Biasing Neural Networks Towards Exploration or Exploitation Using Neuromodulation

Karla Parussel and Lola Cañamero

Adaptive Systems Research Group, School of Computer Science,
University of Hertfordshire, College Lane, Hatfield, Herts, AL10 9AB, U.K.
{K.M.Parussel,L.Canamero}@herts.ac.uk

Abstract. Taking neuromodulation as a mechanism underlying emotions, this paper investigates how such a mechanism can bias an artificial neural network towards exploration of new courses of action, as seems to be the case in positive emotions, or exploitation of known possibilities, as in negative emotions such as predatory fear. We use neural networks of spiking leaky integrate-and-fire neurons acting as minimal disturbance systems, and test them with continuous actions. The networks have to balance the activations of all their output neurons concurrently. We have found that having the middle layer modulate the output layer helps balance the activations of the output neurons. A second discovery is that when the network is modulated in this way, it performs better at tasks requiring the exploitation of actions that are found to be rewarding. This is complementary to previous findings where having the input layer modulate the middle layer biases the network towards exploration of alternative actions. We conclude that a network can be biased towards either exploration of exploitation depending on which layers are being modulated.

1 Introduction

In the brain, different levels of neuro-active substances modulate the sensitivity-to-input of neurons that have receptors for them [1, page 94]. Fellous [2] proposes that emotion can be seen as continuous patterns of neuromodulation of certain brain structures. Kelley [3] argues that in their broadest possible sense, emotions are required for any organism or species to survive. They allow animals to satisfy needs and act more effectively within their environment. She argues that emotions are derived from neurochemically coded systems. These systems have been present in one form or another throughout our evolutionary history. Emotions can be influenced by altering the levels of these neuromodulators in the nervous system.

Emotions also help the reasoning process [4]. Evans puts this idea in a game-theoretical framework in his search hypothesis [5], according to which, in a rational agent confronted to an open-ended and partially unknown environment, emotions constrain the range of outcomes to be considered and subjectively applies a utility to each. The search hypothesis can be seen as an example of an

J. Marques de Sá et al. (Eds.): ICANN, Part II, LNCS 4669, pp. 889–898, 2007.

agent moving from exploration of possible outcomes to an exploitation of the action providing the current expected highest expected utility. However, the best course of action does not need to be learnt through experience. Nesse [6] defines emotions as specialised states of operation that give an evolutionary advantage to an agent in particular situations. LeDoux [7] 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. As an example, predator avoidance driven by fear is an ideal behaviour to be selected for and optimised by evolution. It is a behaviour that needs to be maintained until the prey reaches assured safety regardless of whether it is able to continually sense the predator or not [8]. Nor will the prey benefit from being distracted by less important sensory input while it is still in danger. Successful fleeing behaviour might not require exploration of different actions when instead, exploitation of known strategies for a successful escape should be given priority. On the contrary, positive emotional states are thought to promote openness to the world and exploration of new courses of actions [9].

2 The Agent

We have used the simplest possible agent to test the effect of neuromodulation when applied to an artificial neural network, an agent that cannot directly sense its external environment. It can only sense two critical resources of its simulated body which it must maximise. These resources are referred to here as 'energy' and 'water'. The agent can execute a set of actions that either increase or decrease by a given amount the energy or water level in the body, plus two neutral actions. Neutral actions are useful because if they are used differently to each other then it throws doubt on how well the agent is adapting. The 'inactive' action is used by default when an agent does not choose for itself. This can happen if no activation reaches the output neurons of its neural network. It results in each resource of the agent being reduced by the maximum cost. The effect of this is more costly to the agent than if it deliberately chose the most costly action available to it as that would only result in a reduction of one resource.

2.1 The Neural Network

The agent adapts using a feed forward neural network of spiking leaky integrate-and-fire neurons based on the model described in [10] and [1, page 339]. The network learns which outputs should be most frequently and strongly fired to minimise the subsequent level of input signal in the next turn. Each neural network is made up of three distinct layers; input, middle and output layer. The network is iterated over a fixed number of times within a single turn.

For each resource, the input layer has two neurons that output to the middle layer. One neuron signals the need for the resource and the other neuron signals the satisfaction of that need. There are situations in which an effective behaviour

for an agent may be to decrease a need but not satisfy it. Alternatively there may be situations in which an agent needs to store more resources than it is used to doing. In these experiments the agent is tasked only with maximising its resources.

There is one output neuron per action. The action performed by the agent directly and immediately alters the level of a resource. This consequently determines the strength of the corresponding input signal fed to the network in the next turn. This is fed via the input neurons corresponding to the resource effected by the action. In this way the network acts as a minimal disturbance system [11] as it settles upon actions that reduce its total input activation.

2.2 The Neuron

Spiking neurons were used in the neural network, each one acting as a capacitor to integrate and contain the charge delivered by synaptic input. This charge slowly leaks away over time. The neurons have a fixed voltage threshold and base leakage which are genetically determined.

The neurons also have an adaptive leakage to account for how frequently they have recently spiked. If a neuron spikes then its leakage is increased by a genetically determined amount. If the neuron does not spike then the leakage is decreased by that same amount. Leakage is constrained within the range [0, 1]. The spiking threshold is the same for all neurons in the network and is constant. The neurons are stochastic so that once the spiking threshold has been reached, there is a random chance that a spike will be transmitted along the output weights; either way the cell loses its activation. The neurons send out a stereotypical spike. This is implemented as a binary output. The weights connecting the neurons are constrained within the range [0, 1]. The learning rule employed uses spike timing-dependent plasticity (STDP). The rule used here is implemented using a two-coincidence-detector model [12] Each neuron has its own post-synaptic recording function that is incremented when the neuron spikes and which decays over time in-between spikes. This is compared to the pre-synaptic recording function of the neuron that has transmitted the activation. Each layer of neurons has its own increment and decay rates determined prior to testing via automated parameter optimisation.

2.3 Modulators

Several variants of the network were created; either modulating or non-modulating. Used here, a modulator is a global signal that can influence the behaviour of a neuron if that neuron has receptors for it. The signal decays over time, specified by the re-uptake rate, and can be increased by firing neurons that have secretors for it.

Neurons that are to be modulated are given a random number of receptors. These can be modulated by neurons in other layers that have secretors for those modulators. The receptors modulate either the neuron's sensitivity to input or probability of firing. The effect of this modulation is determined by the level of the associated modulator and whether the receptor is inhibitory or excitatory.

Neurons can also have secretors. These increase the level of an associated modulator. The modulator re-uptake rate, the modulation rate of the receptors and the increment rate of the secretors is determined by artificial evolution along with many other parameters of the neural network before the model is tested.

2.4 Parameter Optimisation

The parameters of the networks were initially optimised using artificial evolution so as to make a fair comparison. Once these constrained evolutionary runs were finished the parameters were hard-coded and tested as a population of 450 agents in order to determine the average performance of the neural network. An average fitness is required because the mapping from genotype to phenotype is stochastic. This is due to the randomisation of weights and the connectivity between neurons. The fitness function used during parameter optimisation was $Energy + Water + Age - absolute(Energy - Water)$.

The absolute difference between the energy and water resource was subtracted from the fitness as both resources were essential for the agent to stay alive. The age was only used for the fitness function during the evolutionary runs and not used afterwards when comparing the average performance of agents with the optimised architectures. This is because agents would generally only die at the beginning of an evolutionary run before the architecture had been optimised.

3 Discrete and Continuous Actions

Modulating and non-modulating versions of the network were implemented and compared in [13]. In all the networks a winner-takes-all selection scheme was used. A single action was chosen each turn by determining the output neuron that had the strongest average activation over multiple iterations of the network. The difference in activation strength between the winning output neuron and the losing neurons was of no consequence. Nor did it matter how strongly the losing output neurons were activated.

If the network is to be used to drive the motors of a robot, or to provide input signals to other neural networks, then it needs to be able to balance the activations of all of its output neurons concurrently.

The previous experiments have used actions that each have one single discrete effect. In the experiments described here, the networks are provided with continuous actions whose effect depend upon the level of activation of the corresponding output neuron. The stronger the activation the greater the effect provided by the continuous action.

In a robot, discrete actions would be the equivalent of motors that either ran at full speed or were switched off. Continuous actions would be the equivalent of motors that ran at a speed determined by the level of the activation they received. The networks have to learn to provide the correct activation to all of the output neurons concurrently rather than only be concerned about which neuron is more strongly activated than all the others.

3.1 Exploratory Two-Modulator Network Optimised for Use with Discrete Actions

The modulating network analysed in [14] and [13] had two modulators, one to signal hunger and another to signal thirst. The neurons in the input layer each had a secretor for the modulator that corresponded to the resource the input neuron pertained to. The neurons in the middle layer had a random number of excitatory or inhibitory receptors for these modulators, see Fig.1a).

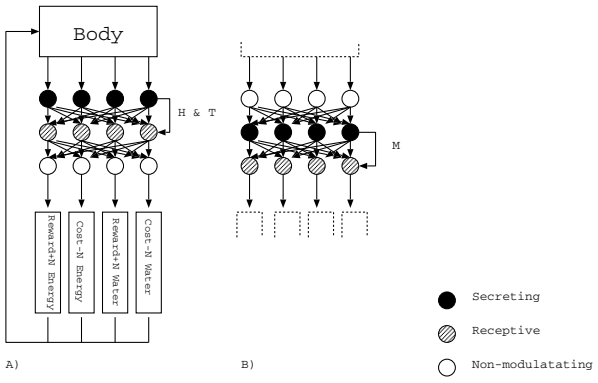

Fig. 1. The agent consists of a body that contains water and energy levels. A) Two-modulator agent: Hunger (and thirst) neurons increase the strength of the hunger (or thirst) modulator when they fire. Neurons in the middle layer have a random number of inhibitory receptors for these modulators. B) Single-modulator agent: Neurons in the middle layer increase the strength of a single modulator when they fire. Neurons in the output layer have a random number of excitatory receptors for this modulator.

Having the input layer modulate the middle layer was shown to increase exploration. As a consequence of this the performance of the modulating agent was slightly below that of the non-modulating network. Actions that were costly or neutral were less likely to be ignored throughout the evaluation period. But conversely, the modulating network was more able to adapt when the effect of actions changed.

3.2 Networks Optimised for Use with Continuous Actions

Many different variants of the network were implemented and tested. The aim was to find the best way of modulating a minimal disturbance network for use with continuous actions. Permutations included having the input layer modulate the output layer, using between one and four modulators and having layers modulate themselves. The parameter sets were optimised for use with continuous actions using artificial evolution. If the architecture performed particularly well then the parameters were hard-coded and tested more thoroughly.

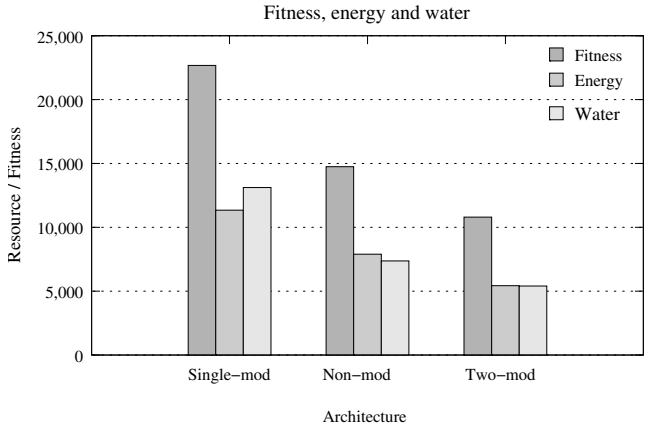

Fig. 2. Fitness, energy and water levels of the different architectures. In order of performance: single modulator (middle to output layer), non-modulating and two modulators (input to middle layer).

The best performing design used a single modulator secreted by the middle layer to modulate neurons in the output layer, see Fig.1b). The non-modulating and two-modulator architectures, originally optimised for use with discrete actions, were re-optimised for use with continuous actions. The parameters of all three architectures were hard-coded and tested using a population of 450 agents. The average fitness, energy and water levels for each architecture can be seen in Fig.2. The explorative behaviour of the two-modulator architecture carries a cost in performance when used in relatively stable environments as the agent tries other actions that have not necessarily proven successful in the past.

4 Adaptive Performance of the Networks

The synaptic weights between the input and the middle layer of the network can be thought of as providing 'activity diffraction' to allow the input signals to filter through the system at different speeds. The synaptic weights between the middle layer and the output layer can be thought of as providing 'activity integration', integrating those signals back into combinations that allow particular output neurons to fire more frequently than others.

Because activity filters through the network at different speeds, some output neurons will fire earlier than others. If an action is rewarding and subsequently reduces the input signal to the network, synaptic activity will be reduced for the other neurons and therefore will be less likely to fire. If an action is not rewarding, the input signal is not reduced, other neurons will eventually fire and other actions will be tried instead.

4.1 Input to Middle Layer Modulation

The hunger and thirst modulators of the two-modulator agent optimised for use with discrete actions inhibit the neurons in the middle layer. The strongest firing neurons have more activation to lose when being inhibited. These are also the neurons more likely to be firing the output neurons that lead to actions that reduce total input activity into the network. So by inhibiting the neurons in the middle layer the 'diffraction' of activation throughout the network is reduced and other actions have a greater chance of being performed. This increases exploratory behaviour.

4.2 Middle to Output Layer Modulation

The most successful network optimised for use with continuous actions has the middle layer modulating the output layer. The receptors of the output layer for the single-modulator network have all evolved to be excitatory. This suggests that modulation is used to excite output neurons that lead to rewarding actions. In other words, modulation is used to balance the outputs of the neural network.

Evidence for this comes from using the single-modulator network with discrete actions even though it has been optimised for use with continuous actions. It performs better than a non-modulating network optimised for use with discrete actions. Not only does the single-modulating network achieve greater average energy and water resource levels (energy=853, water=853) than the non-modulating network (energy=790, water=757), it also manages to avoid having more of one resource than the other.

With the non-modulating network, the more rewarding an action, the stronger the activation of the corresponding output neuron. In contrast, the single-modulating network only fires the outputs leading to rewarding actions and ignores the neutral ones even when there is no need to do so, (see Table 1).

Table 1. The average frequency of discrete actions chosen by a population of 450 agents. Two architectures are compared, the non-modulating architecture optimised for use with discrete actions, and the single-modulator architecture optimised for use with continuous actions.

Action	Amt	Resource	Non-mod freq.	Single-mod freq.
Inactive	-2&-2	E&W	0.0131111%	0.0948889%
Cost	-2	E	1.32978%	0.944%
Cost	-1	E	1.30733%	0.960889%
Neutral	0	E	2.42911%	0.994889%
Reward	+1	E	7.15533%	5.92333%
Reward	+2	E	37.9384%	41.2469%
Cost	-2	W	1.53378%	0.922667%
Cost	-1	W	1.652%	0.956222%
Neutral	0	W	2.60533%	1.00356%
Reward	+1	W	7.57089%	5.59422%
Reward	+2	W	36.4649%	41.3584%

This suggests that the single-modulating network provides reduced activations for all of its output neurons by default and uses modulation to excite the output neurons which are rewarding.

5 Exploitation vs. Exploration

Having the middle layer modulate the output layer helps the agent exploit the actions that are found to be the most rewarding whilst ignoring those actions that are neutral or costly. To further demonstrate this, the networks were tested using discrete cost / reward actions modified to work on the principle of 'use-it-or-lose-it'. This gives exploitative agents an advantage. The actions work as follows:

- If an action is performed for the first time then it provides its maximum effect.
- If the agent continues to perform that action then the it will continue to provide its maximum effect.
- If another action is performed then the potential effect of the original action will decrease each turn until it reaches a minimum regardless of whether the agent uses it or not. The minimum potential effect is anything less than 1 resource point. After the action reaches this minimum it will return to providing its maximum effect when used.

If an agent explores other actions and returns to the original action found to be the most rewarding so far, the effect of that action will be reduced for each turn that the agent performed other actions. If the agent continues to use that action thereafter, the effect will continue to be reduced each turn until it reaches a minimum. At this point the action returns to providing its maximum effect again.

Each network was tested using a population of 450 agents. They were tested 102 times; for each evaluation the ratio of the action's previous effect being retained was incremented by 0.01. For example, at a ratio of 0.5 the potential effect that an action can provide is halved each round once the agent stops exploiting it continuously. The actions are discrete so the agents can only pick one action per turn. This is the action whose corresponding output neuron has the strongest average activation.

The performance of the three architectures can be compared in Fig.3. It can be seen that the performance of each architecture declines as the ratio reaches 0.99. This is because once the agent stops using an action, it takes longer for the potential effect of using that action to reduce to the minimum before returning to its maximum level again. When the ratio reaches 1 the performance of all three architectures reverts to the same level as at 0. It is not plotted here for the sake of clarity.

The single-modulator architecture is the best performer with each agent in the population increasing their energy and water levels by the highest average amount each time. At a ratio of 0.99, the single-modulator architecture performs as well as the non-modulating architecture but the performance increases as

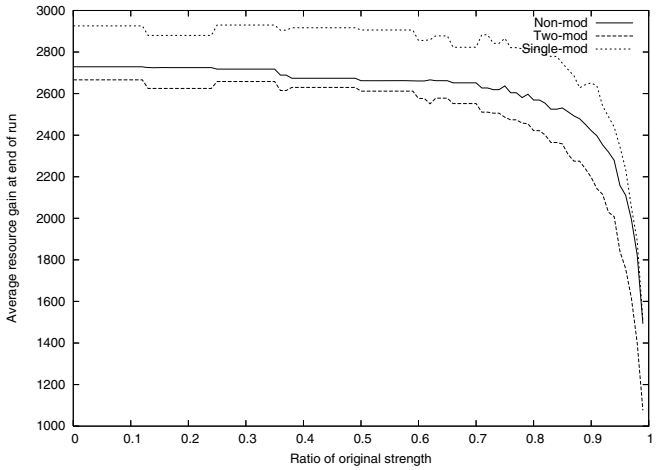

Fig. 3. Testing the networks using discrete use-it-or-lose-it actions show how well they cope with tasks benefitting from exploitative behaviour. The single-modulator network performs significantly better. The two-modulator network, previously shown to perform better at tasks benefitting from explorative behaviour, performs worst of all.

the ratio decreases. The two-modulator architecture performs worst of all. Its performance at a ratio of 0.99 is significantly below that of the other two.

6 Conclusion

Taking modulation as a mechanism underlying emotions, we have investigated how such a mechanism can bias an artificial neural network towards exploration of new courses of action, as seems to be the case in positive emotions, or exploitation of known possibilities, as in negative emotions such as predatory fear. Modulation can be used to both concurrently provide the correct activation to each neuron in the output layer, and to bias a network towards either exploration or exploitation.

If an emotion is merely a particular subset of neural functions found by evolution to provide the optimal behaviour for an agent given a certain environmental or bodily state, then those neural substrates need to be activated concurrently. Each neural function may also require a different degree of activation. This means that we may need a single neural network to find the optimal balance of activation for each of its output neurons so that it can later be used to drive other neural networks.

Further work is required to determine whether exploration and exploitation networks should be driven by a third, arbitrating neural network, and whether the correct network can be selected using neuromodulators. It may also be the case that a single neural network can be biased towards either exploitation or exploration at runtime, as in [9], by modulating the re-uptake rate.

References

1. Koch, C.: Biophysics of Computation. Oxford University Press, Oxford (1999)
2. Fellous, J.M.: The neuromodulatory basis of emotion. The neuroscientist 5(5), 283–294 (1999)
3. Kelley, A.E.: 3. In: Who needs emotions? The brain meets the robot, pp. 29–77. Oxford University Press, Oxford (2005)
4. Damasio, A.: Descartes' Error: Emotion, Reason, and the Human Brain. Quill (1994)
5. Evans, D.: The search hypothesis of emotion. British Journal for the Philosophy of Science 53(4), 497–509 (2002)
6. Nesse, R.: Evolutionary explanations of emotion. Human Nature 1(30), 261–289 (1990)
7. LeDoux, J.E.: The Emotional Brain. Simon & Schuster (1998)
8. Avila-García, O., Cañamero, L.: Hormonal modulation of perception in motivation-based action selection architectures. In: Avila-García, O. (ed.) Proceedings of the Symposium on Agents that Want and Like: Motivational and Emotional roots of Cognition and Action at the AISB-05 conference, The society for the study of artificial intelligence and the simulation of behaviour, pp. 9–16 (2005)
9. Blanchard, A., Cañamero, L.: Developing affect-modulated behaviors: Stability, exploration, exploitation, or imitation? In: Kaplan, F. (ed.) Proc. 6th Intl. Workshop on Epigenetic Robotics, vol. 128, Lund University Cognitive Studies (2006)
10. Wehmeier, U., Dong, D., Koch, C., van Essen, D.: Modeling the mammalian visual system. In: Koch, C., Segev, I. (eds.) Methods in Neuronal Modeling: From synapses to networks, pp. 335–360. MIT Press, Cambridge (1989)
11. Wörgötter, F., Porr, B.: Temporal sequence learning, prediction and control - a review of different models and their relation to biological mechanisms. Neural Computation 17, 1–75 (2004)
12. Karmarkar, U.R., Najariana, M.T., Buonomano, D.V.: Mechanisms and significance of spike-timing dependent synaptic plasticity. Biological Cybernetics 87, 373–382 (2002)
13. Parussel, K.M.: A bottom-up approach to emulating emotions using neuromodulation in agents. PhD thesis, University of Stirling (2006)
14. Parussel, K., Smith, L.: 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, The society for the study of artificial intelligence and the simulation of behaviour, pp. 98–101 (2005)