USING SELF-ORGANISING FEATURE MAPS FOR ADAPTIVE CONTROL

Kevin Warwick* and Nigel Ball+

*Department of Cybernetics, School of Engineering and Information Sciences, University of Reading, Whiteknights, Reading RG6 2AY, UK.

+Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK.

Introduction

Kohonen's self-organising feature map [1] maps input data from an n-dimensional space of real numbers on to a 2-dimensional array of S neurons. The neurons are laterally interconnected, forming a feature map which can adapt itself in accordance with the input data.

Self-organizing feature maps have been implemented in a wide range of application areas such as speech processing, image processing, optimization and robotics. Recent variations to the basic model have been proposed to enable it to order a state space using a subset of the input vector and to apply a local adaptation procedure that does not rely on a predefined test duration limit. Both these variations have been incorporated into a new feature map architecture that forms an integral part of an Hybrid Learning System (HLS) based on a genetic classifier system. Problems are represented within HLS as objects characterized by environmental features. Objects controlled by the system have preset targets set against a subset of their features. The system's objective is to achieve these targets by evolving a behavioral repertoire that efficiently explores and exploits the problem environment. Feature maps encode two types of knowledge within HLS - long term memory traces of useful regularities within the environment and the classifier performance data calibrated against an object's feature states and targets. Self-organization of these networks constitutes non genetic-based (experience-driven) learning within HLS.

This paper presents a description of the HLS architecture and an analysis of the modified feature maps and genetic adaptive strategy. Initial results are presented that demonstrate the behaviour of the system on simple control and maze running tasks.

HLS Architecture

HLS's feature maps represent a new application of Kohonen Feature Maps to the adaptive control of systems with nonlinear behaviour [2,3]. Learning within the modified feature map is via a local learning algorithm which operates within each classifier layer (intra) and between layers (inter). The learning process calibrates classifiers according to their behaviour in the task environment as defined by the state/goal space.

The HLS encodes classifier condition values within self-organizing neural network structures, and controls a number of objects in accordance with the defined feature objectives using an evolving behaviour pattern base [4].

The overall system, shown in Figure 1, has two roles, firstly it acts as a long term associative memory between states in terms of internal goals and external sensory characteristics, and secondly as a classifier to produce optimal behaviour based on time dependent feature objectives. The two roles are dealt with by a adaptive algorithm and feature maps, both the network and maps employing Kohonen's self-organizing feature maps. The rest of this paper addresses the structure and behaviour of the correlation network.

Adaptive Algorithm

Adaptation within each network node is controlled by two variables, state match (S) and goal match (G). S indicates the difference between a node's weight vector and the input state at the time of last node selection. The variable controls the excitatory adaptation that the node applies to its network neighbours. G meanwhile, indicates the difference between a node's weight vector and the current feature goals at the time of last node selection. This allows the network to differentiate between goal directed and non-goal directed data.

The state match is given by:

$$S = [\sum_i (x_i - w_i)^2]^{0.5} \tag{1}$$

where x_i is the ith element of the current input vector, whereas w_i is the i^{th} element of the node weight vector. The winning node *c*, is then defined by:

$$S_c = \text{minimum } S \tag{2}$$

The goal match of the winning node *c* is then given by:

$$G_c = \min_{if}(G_{if} - E_{if})^2 + (\sum_{ef}(E_{ef} - W_{ef})^2)^{0.5} \tag{3}$$

in which if/ef denotes an internal/external feature respectively. Also, G_{if} is the current goal for an internal feature, $E_{ef/if}$ is the current state for a feature and W_{ef} is the node weight element for the external feature.

Weight adaptation within node i by node c for feature f is given by:

$$w_f^{t+1} = w_f^t + C' * [S_c * G_i] * (E_f^t - w_f^t) \tag{4}$$

C' = adaptation rate associated with three adaptation zones -

1° (+), 2° (-), 3° (+) - of the "mexican hat" function.

Feedback is applied to node c if $g_c > 0$ as defined by:

$$w_f^{t+1} + 1 = w_f^t + C'/G_c^2 * (E_f^t - w_f^t) \tag{5}$$

The correlation network is interrogated iteratively with each pre-defined feature objective to produce "n" correlation vectors corresponding to the best correlation known by the system for achieving the n feature objectives. The correlation of node *c* for an internal feature *if* is given by:

$$d_{if} = (G_{if} - W_{if}) + \sum_{ef} (\epsilon_{ef} - W_{ef})^2)^{0.5}) * G_c \tag{6}$$

The best correlation for each internal feature is given by the set of nodes $\underline{c}$ such that for each c:

$$d_{ifmin} = \text{minimum (over all nodes) } (d_{if}) \tag{7}$$

These correlations are then ranked to establish a priority internal target for the current time epoch given by:

$$d_{ifmax} = \text{maximum (over } \underline{c}) \ (d_{ifmin}) \tag{8}$$

The node c associated with d_{ifmax} has its goal match value adjusted by a decay term:

$$G_c := G_c * (1.00 + \text{decay}) \tag{9}$$

The priority internal target is translated into a set of external targets that can be achieved through classifier activation by associated with the corresponding external feature elements of the correlation vector.

Application Results

Operation of the HLS system has been documented in a number of papers [5]. Recent research has however concentrated on comparisons of HLS's performance against Holland and Reitman's CS-1 system [6]. CS-1 guided an artificial animal in two simple mazes (Figures 2(a & b)) so as to minimize the number of steps taken to get to food and water. Knowledge learnt by CS-1 on the first maze was successfully transferred to the second maze where CS-1 almost immediately converged to an optimal time epoch of 6 steps.

Without prior exposure to the first maze, CS-1 took approximately 1200 time epochs (10,000 time steps) to achieve optimal performance. The second CS-1 maze environment (Figure 2(c)) has a toroidal wrap-around applied to an extended 1D maze ranging over locations 1 - 24. Water and food resources are located at $x = 6$ and $x = 18$ respectively and an artificial organism under HLS's control consumes any resource within its current location. The initial classifier population contains the following members: +x, -x, {null}. The organism detects one external feature, x location, and two internal features - thirst (t) and hunger (h) (both inversely related to the object's food/water reservoirs). Values of the food reservoir and water reservoir range from 0 (empty) to 36 (full). Food and water are used up at a rate of 0.2 units/time step if the organism is stationary and 0.4

units/time step if the organism moves. When the organism lands on a food/water resource, it completely replenishes the corresponding reservoir. Initial values for each feature are x = 12, t = 9, h = 19. Goals are specified for the internal features of t = 18, h = 36 i.e. complete satiation, which the organism attempts to meet continuously. However this is not possible since the goals are mutually exclusive as food and water sources are located at opposite ends of the maze. Therefore an optimum strategy is to alternate between the sources in such a way as to maintain food/water reservoir levels at 73% each. To achieve this the organism must learn how to apply its behavioral repertoire (+x, -x, null) in its exploration of the maze and then learn the stable points associated with the food/water sources.

The problem has been tested with the feedback mechanism on the correlation network enabled and disabled to evaluate the effect of feedback on the rate of goal learning and association.

With the feedback mechanism disabled the organism was able to find food and water most of the time, but dithered somewhat between sources because no stable memory trace was formed. Further, food and water reservoir levels were erratic and only averaged 60% and 40% respectively. However with feedback enabled, the correlation network, encodes a stable memory of the location of both food and water, with average reservoir levels at 80% and 60% respectively. This is near optimal, although the organism's strategy is one of continuous alternation between sources rather than favouring the water resource.

Conclusions

It has been shown in this paper that it is possible to employ a local learning algorithm which uses domain-specific data in self-organizing a correlation around useful environment states. The resulting network functions as an associative memory enabling the system to develop a set of useful behaviours (classifiers) to control a multi-feature object.

Future research into hybrid genetic and neural networks will include investigations into different network self-organization algorithms to improve the rate of classifier calibration (such as those with no adaptation profile or with different selection/adaptation vectors). Improvements in the classifier fitness function must also be found to enable the rate of evolution within the classifier population to be increased.

References

1. Kohonen, T., "Self-organization and associative memory", Springer-Verlag, 1984.

2. Hecht-Nielsen, R., "Applications of counter-propagation networks", Neural Networks, Vol. 1, No. 2, pp. 131-139, 1988.

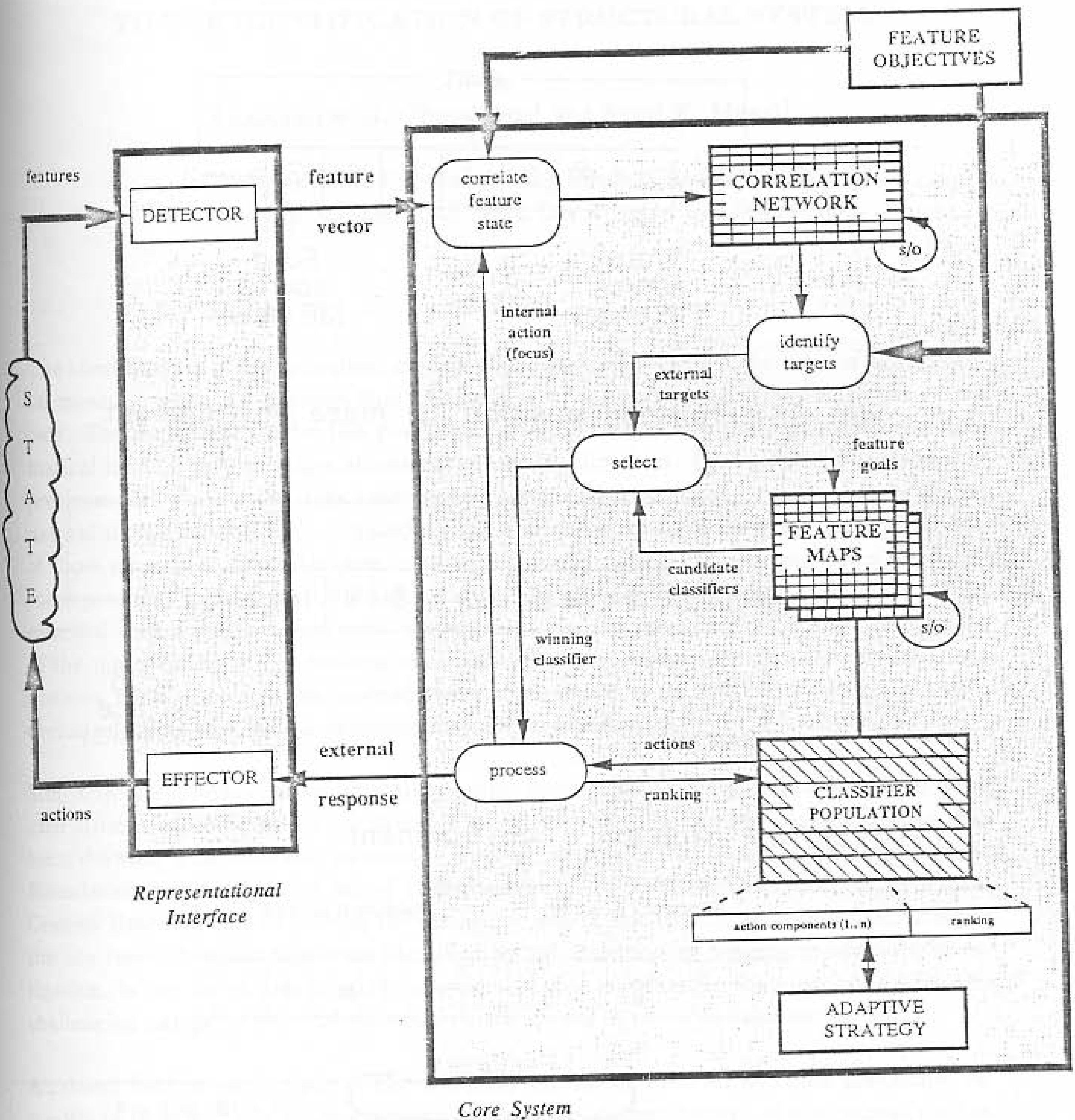
3. Goldstein, M., "Self-organizing feature maps for the multiple travelling salesmen problem", Proc. INNC-90, Vol. 1, pp. 258-261, Paris, Kluwer Academic Publishers, 1990.

4. Ball, N. and Warwick, K., "Using self-organizing feature maps for the control of artificial organisms", Proc. IEE, Part D, Vol. 140, No. 3, 1993.

5. Ball, N. and Warwick, K., "Application of augmented output self-organizing feature maps to the adaptive control problem", Proc. INNC-90, Paris, No. 1, Kluwer, 1990.

6. Holland, J. and Reitman, J., "Cognitive systems based on adaptive algorithms", in Waterman, D. and Hayes-Roth, F. (eds.), "Pattern Directed Inference Systems", pp. 313-329, Academic Press, 1978.

HYBRID LEARNING SYSTEM

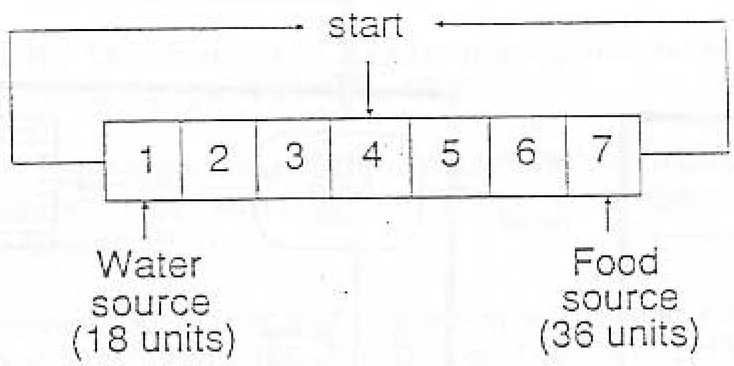


ADAPTIVE STRATEGY - application of G.A operators (based on classifier ranking)

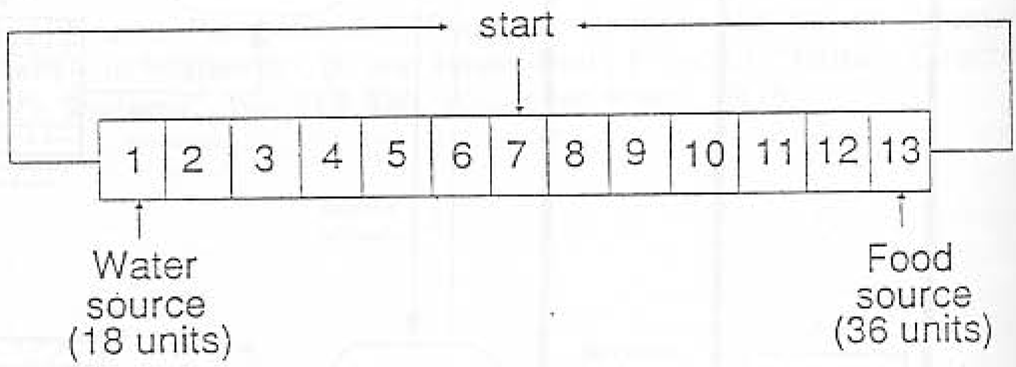
S/O - self organization of neural networks

Figure 1.

(a) CS-1 seven position 1D maze environment



(b) CS-1 thirteen position 1D maze environment



(c) Test problem 6 environment

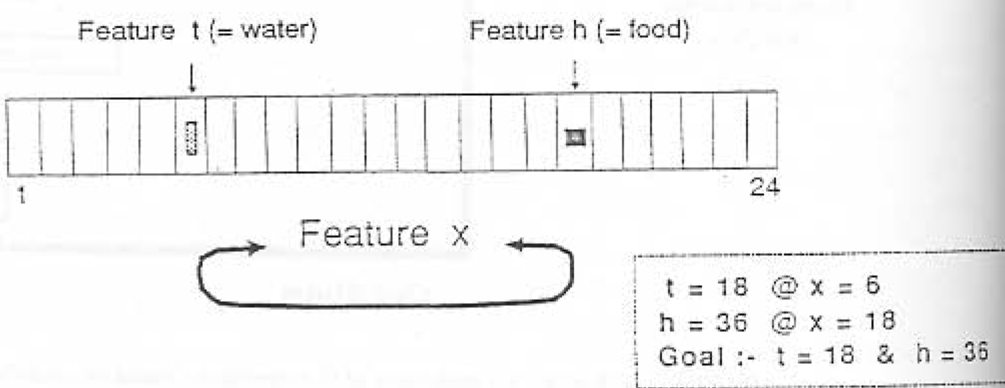


Figure 2.