INTERNALLY SELF ORGANISING NEURAL NETWORK FOR ONLINE LEARNING OF PERCEPTION TO ACTION MAPPINGS IN SENSOR NETWORKS

D.D. Ediriweera, I.W.Marshall
University of Kent, U.K.
Email: dde3@kent.ac.uk, i.w.marshall@kent.ac.uk

***Abstract* – This paper presents a neural approach for learning relationships between perceptual classifications and real world actions in sensor networks. The mapping is constructed online using short-term non-specific feedback from the environment. The presented architecture combines an online adaptable classifier with a novel neural mapping function. Simulations demonstrate learning to be fast and stable, and the network to be topologically and synaptically adaptable in response to changing feedback. Results obtained from our experiments demonstrate the possibility of effective online learning using non-specific indicators.**

1. INTRODUCTION

The paper presents a neural approach for learning relationships between perceptual classifications and real world actions. These neural mappings are useful for sensor networks where meaningful and adaptable mappings are required between sensors and actuators; for example in home automation networks. Here we present an approach capable of providing performance guided learning and autonomous adaptation for sensor based inference. Due to the close analogies between robot sensor configurations and actual sensor networks, and due the immense amount of research done in robotics for similar problems, we use robotics as a source of insights, and further as a model for presenting and testing the proposed approach.

Mobile robot control is a problem explored by many researchers over time. Robot controllers in early days were mostly designed using the Sense-Model-Plan-Act (SMPA) as described by Brooks in [1]. Although quite useful, SMPA encompassed several weaknesses which led to the investigation of other inference models. The main problem with SMPA is its approach of selecting and planning actions based on an internal model of the real world. This meant that accurate actions can be chosen only if the internal model represented the external environment accurately. This proves to be quite difficult considering the complexity and the dynamic behaviour of the real world. As a result, a more dynamic and reactive mechanism for action selection was introduced. This is known as Behaviour Based Robotics [1, 2]. In this approach, the world is considered to be its own model, and actions are selected by continually referring to sensors, and thereby building a dynamic perception of the environment. Computationally this is much simpler, and does not require any internal models, but interestingly this introduces us to the problem of mapping different perceptions into actions. Different approaches to solve this problem have been explored. Behavioural modules have been built using fuzzy logic controllers, potential field techniques, back propagation networks, self-organising maps and associative memory networks. As outlined by Dubrawski [3] most have their own limitations. Compared to these techniques Reinforcement Learning (RL) [2] is relatively novel, and although still in its early stages, it has already proved itself to be quite useful for designing controllers [2, 4, 5]. The main advantage with RL is its ability to discover near optimum solutions based on external feedback. Thus, in contrast to some earlier techniques, RL can be used to develop solutions for problems which are not clearly defined.

Many approaches have been investigated on using RL for robot control [2, 4, 5]. Specific attempts to combine a topologically altering neural classifier with RL were made by Andres Perez [5]. He explores the RL based algorithm SARSA [5], built in combination with a Flexible Adaptable Size Topology (FAST) [6]. RL is used to update a table of Q values which allows the selection of the best action for a given state. The problem with this approach is its relatively long convergence time in high dimensional output spaces, and its limitations in generalisation. In contrast, we investigate a different approach to use feedback to resolve a simple neural mapping function between perceptions and actions (Figure 1). The perceptions are defined by a topologically adaptable classifier. The classifier receives sensor readings, and according to their features classifies them in to dynamic perceptual categories. Categories are considered dynamic as they are created and destroyed as required. The sensitivity of each category can be controlled by parameters allowing the network to support parametric control over generalisation. Each of these categories directly map into a neuron in the dynamic input field of the mapping network. This input layer is dynamic as it allows input neurons to be added and deleted, and to do so with minimal impact on previously learnt knowledge. This dynamic topology enables the architecture to support an undefined number of internal perceptions, and corresponding mappings to actions. As the topology of the network is configurable according to performance, it is able to autonomously and dynamically discover near optimum topologies for changing environments and requirements.

2. THE PROPOSED ARCHITECTURE

The proposed architecture consists of two main components (Figure 1); the perceptual classifier and the neural mapping function. Considering the current existence of several topologically adaptable classifiers; for example ART and FAST, this paper concentrates more on the task of constructing the neural mapping function. The classifier used for our experiments is an enhanced version of the FuzzyART [7] network. FuzzyART is a latter version of the Adaptive Resonance Theory (ART) introduced by Stephen Grossberg [9]. The reason for selecting FuzzyART is its simplicity over other ART networks supporting analogue classification.

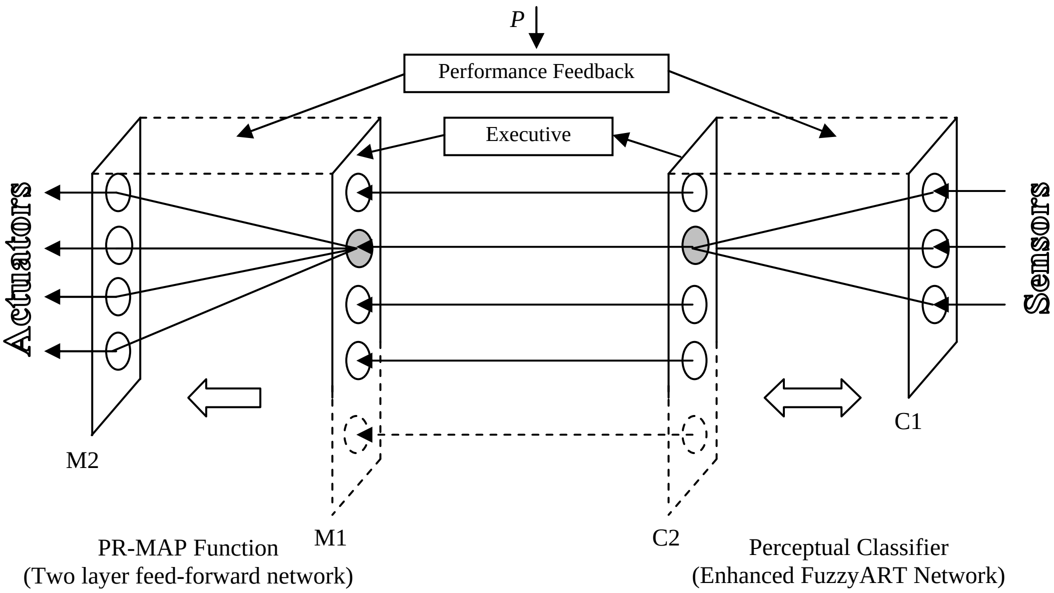


Figure 1: Proposed Architecture featuring the perceptual classifier, and the neural mapping function

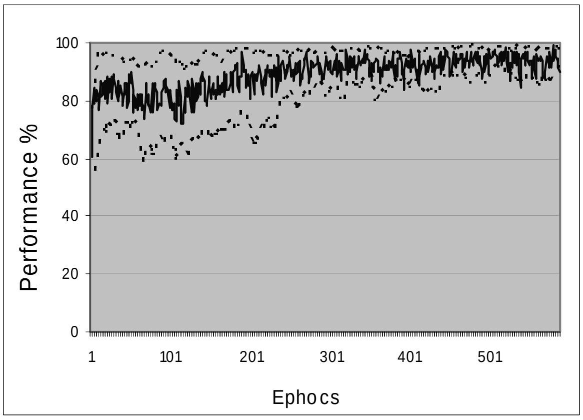
Layers C1 and C2 of the proposed architecture (Figure 1) compose the FuzzyART classifier. Weights between C1 and C2 are bidirectional. The FuzzyART classifier uses a normalisation mechanism known as complement coding to avoid category proliferation. We use the Snap-Drift [8] algorithm as the learning mechanism for the classifier. This helps to ensure better network stability. The FuzzyART network uses Winner Take All (WTA) activation at the C2 layer. This allows the network to preserve knowledge while allowing its internal configuration to be dynamic. If new input patterns are distinctly different from existing clusters, the network adds a new neuron at C2, and trains this to recognise the new pattern. This results in topological changes at C2.

Layers M1 and M2 define the neural mapping function. M1 neurons receive inputs from the C2 neurons. Mappings from C2 to M1 are one to one; these neural pathways do not accompany weights. M1 layer is a dynamic neuron field with a changing topology. Changes are triggered by reconfiguration at the C2 field. M1 topology adaptations are communicated by an executive process external to the network. When a new neuron j is added to M1, its weight vector W_j is initialised by the executive process. The initialisation values are selected by copying weight values from an existing M1 neuron. The selected M1 neuron for this purpose is the M1 counterpart of the C2 neuron to which the said input generated the highest activation. This process is performed to ensure that a sensible generalisation is possible for inputs which are novel to the system. Random 1s and 0s are used to initialise weights for the first M1 neuron. Feedback received from the environment is short-term and non-specific. Short-term, meaning feedback is received on the basis of each action, and non-specific, meaning feedback does not carry any localised meaning relating to output patterns, nor to individual output neurons; it simply resembles the overall performance of the system.

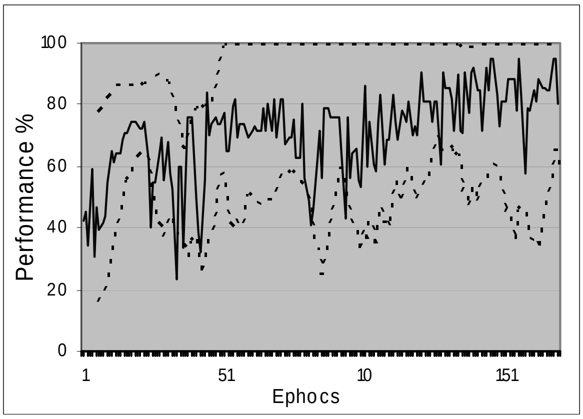
3. EXPERIMENTS AND SIMULATIONS

Experiments were targeted to investigate network convergent rate, network accuracy, and network sensitivity to changing feedback patterns. For this we evaluate the network using several experiments: (i) simulations to investigate network ability to reach a convergent state; (ii) simulations to investigate network sensitivity to complex and non-complex feedback; (iii) simulations to analyse topological changes of the network; (iv) simulations to analyses network responses to changing environments. These simulations were performed using simulated data. Data was generated to simulate readings from a 2x4 Passive Infrared Sensor array. All inputs were continuous between 1 and 0.

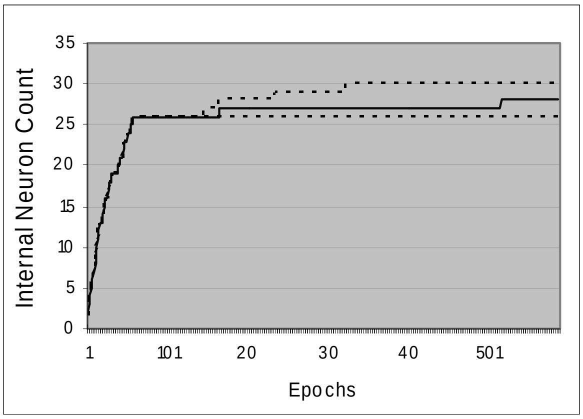
The experiments model a navigation task for a mobile robot. The purpose of the network was to dynamically cluster inputs, and to then map each cluster to an appropriate 10 bit output pattern. The output pattern indicates suggested actuator activations. The clusters are dynamically formed by grouping geographically similar inputs; for example, IR patterns suggesting obstacles on the left might form one cluster, and IR patterns suggesting obstacles on the right might form another; this is sensible for navigation tasks. For each such cluster, we randomly generated a desired output pattern. These patterns are used as stereotypes to provide non-specific feedback (*P*) to the network. Depending on the feedback, size of individual clusters might be adjusted; this could lead to the generation of further clusters. The task of the network is to fine-tune clustering, and to autonomously train itself to maximise the match between its outputs and the desired output patterns. Figure 2 illustrates a set of selected results. Performance indicated below indicates typical accuracy and consistency of the network in mapping a given 8bit IR input pattern to a desired 10bit output pattern (actuator activation).



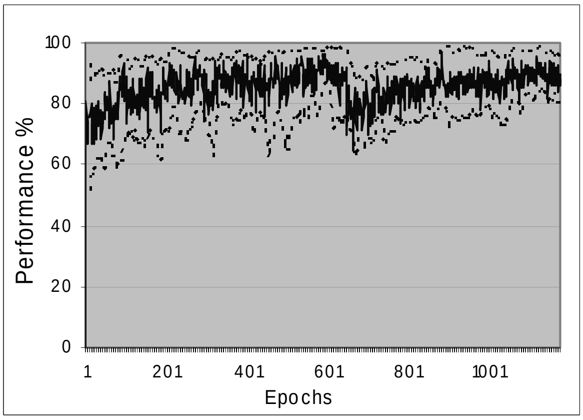
**(A) Network performance over a period of 600 epochs.
Epoch Size = 25 inputs.**



**(B) A closer view of the first 170 inputs of graph (A).
170 epochs with epoch size set to 1.**



**(C) Internal neuron count over a period of 600 epochs.
Epoch Size = 25 inputs**



**(D) Network performance over a period of 1200 epochs.
The feedback indicator is changed at 600 epochs.**

Figure 2 (A.B.C.D): Illustrates selected results from our experiments. The dotted lines represent the maximum and minimum values recorded at each epoch. The graphs represent the average values at each epoch. Values are calculated over 5 to 10 runs of each simulation. Each experiment was performed with 75%-100% complex feedback and random noise. 1400 input patterns were used.

The first experiment investigates the ability of the network to converge. As Figure 2 (A) illustrate, the network reaches a state of convergence in approximately 150 epochs. Because the network is continually learning, in this experiment we consider average performance of 80% to be a convergent state, and further consider the network to be stable if performance remains above this.

In the second experiment we investigate the networks response to complex and non-complex feedback. Feedback is considered complex when it is increasingly dependent on the complete output pattern rather than on separate segments of the output. These patterns are harder to discover using non-specific feedback. Based on results observed in our experiments, we can conclude that the network is able to converge under both complex and non-complex feedback, but as expected, it is faster under non-complex feedback (in this case approximately 50 epochs vs. 120 epochs).

The third experiment analyses the topological response of the network to changing performance. As Figure 2 (C) illustrates, network topology starts to stabilise as network performance starts to increase. This is the desired behaviour. This allows us to draw the conclusion that the network is able to discover suitable internal configurations based on non-specific feedback. An additional aspect here is that results illustrated are based on a network which performs additive adaptation. In the future we plan to develop a mechanism of adding as well as removing internal neurons for better adaptability.

In our fourth experiment we investigate the adaptability of the network. Here we allow the network to freely operate for 600 epochs; as Figure 2 (D) illustrates, at this stage the network is at a convergent and a stable state. At 600 epochs we changed the performance indicator of the network and allow it to run an additional 600 epochs. As in Figure 2 (D), soon as the performance indicator is changed, network performance drops, and in response the network starts a process of learning until it again reaches a convergent stable state at approximately 750 epochs.

Our initial results indicate the plausibility of the proposed network and its associated architecture. Initial studies indicate its ability to perform online adaptive learning using non-specific feedback. The network is able to reach stable convergent states relatively fast [Figure 2 (A), (B)]. From our findings, we hope to extend our work to perform more intensive tests using robot simulators, and based on these results to implement the architecture on a real robot using parallel hardware architecture which can be generalised well towards sensor networks.

4. Conclusion and Further Work

Work outlined in this paper explores a performance guided, internally self organising neural network with a dynamic topology. The dynamic nature of the network allows it to be adaptable and support controlled generalisation. The presented mapping algorithm is able to converge relatively fast as it prioritises search for non-complex patterns over complex patterns. The proposed architecture is tested using specific test simulations. Results of these simulations illustrate the ability of synaptic learning and network topology changes using non-specific feedback. This approach is ideal for pure behavioural systems, but is not sufficient when short-term planning is required. As a result, in addition to performing further testing on the current architecture, we also hope to investigate the possibility of mappings perceptions into actions distributed spatially as well as temporally, and further to explore possibilities for faster learning using complex feedback.

References

[1] R.A. Brooks. Intelligence Without Reason. Technical Report A.I. Memo 1293, Massachusetts Institute of Technology, 1991.

[2] R.C. Arkin. Behavior-Based Robotics, ISBN: 0262011654. The MIT Press, Cambridge, MA, 1998.

[3] Learning locomotion reflexes: A self-supervised neural system for a mobile robot. Artur Dubrawski, James L. Crowley. Robots and Autonomous Systems, Volume 12, pp 133-142, 1994.

[4] Biological Robot Arm Motion through Reinforcement Learning.Jun Izawa.Toshiyuki Kondo.Koji Ito, Proceedings of the 2002 IEEE International Conference on Robotics & Automation. pp 3398-3403. 2002.

[5] A. Pérez-Uribe. A non-computationally-intensive neurocontroller for autonomous mobile robot navigation. Chapter 8 in Biologically inspired robot behaviour engineering, Series: Studies in Fuzziness and Soft Computing. VOL. 109, R. J. Duro, J. Santos, M. Grana (Eds.), Springer-Verlag, 2002, pp. 215-238.

[6] A. Pérez-Uribe. Eduardo Sanchez. The FAST Architecture, a Neural Network with Flexible Adaptable-Size Topology. Proceedings of the V International Conference on Microelectronics for Neural Networks and Fuzzy Systems, MicroNeuro'96.

[7] G.A.Carpenter. S.Grossberg. Fuzzy ART: Fast Stable Learning and Categorization of Analogue Patterns by an Adaptive Resonance System. Neural Networks, Vol. 4, pp. 759-771, 1991.

[8] S. W. Lee. D. Palmer-Brown. C. M. Roadknight. Performance-guided Neural Network for Rapidly Self-Organising Active Network Management, Neurocomputing, special issue on Hybrid Neurocomputing, Elsevier Science, Netherlands, 61C: 5 - 20. 2004.

[9] G.A. Carpenter, S. Grossberg, “A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition machine”, Computer Vision, Graphics and Image Processing, 37, pp 54-115, 1987.