Sá, Luís A Alexandre, Wlodzislaw Duch, and Danilo Mandic, volume 4669, pp. 889–898. Springer-Verlag, (2007).

[22] Karla M. Parussel and Leslie S. Smith, 'Cost minimisation and reward maximisation. a neuromodulating minimal disturbance system using anti-hebbian spike timing-dependent plasticity.', in *Proceedings of the Symposium on Agents that Want and Like: Motivational and Emotional roots of Cognition and Action at the AISB-05 conference*, pp. 98–101. The society for the study of artificial intelligence and the simulation of behaviour, (2005).

[23] Eric D. Schneider and Dorion Sagan, *Into the Cool: Energy Flow, Thermodynamics, and Life*, University Of Chicago Press, 2006.

[24] Erwin Schrödinger, *What is Life - The Physical Aspect of the Living Cell*, Cambridge University Press., 1944.

[25] Christoph von der Malsburg, *The Handbook of Brain Theory and Neural Networks*, chapter Self-Organization and the Brain, 1002–1005, The MIT Press, 2003.

[26] Udo Wehmeier, Dawei Dong, Christof Koch, and David van Essen, 'Modeling the mammalian visual system', in *Methods in Neuronal Modeling: From synapses to networks*, eds., Christof Koch and Idan Segev, chapter 10, 335–360, The MIT Press, (1989).